Shape

Description automatically generated with medium confidence



**Essay / Assignment Title:** Database System Design for a Library

**Programme title: MSc of Data Analytics**

**Name: Turan Can Gün (Q1055616)**

**Year: 2024 (2023 October Intake)**

# CONTENTS

[CONTENTS 1](#_Toc1379585479)

[INTRODUCTION 4](#_Toc1295676359)

[1. DATABASE DESIGN 5](#_Toc580963227)

[1.1 Importance of Database Design 5](#_Toc670222410)

[1.2 Scope and Objectives of the Library Database Design 7](#_Toc1365130315)

[1.3 Entity Relationship (ER) Diagram 9](#_Toc42612324)

[1.4 Design Choices 11](#_Toc782953048)

[2. IMPLEMENTATION 13](#_Toc706191765)

[2.1 Creating Databases 14](#_Toc1116022327)

[2.2 Providing Library Scenarios 15](#_Toc1134761268)

[2.2 SQL Queries 20](#_Toc1616560341)

[3. CAP THEOREM 22](#_Toc851879865)

[3.1 General Review of CAP Theorem 23](#_Toc1117730371)

[3.2 Importance of Three Components for Lbrary and Database Design – Relationship of the Components with Library Database 25](#_Toc2034214470)

[CONCLUDING REMARKS 26](#_Toc610160611)

[BIBLIOGRAPHY 27](#_Toc191189118)

[APPENDIX (if necessary) 28](#_Toc1449928650)

**Statement of compliance with academic ethics and the avoidance of plagiarism**

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

Name and Surname (Capital letters): TURAN CAN GÜN

Date: 05 / 07 /2024

# INTRODUCTION

Database design is crucial for creating structured and efficient systems for data storage, retrieval, and management. Elmasri and Navathe (2016) state that in order to guarantee data integrity, accessibility, and scalability, it entails organizing data elements and defining relationships among them. A well-designed database serves as the backbone of any data-driven application, providing a solid foundation for data management and significantly affecting overall system performance and user experience (Date, 2004).

In the context of a library, an effective database design supports the management of books, members, loans, and reservations. Key components include storing book details such as titles, authors, and availability for easy search and inventory management. Also which involves member records, tracking book loans with dates, and reservation details to be sure book availability. By organizing these elements efficiently, a library database ensures accurate records, enhances user experience, and streamlines operations, making it a fundamental aspect of library management (Hoffer, Ramesh, & Topi, 2016).

# 1. DATABASE DESIGN

## 1.1 Importance of Database Design

Creating information systems that are successful and efficient requires careful consideration of database design. In order to ensure that the database can efficiently store, handle, and retrieve data while supporting a variety of applications and services, it entails the process of structuring data according to a specified model. In addition to enhancing data security and integrity, a well-designed database also boosts system performance and user friendliness.

Making a physical and logical schema that meets an organization's information demands is the main objective of database design. As part of this process, restrictions, relationships, and entities must be defined in order to appropriately represent real-world circumstances. Elmasri and Navathe (2016) state that requirements analysis, conceptual design, logical design, and physical design are just a few of the phases that make up database design.Each stage plays a vital role in ensuring that the database meets user requirements and operates efficiently.

Stakeholder demands are gathered and examined during the requirements analysis phase in order to determine the data that needs to be saved and the actions that need to be done on it. In addition to defining important entities and relationships, this step is crucial for comprehending the goals and scope of the database (Connolly & Begg, 2015).

An abstract model of the database is created during the conceptual design phase; this model is frequently shown as an entity-relationship (ER) diagram. This diagram serves as a clear template for the following design stages by graphically representing the entities, properties, and relationships found in the database. Usually utilizing a relational model, the logical design phase converts the conceptual model into a logical schema.This schema defines tables, columns, primary keys, and foreign keys, ensuring that data is organized in a normalized format to reduce redundancy and improve data integrity (Elmasri & Navathe, 2016).

Finally, the physical design phase focuses on optimizing the database for performance and storage efficiency. This involves selecting appropriate indexing strategies, partitioning data, and determining storage requirements. The physical design must consider the specific hardware and software environment in which the database will operate, as well as anticipated workloads and access patterns (Connolly & Begg, 2015).

## 1.2 Scope and Objectives of the Library Database Design

The database's main purpose is to oversee and simplify the library operations. This entails keeping track of the books, personnel, members, loans, and reservations. maintaining records of books, members, staff, loans, and reservations. The database will facilitate efficient book lending, reservation management, and overall library administration.

We will illustrate objectives of the library databse as table based on 4 features;

|  |  |  |
| --- | --- | --- |
| **Efficient Record Management** | **Enhanced Member Services** | **Operational Efficiency** |
| Maintain accurate and up-to-date records of books, members, and staff. | Provide members with the ability to reserve books. | Simplify the process of lending and returning books. |
| Track the lending and return of books. | Notify members about due dates and overdue books. | Streamline the management of staff responsibilities and communication |
| Manage reservations and ensure books are available for reserved members. | Allow staff to manage member details, including contact information. | Facilitate reporting and analytics on library usage and book circulation. |

Table 1.1 Fundamental Objectives of the Library Database

|  |  |
| --- | --- |
| Stakeholders | Requirements |
| Library Members | Require a user-friendly system to search for, reserve, and borrow books.Need timely notifications regarding due dates and reservations. |
| Library Staff | Need tools to manage book inventory, member records, loans, and reservations. Require functionality to update and maintain member and book information. |
| Library Management | Require reports on book circulation, member activity, and operational efficiency.Need insights into popular books and member preferences to make informed purchasing decisions. |

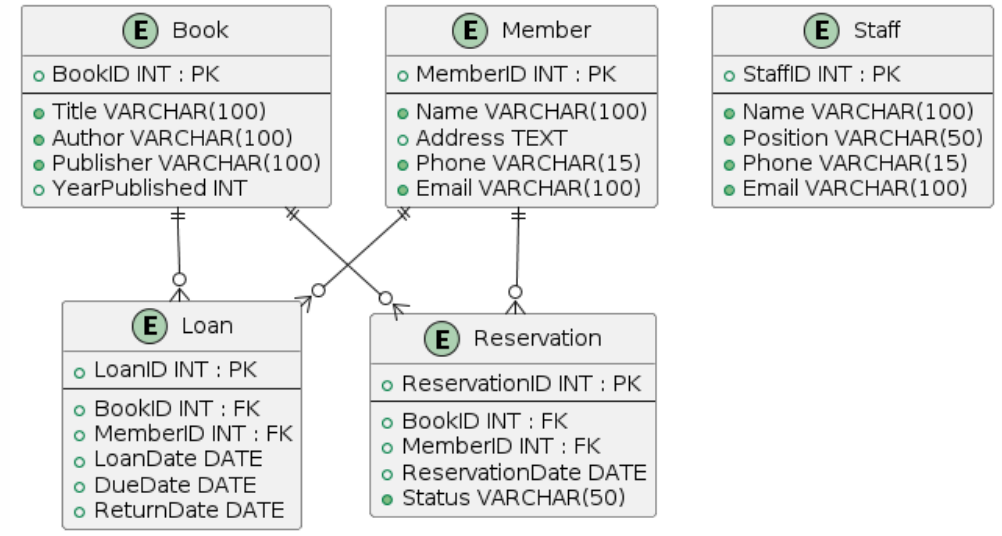
Table 1.2 Stakeholders of the Library Database

|  |  |
| --- | --- |
| Functional Requirements | Non - Functional Requirements |
| |  | | --- | | **Book Management** |  |  | | --- | | - Add, update, and delete book records. Track book availability and lending history. | | |  | | --- | | **Performance** |  |  | | --- | | - Ensure quick retrieval of book and member information. Handle multiple concurrent users without performance degradation. | |
| |  | | --- | | **Member Management** |  |  | | --- | | - Add, update, and delete member records. Manage member contact information. | | |  | | --- | | **Security** |  |  | | --- | | - Protect member and staff personal information.- Ensure only authorized access to sensitive data. | |
| |  | | --- | | **Loan Management** |  |  | | --- | | - Record book loans, due dates, and return dates.Notify members of due dates and overdue books. | | |  | | --- | | **Usability** |  |  | | --- | | - Provide a user-friendly interface for both staff and members. Ensure easy navigation and access to all functionalities. | |
| |  | | --- | | **Reservation Management** |  |  | | --- | | - Allow members to reserve books.Track and update reservation status. | | |  | | --- | | **Reliability** |  |  | | --- | | - Ensure the system is available and functional during library hours.- Regularly backup data to prevent loss. | |
| |  | | --- | | **Staff Management** |  |  | | --- | | - Add, update, and delete staff records. Assign roles and responsibilities to staff members. | |  |

Table 1.3 Functional and Non-Functional Attributes of the Library Database

## 1.3 Entity Relationship (ER) Diagram

Entity-Relationship (ER) diagrams are pivotal tools in database design, providing a visual representation of the data structure by illustrating entities, attributes, and relationships. Introduced by Peter Chen in 1976, ER diagrams help model the data and its interconnections, ensuring a logical and efficient organization of information (Chen, 1976). They are crucial for identifying data elements and their relationships, facilitating communication among stakeholders, and addressing potential issues like data redundancy and integrity constraints early in the design process (Elmasri & Navathe, 2016). Overall, ER diagrams serve as blueprints for creating robust databases that meet organizational needs and promote data integrity.

Figure 1.1 ER Diagram of the Library Database

From Figure 1.1 these informations can be extracted as following;

* One-to-Many (1:N) Relationship: One entity is connected to several instances of another entity through a one-to-many relationship.For example;
* Book to Loan: One book may be borrowed numerous times (many loans for one book).
* Member to Loan: One member, numerous loans; each member may have many book loans.
* Many-to-One (N:1) connection: A many-to-one connection is one in which a single instance of one entity is linked to several instances of another.For instance;
* Loan to Book: A single book is linked to numerous loans (many loans, one book).
* Required Relationships: A required relationship is one in which an instance of one entity needs to be connected to an instance of another. As examples;
* Loan to Book: There must be a required association between a loan and a book.
* Loan to Member: There has to be a mandatory relationship (association) between a loan and a member.
* Optional Relationships :
* An instance of one entity can be linked to zero or more instances of another entity using an optional relationship.
* Reservation to Book: There is an optional relationship between a reservation and a book.

## 1.4 Design Choices

* Book Entity: Identified by its unique primary key, BookID. A book's title, author, publisher, and year of publication are among the attributes that are selected to convey key details about it. These characteristics were selected because they are useful for many library functions, such as inventory management, searching, and classification.
* Member Entity: MemberID is used to identify it and acts as a primary key. Name, address, phone number, and email are examples of attributes. These are necessary for maintaining member communications and records.

Connections:

* Loan Relationship: Uses LoanID, BookID, and MemberID to establish a connection between Book and Member entities. This arrangement mimics what happens when a member checks out a book from the library. The addition of LoanDate, DueDate, and ReturnDate makes loan transactions easier to handle and supports the library's operating requirements.
* Reservation Relationship: Using ReservationID, BookID, and MemberID, this relationship establishes a connection between Book and Member entities, just like Loan does. Members can make reservations for books using features like ReservationDate and Status, which show the status of the reservation as of right now.
* Efficiency and Normalization:

The schema exhibits conformity to the following normalization principles:

Without duplication, each table (Books, Members, Loans, Reservations, Staff) fulfills a single function.To reduce data duplication and maintain data integrity, attributes are logically organized within entities.Referential integrity is ensured via foreign key restrictions (FOREIGN KEY), which guarantee the consistency of relationships between entities

* Flexibility and Scalability:

Scalability is enabled by the design, which means that adding additional books, members, reservations, loans, or staff entries doesn't require a major change to the current tables.

Foreign keys are used to create relationships, making it simple to expand the database schema as library activities change or expand over time.

* Performance and Usability:

Features are selected to maximize usability:

Title, Author, Publisher, Name, Address, Email, Position, Status, and other variable-length string types are supported by VARCHAR types, which can handle different data lengths that are commonly found in library records.

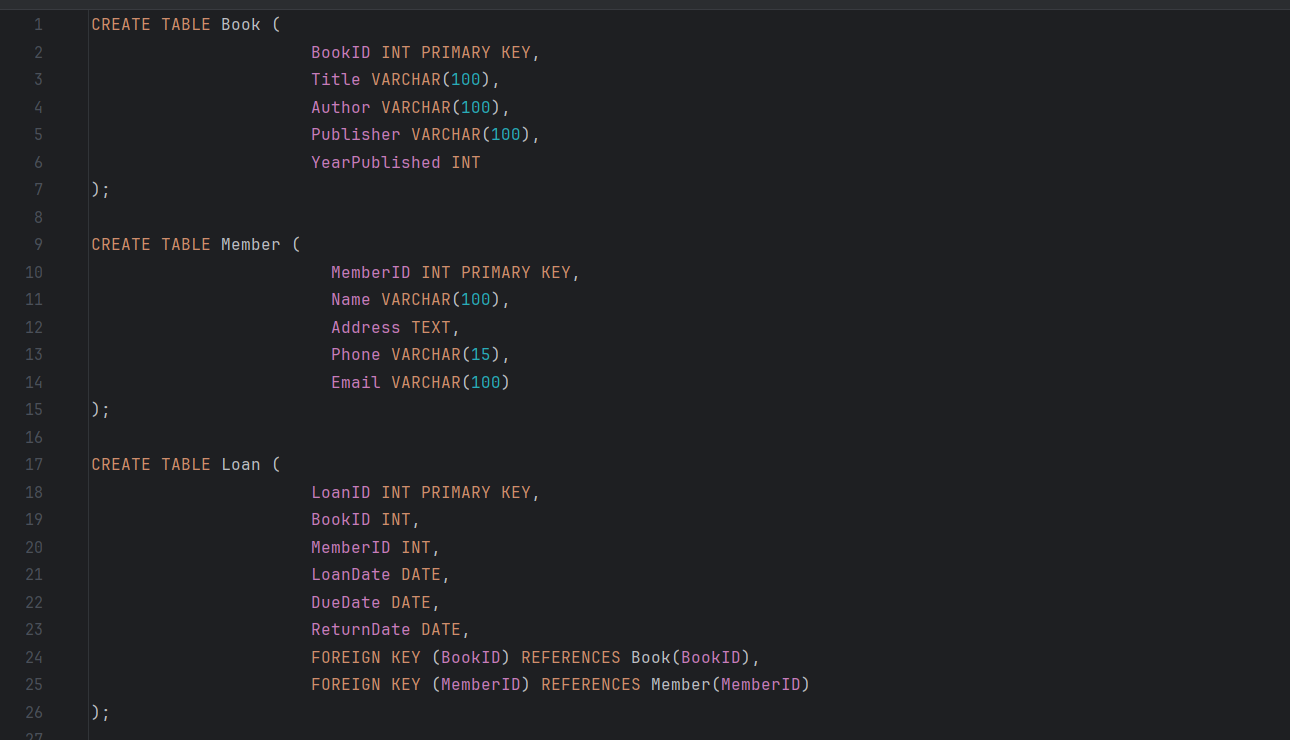
Numerical and date data are efficiently stored and retrieved with the help of INT types (BookID, MemberID, LoanID, ReservationID, YearPublished) and DATE types (LoanDate, DueDate, ReturnDate, ReservationDate).

In conclusion, the library's database design is set up to specifically handle the needs of staff, reservations, loans, books, and members.

# 2. IMPLEMENTATION

## 2.1 Creating Databases

To implement MYSQL code structure to obtain the library database creating tables , setting relationships , defining constraints steps must be followed.



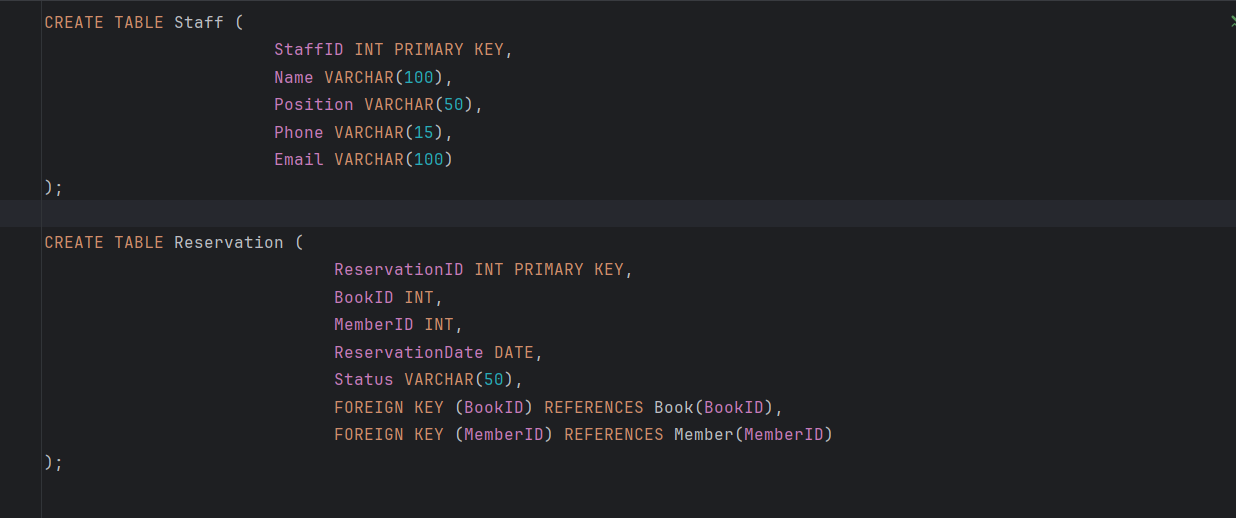


Figure 2.1 Creating the Library Database on Docker

|  |  |  |
| --- | --- | --- |
| Attrbutes | Subtitle | Explanation |
| Book | BookID (Primary Key), Title, Author, Publisher, YearPublished | Information about books stored in the library. |
| Member | MemberID (Primary Key), Name, Address, Phone, Email | Information about members who is able to borrow books from the library |
| Loan | LoanID (Primary Key), BookID (Foreign Key), MemberID (Foreign Key), LoanDate, DueDate, ReturnDate | Tracking loans of books to members, including loan dates and return status. |
| Staff | StaffID (Primary Key), Name, Position, Phone, Email | Information across staff members who manage the library. |
| Reservation | ReservationID (Primary Key), BookID (Foreign Key), MemberID (Foreign Key), ReservationDate, Status | Tracking reservations are made by members for books, including reservation status. |

Table 2.1 Desciptions of MYSQL Database

## 2.2 Providing Library Scenarios

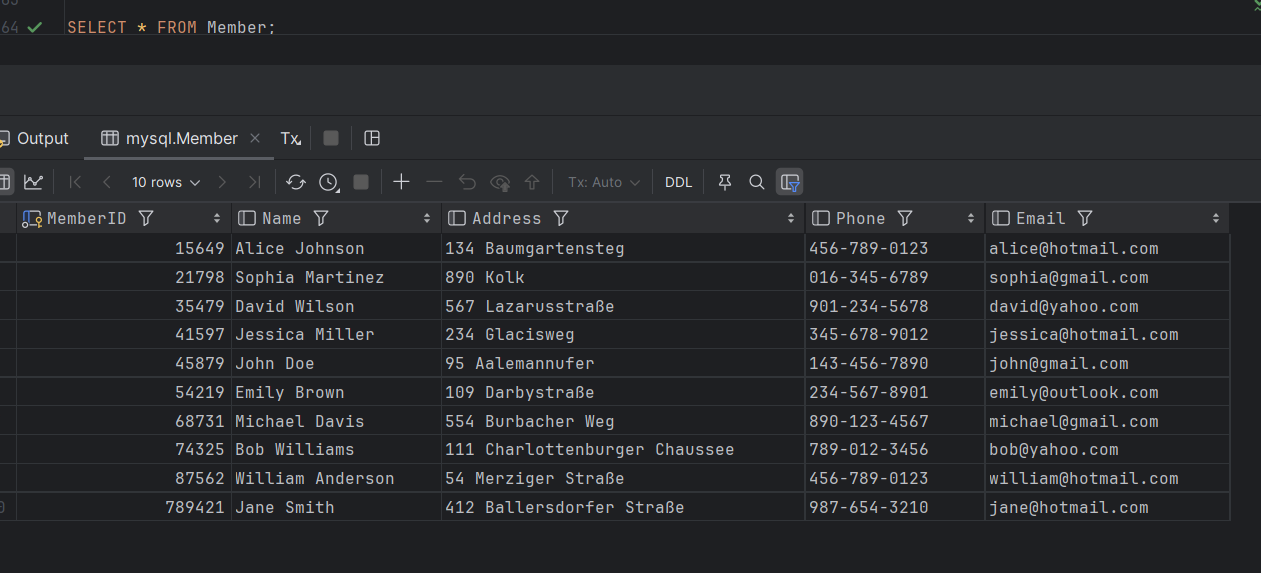
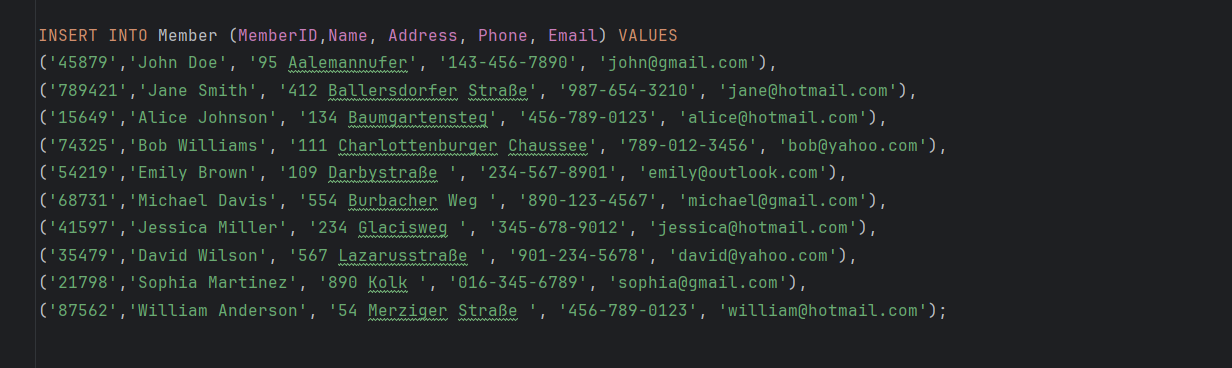
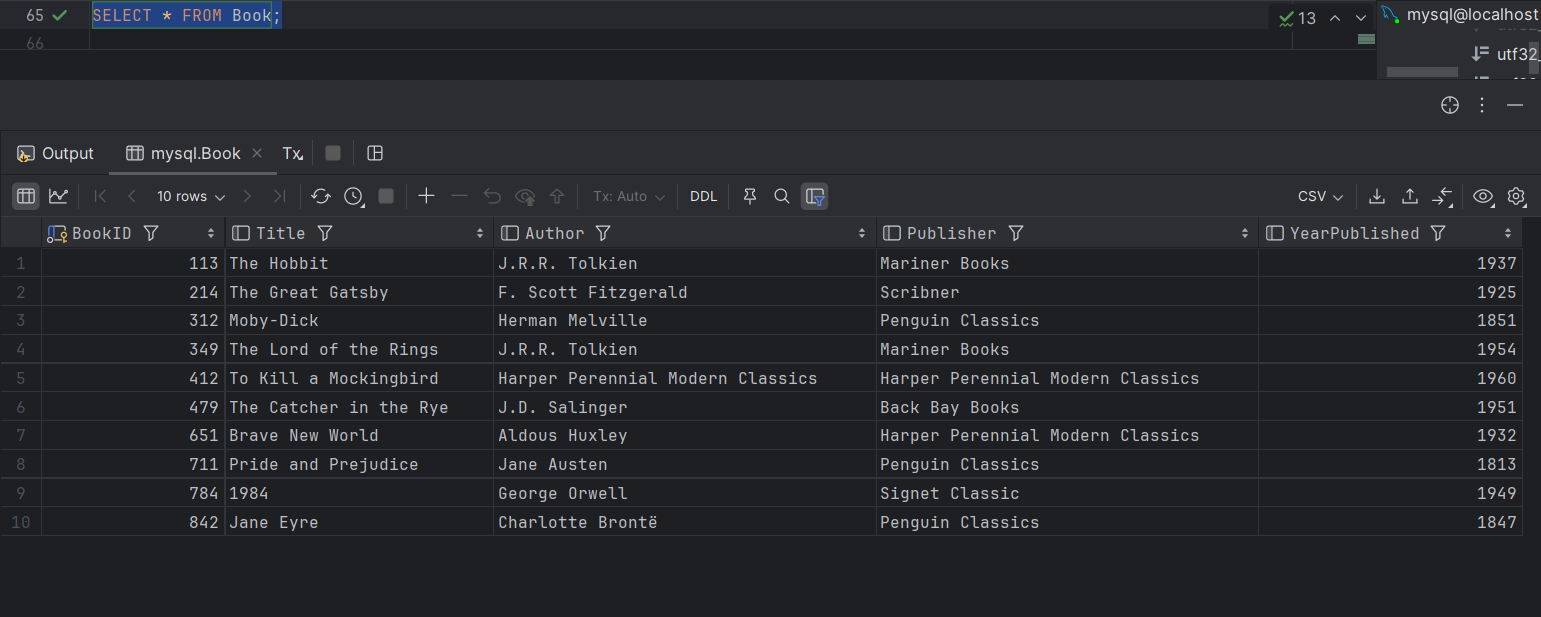
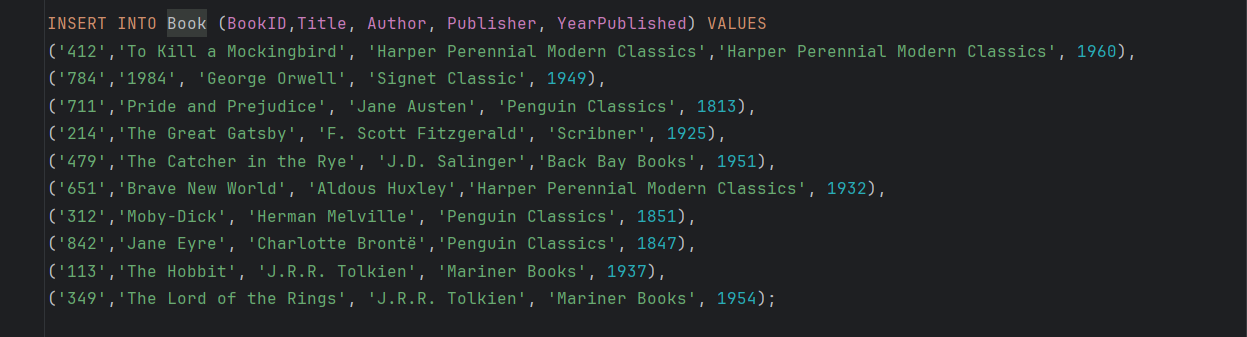
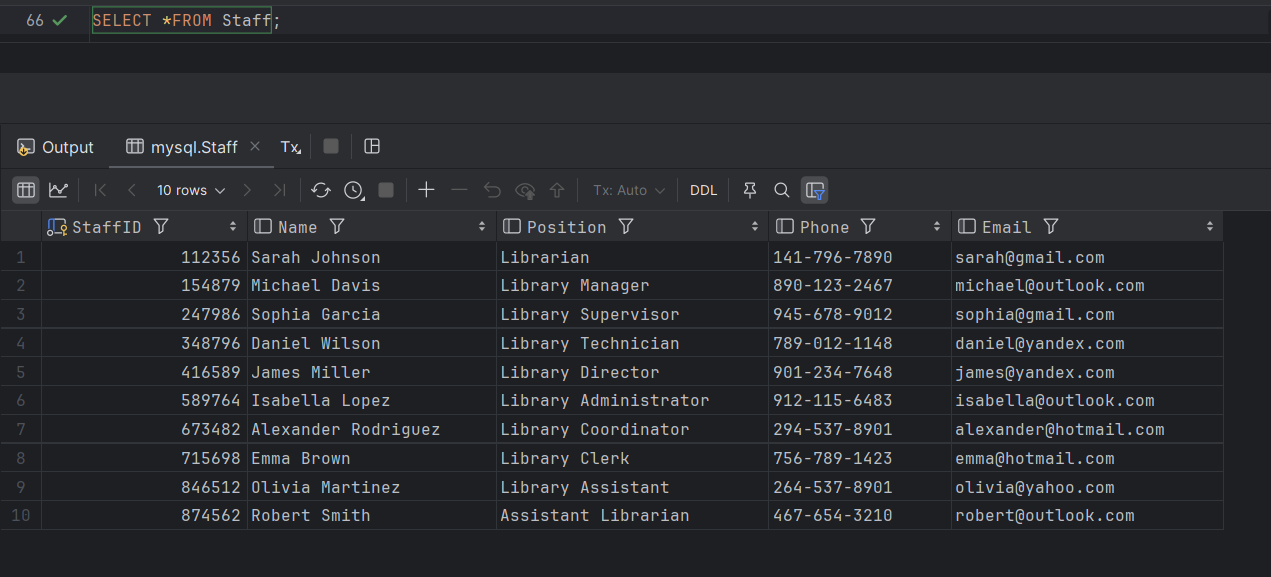
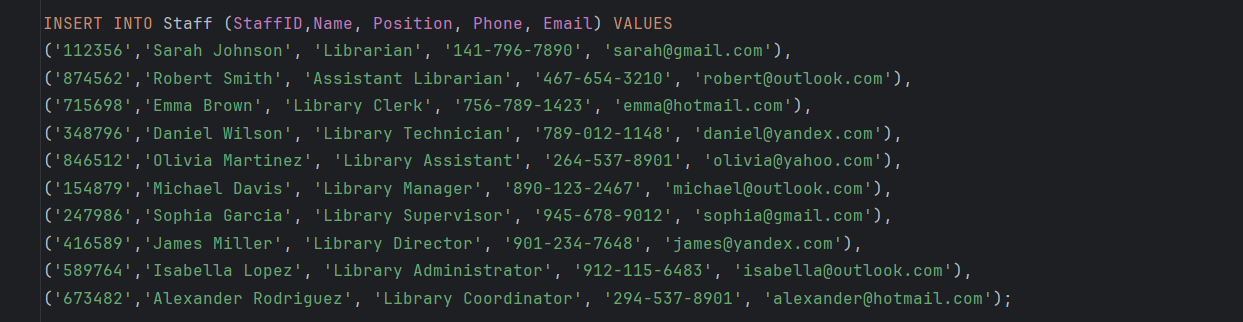


Figure 2.2 Inserting Members into the Dataset

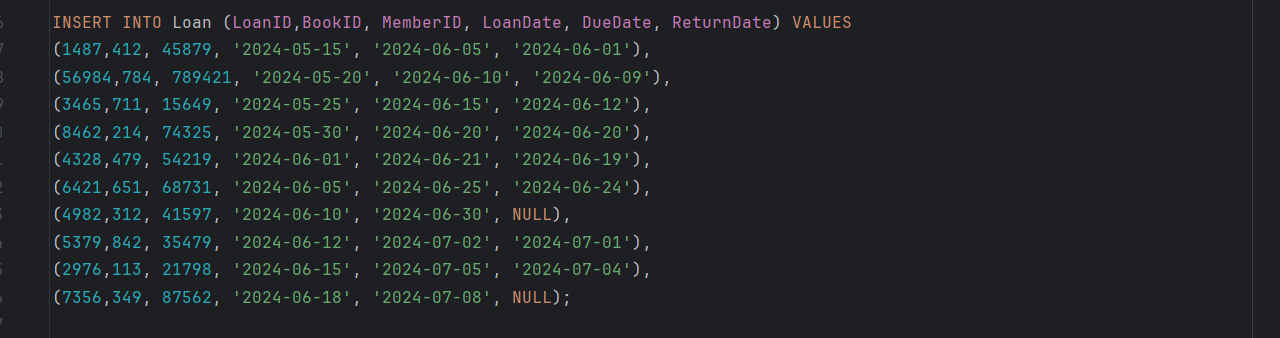
In first step we have been inserted related data into member then which is checked by using select statement to be sure.

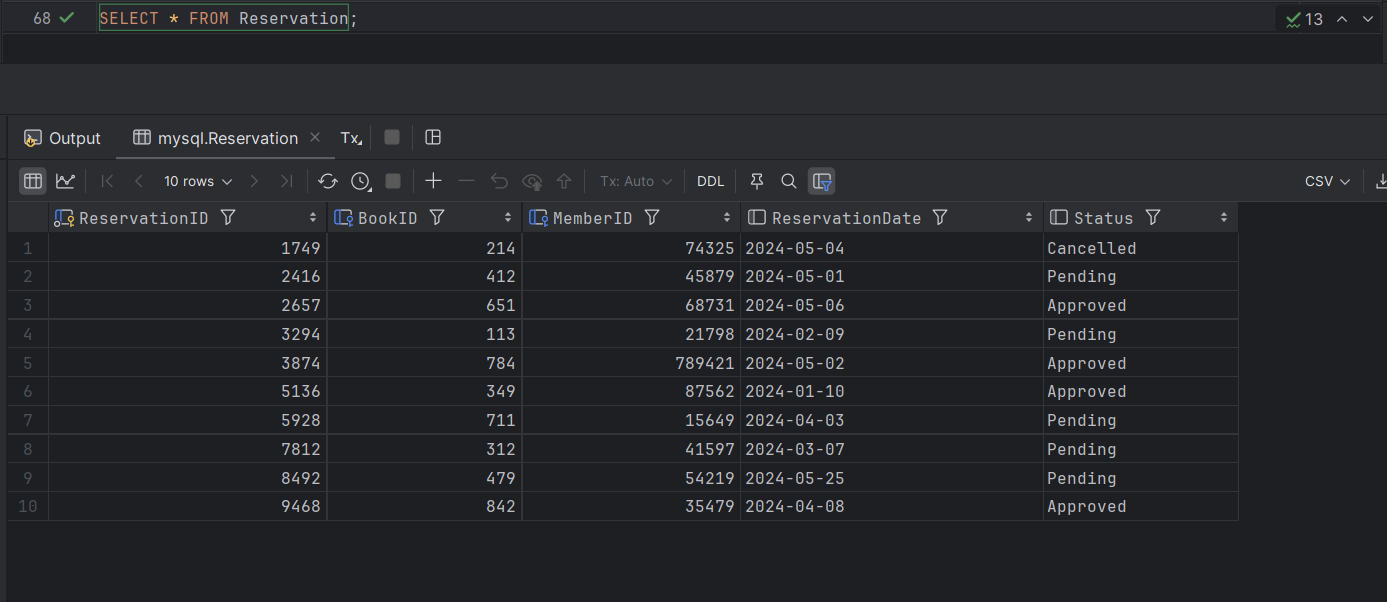
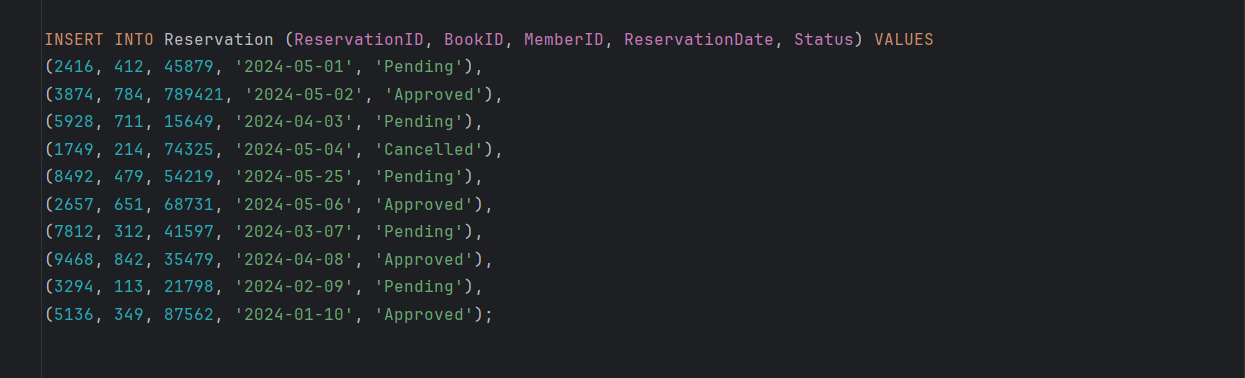
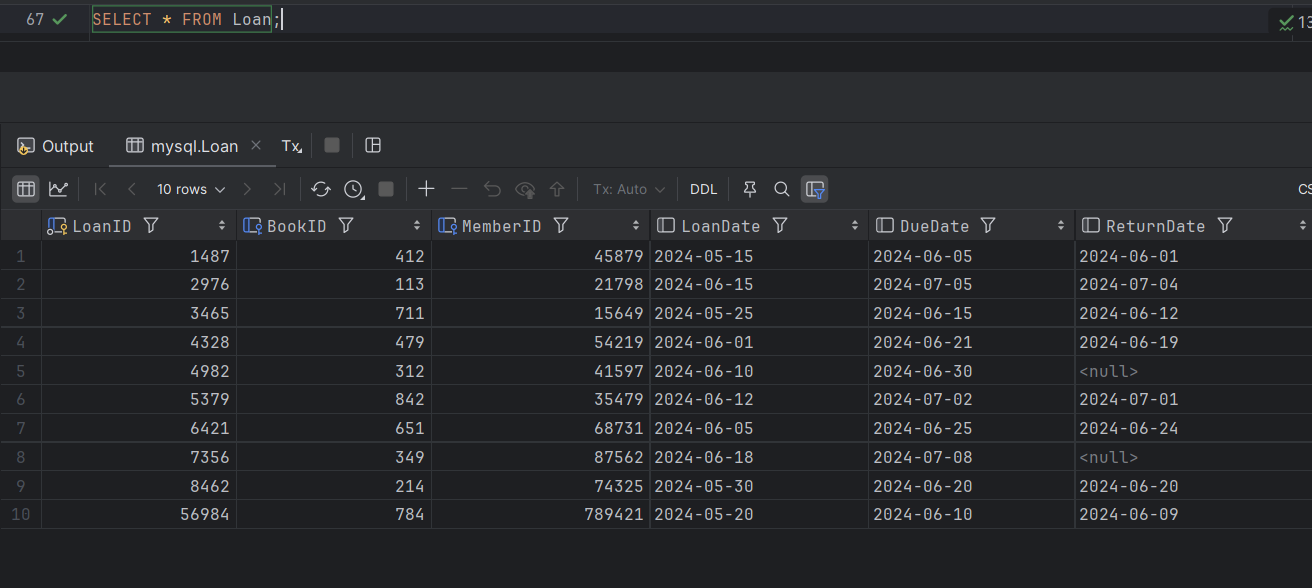
Figure 2.3 Inserting Book Details into the Dataset

In second step book details were inserted into the dataset based on its attributes like BookID,Title,Author,Publisher and PublishedYear.After that for to be sure select statement was used to demonstrate book details.

Figure 2.4 Inserting Staff Informations into the Dataset

In third step library staff details were inserted into the dataset based on its attributes like StaffID,Name,Position,Phone and Email.After that for to be sure select statement was used to demonstrate staff details.

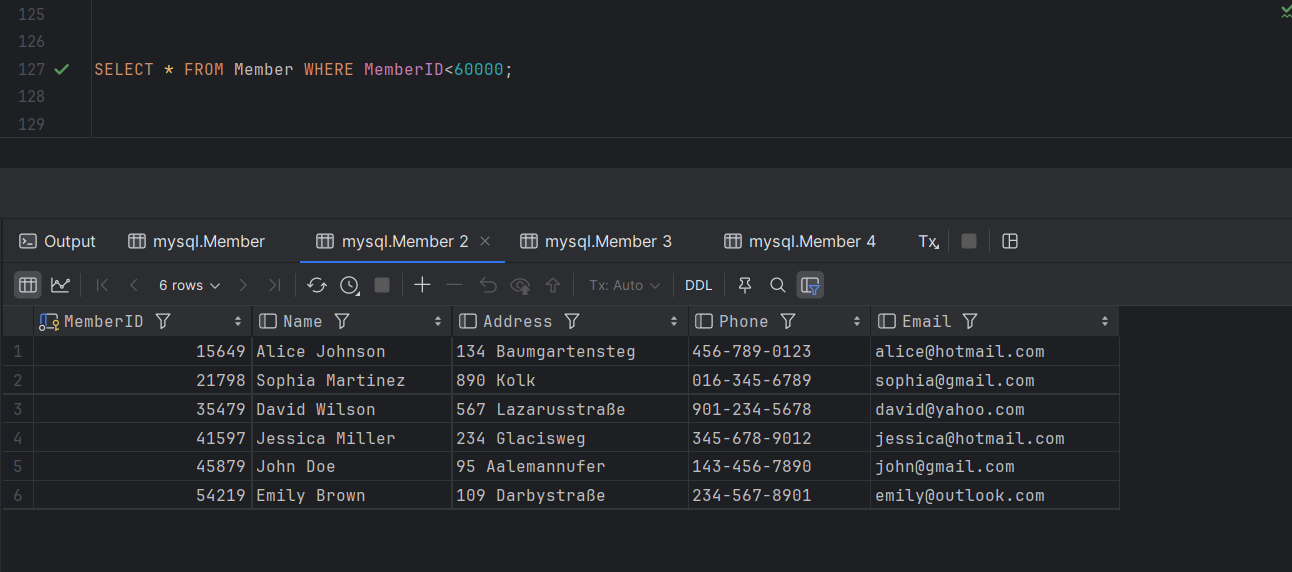
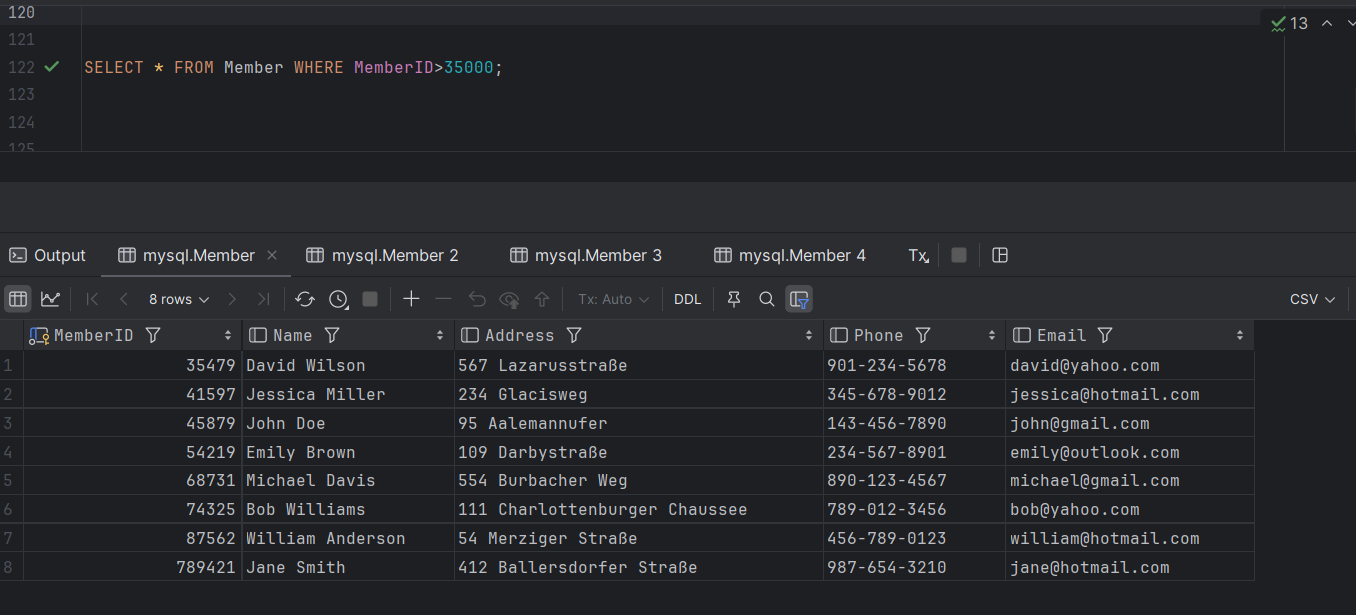


Figure 2.5 Inserting Loan - Reservation Details into the Dataset

In final step loan reservation details were inserted into the dataset based on their attributes like LoanID,BookID,MemberID,ReservationID,Dates and Status .After that for to be sure select statement was used to demonstrate these details.

## 2.2 SQL Queries

In this part select statements and join applications will be used based on various parameters.



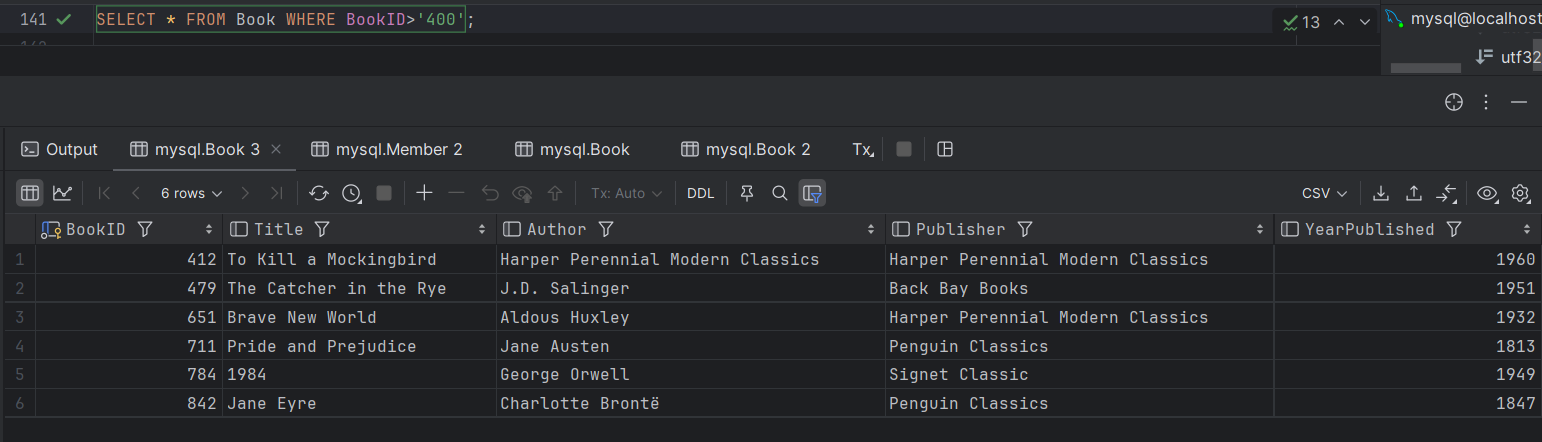
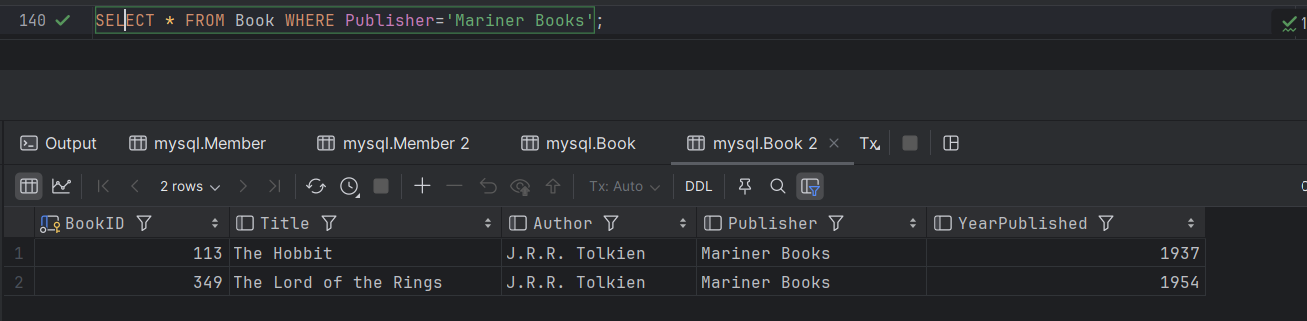
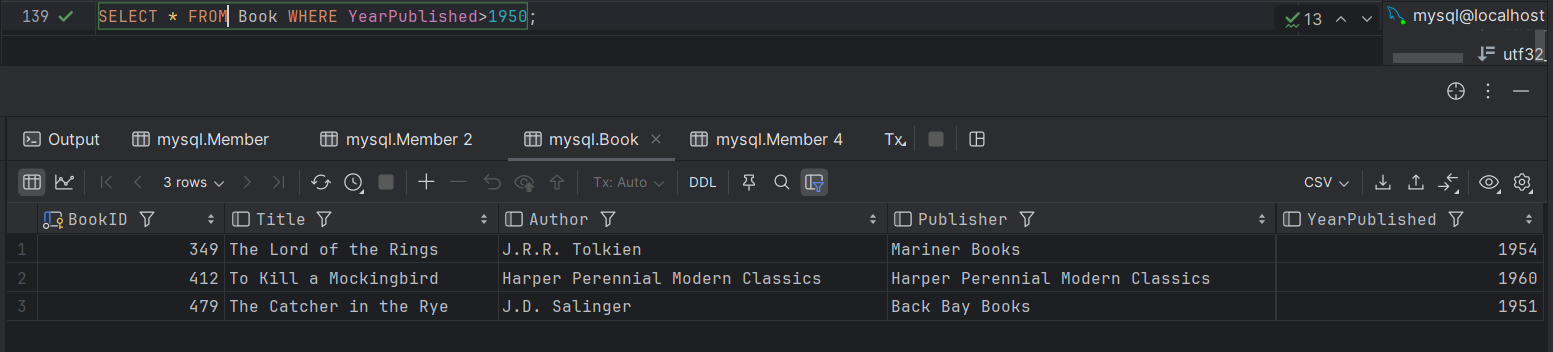
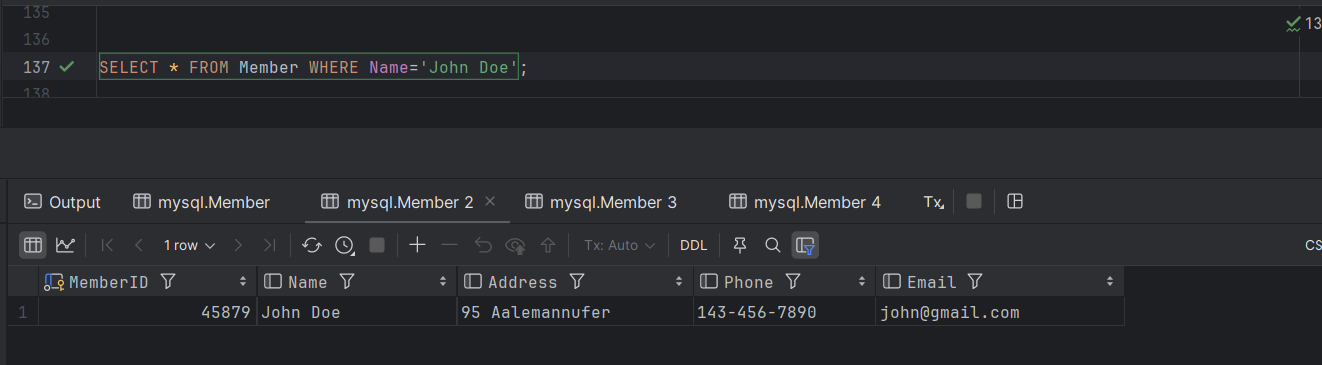
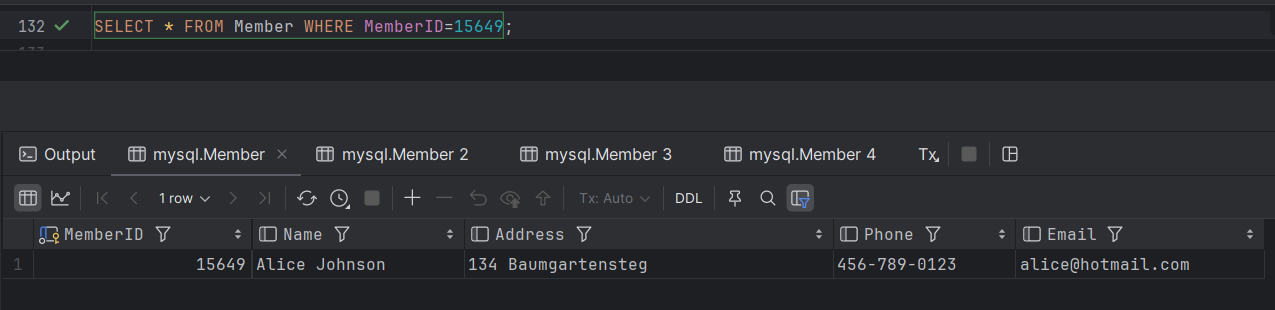
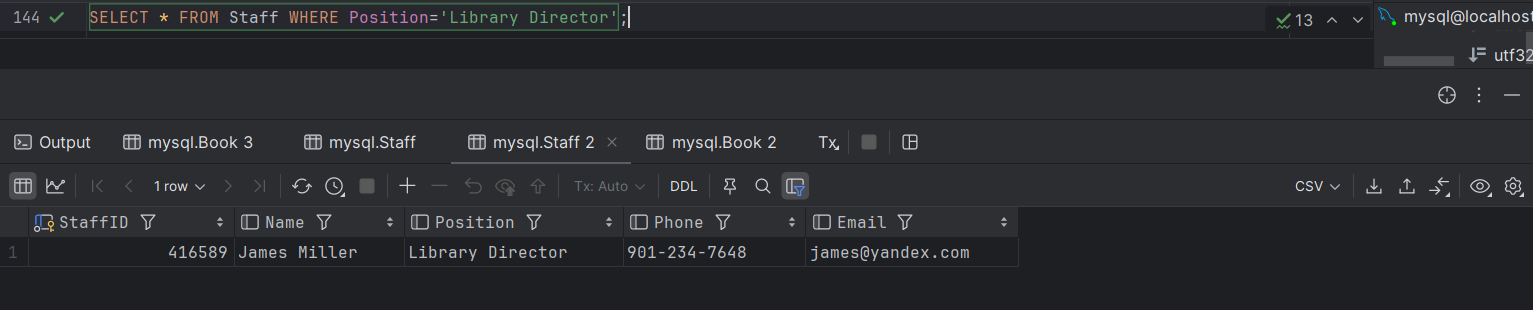
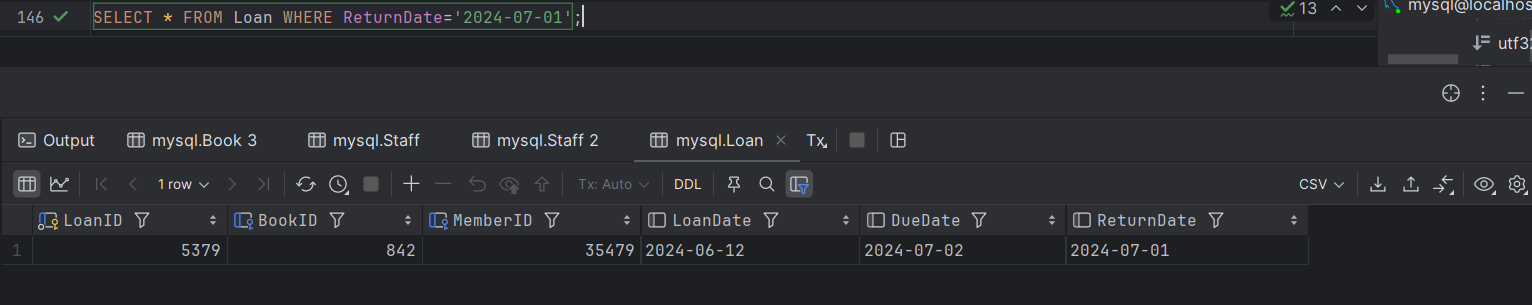
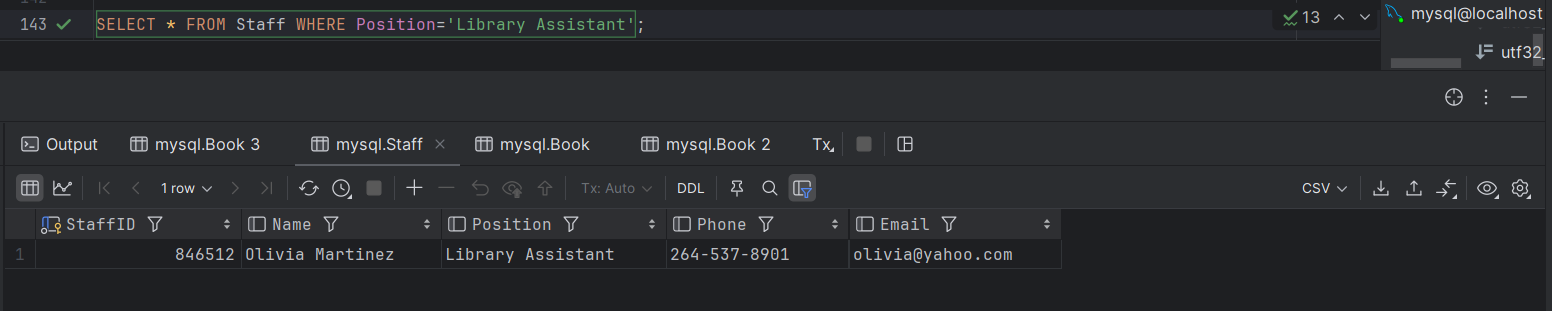
  

Figure 2.6 Select Statements to Explore the Dataset

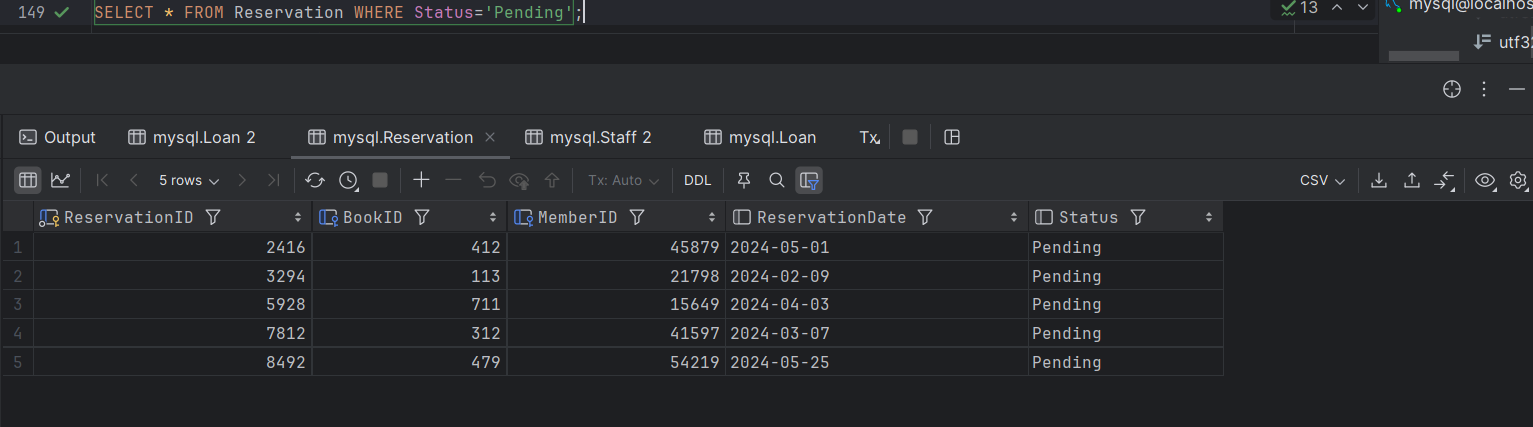
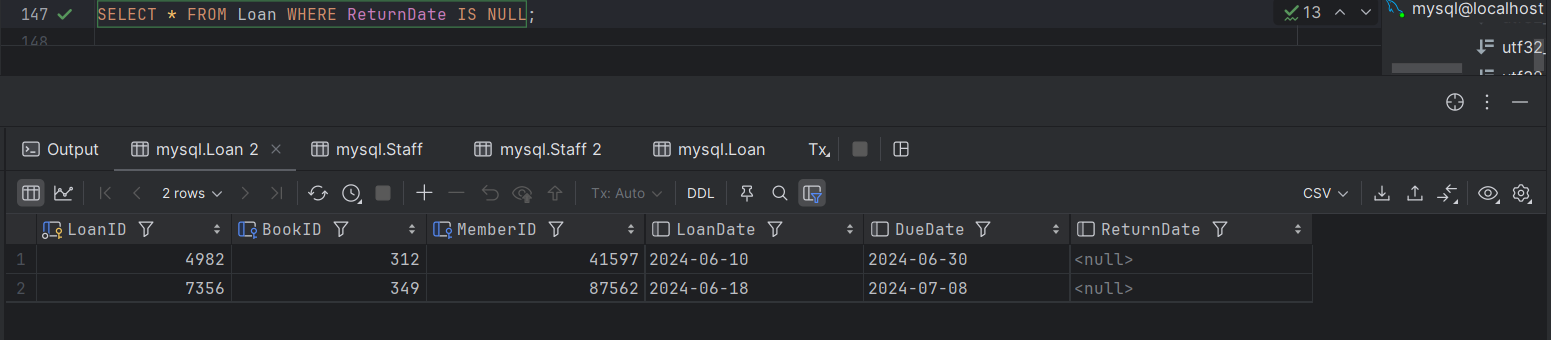


Figure 2.7 Select Statements to Explore the Dataset

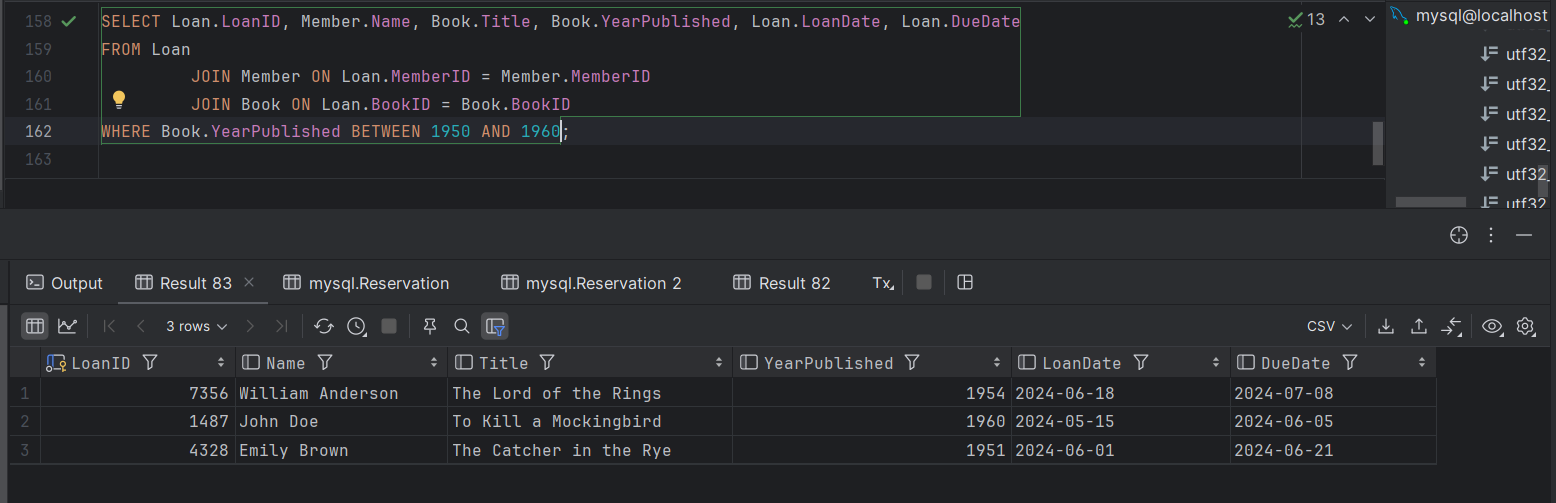
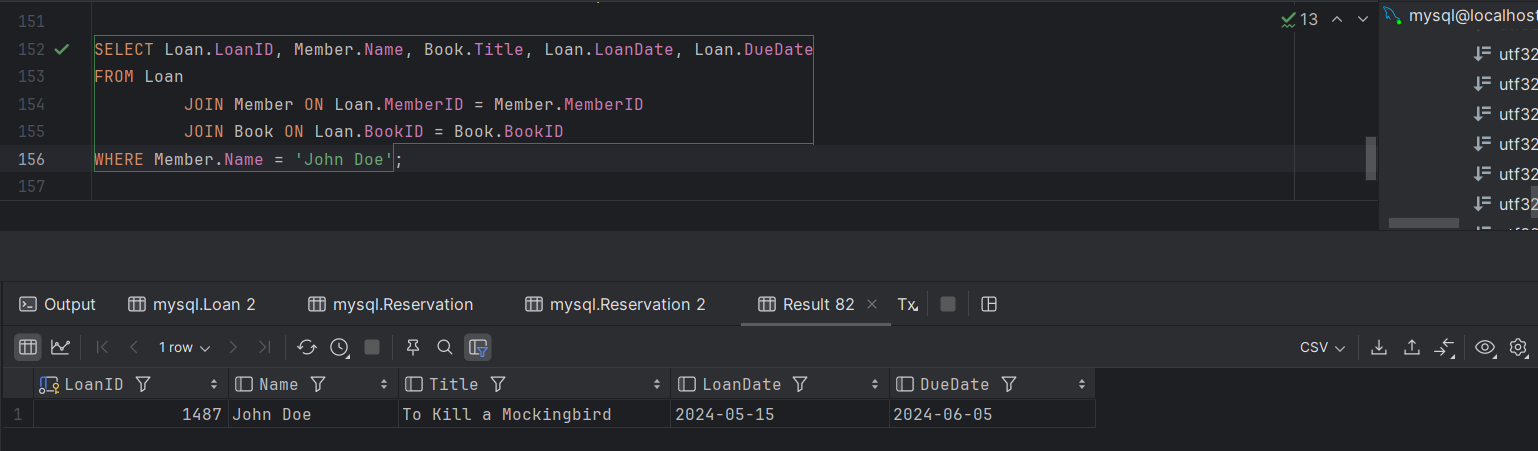


Figure 2.8 Join Operations

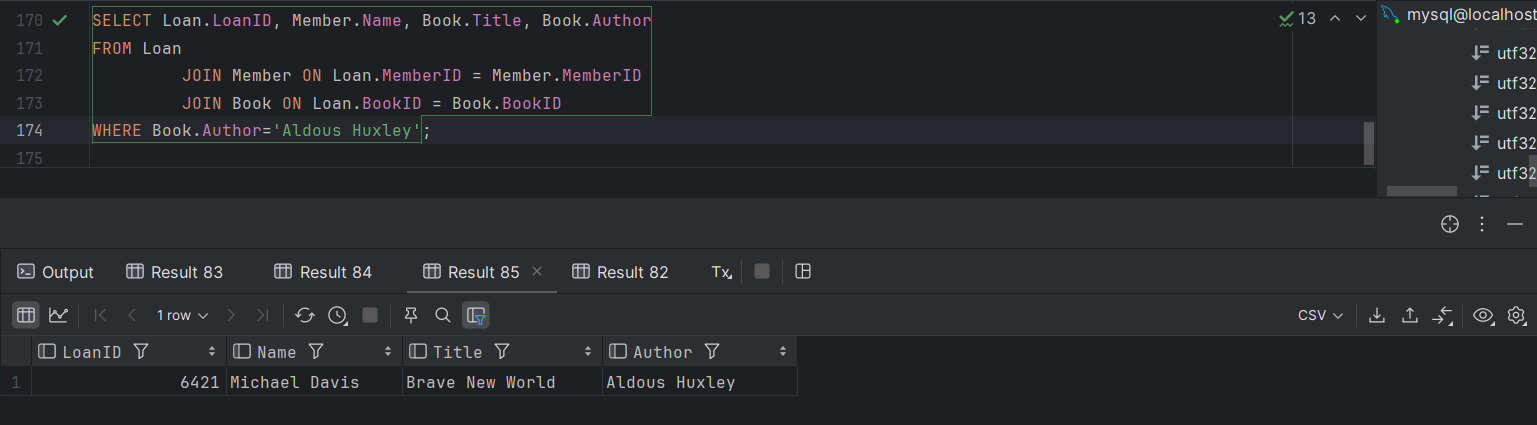
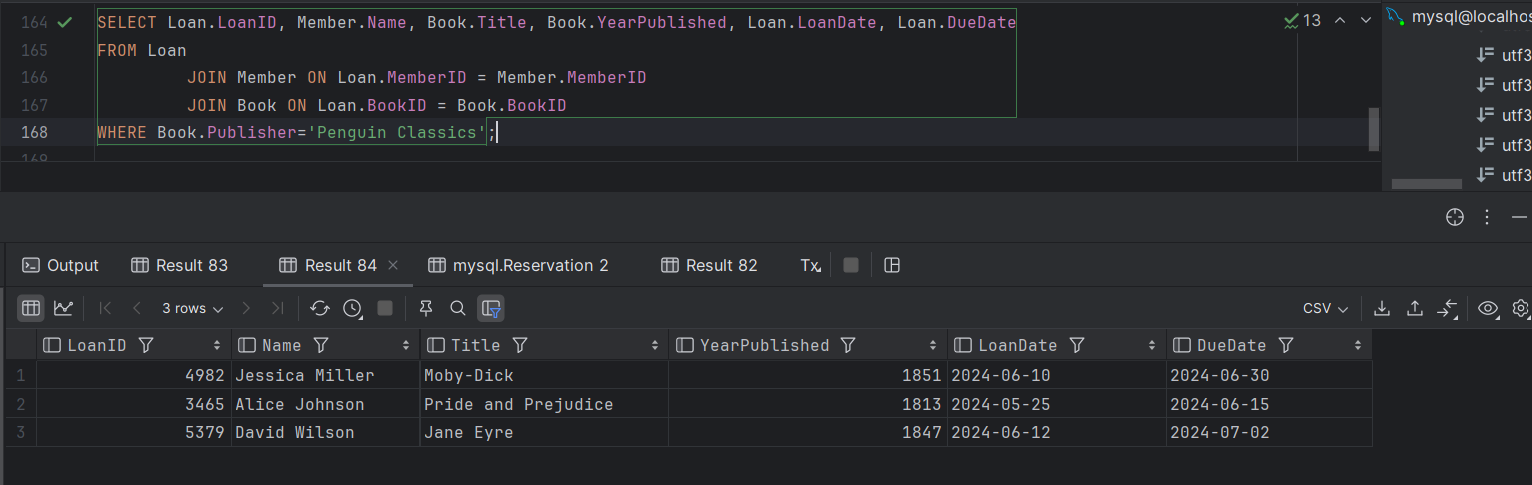


Figure 2.9 Join Operations

As Figure 2.8 indicates we can find out which person borrow which book also based on specific dates which books are borrowed by who.Moreover Figure 2.9 implies that borrowed books can be filtered based on author as well as to able to learn member details at the same time this exploration can be done across author.

# 3. CAP THEOREM

## 3.1 General Review of CAP Theorem

An essential principle of distributed database systems, the CAP theorem was first proposed by Eric Brewer in 2000. It argues that a system can only ensure two of the three criteria listed before: consistency, availability, and partition tolerance (Brewer, 2000). When creating a database system for a library, it is essential to comprehend the CAP theorem in order to properly balance these attributes.

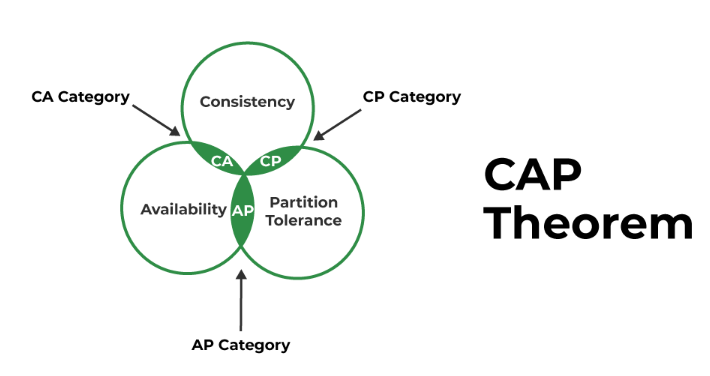


Figure 3.1 Illustration of CAP Theorem as Venn Schema

* Consistency

In a distributed database, consistency guarantees that every node reflects the same data simultaneously. This means that in a library system, when a member checks out a book, the change should replicate to all database replicas instantaneously. Maintaining uniformity is essential to avoid problems such as making two reservations for the same book or having inconsistent loan and return records.

However, achieving strict consistency can introduce latency, as the system must ensure that all replicas are synchronized before confirming a transaction.

* Availability

Availability guarantees that, even in the event that some nodes go down, every query sent to the database will be answered, whether it succeeds or fails. This means that, in the case of a library, the system ought to let users to look up books, schedule appointments, and check them out whenever they like, avoiding network or server interruptions. Since high availability guarantees that library services are always available, it is essential for customer satisfaction. Nevertheless, maintaining high availability may occasionally result in consistency issues, particularly when a network partition occurs.

* Partition Tolerance

Partition tolerance describes the system's capacity to carry on operating even in the event of a network partition, which prevents some system components from communicating with one another. Partition tolerance is necessary for a database system in a library to continue functioning even in the event of a network outage or interruption. Libraries with several branches or those whose systems must continue to function even in the event of a temporary loss of branch connectivity should pay special attention to this.

* Balancing CAP Theorem

Based on the unique needs and goals of the library, a balance between consistency, availability, and partition tolerance must be achieved in the context of the database system of the library. For example:

* High Consistency and Partition Tolerance (CP): For libraries where exact book inventory tracking is crucial, maintaining current and correct records at the price of availability during network outages may be essential.
* High availability and Partition tolerance (AP): Libraries that place a high priority on user access and service continuity may find it appropriate to ensure that the system maintains functionality and responsiveness in the event of a network failure, sometimes at the sacrifice of instant consistency.

The trade-offs that must be taken into account when developing and overseeing distributed systems are highlighted by the CAP theorem. Library database designers can develop a system that precisely matches the requirements of their users and operating environment by carefully weighing these trade-offs (Brewer, 2012).

## 3.2 Importance of Three Components for Lbrary and Database Design – Relationship of the Components with Library Database

The CAP theorem emphasizes the necessity of balancing consistency, availability, and partition tolerance based on particular operational goals and limitations while developing and implementing a library's database system. The CAP theorem has the following effects on decision-making:

1. Prioritizing Consistency (CP): The system would be built to make sure that all data changes are reliably propagated to all nodes before verifying transactions, if the library places the utmost priority on data quality and integrity. Strict transactional protocols and distributed locks may be used in this situation to ensure accurate records while reducing system availability during network outages.
2. Prioritizing Availability (AP): The system would be built to process requests even during network splits, acknowledging that there may be some brief inconsistencies. This would be the case if the library placed a high priority on user access and ongoing functioning. The use of eventual consistency techniques, which propagate updates asynchronously and guarantee high availability but run the risk of stale data during partitions, may be necessary to achieve this.
3. Balancing All Three Components: In actuality, most library systems require a balanced strategy. To preserve partition tolerance and reach reasonable consistency and availability, they may combine several approaches. For instance, while utilizing eventual consistency models, high availability and partition tolerance can be guaranteed.Conflict resolution mechanisms to handle inconsistencies once partitions are resolved (Vogels, 2009).

# CONCLUDING REMARKS

In designing a library database, striking a balance among the CAP theorem's components—Consistency, Availability and Partition Tolerance is critical. Each component provides various benefits. For instance prioritizing consistency ensures accurate and up-to-date records of book loans and member of activities, which is important across maintaining operational efficiency. Nevertheless this might come at the expense of system availability during network partitions. Conversely, prioritizing availability is able to provide uninterrupted access to library services, enhanced user experience, but might lead to temporary inconsistencies during network issues .

Ultimately, the design and implementation of a library's database system must be carefully considered the library's specific demands and operational targets by related specialist. By understanding and applying the CAP theorem, library database specialists can make informed decisions which sets balance between consistency, availability, and partition tolerance to create a reliable, and user-friendly system. This approach not only supports efficient library operations but also enhances the overall service quality performance .

# BIBLIOGRAPHY

1. Brewer, E. A. (2000). Towards Robust Distributed Systems. Proceedings of the Annual ACM Symposium on Principles of Distributed Computing.
2. Brewer, E. A. (2012). CAP Twelve Years Later: How the "Rules" Have Changed. Computer, 45(2), 23-29.
3. Chen, P. P. (1976). The Entity-Relationship Model—Toward a Unified View of Data. ACM Transactions on Database Systems, 1(1), 9-36.
4. Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson.
5. Date, C. J. (2004). An Introduction to Database Systems. Addison-Wesley.
6. Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems. Pearson.
7. Gilbert, S., & Lynch, N. (2002). Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. ACM SIGACT News, 33(2), 51-59.
8. Hoffer, J. A., Ramesh, V., & Topi, H. (2016). Modern Database Management. Pearson.
9. Vogels, W. (2009). Eventually Consistent. Communications of the ACM, 52(1), 40-44.

# APPENDIX (if necessary)

CAP : Consistency – Availability – Partition Tolerance

ER : Entity Relationship

SQL : Structured Query Language