



CCE104 – Fundamentals of Database

# Library Borrowing and Returning Management System

First Term, First Semester SY 2025-2026

Submitted by:

Lourde Vincent M. Pablo

Submitted to:

Jenny Espadero

# Chapter 1

## A. Introduction

The business entity in this study is the Learning and Information Center (Library) of University of Mindanao. The library provides services such as borrowing and returning of books, assisting students in research, and maintaining records of available learning materials. Disclaimer: The university has no prior knowledge that the researcher utilized the LIC as a reference for this project. The information presented is solely derived from the researcher's personal experiences in conducting transactions within the university's LIC.

Currently, transactions involve students borrowing books, staff processing checkout/return, and tracking due dates. A book that is currently borrowed (not yet returned) cannot be checked out again by another member until it is returned. Each process includes important details such as the student ID, book barcode/code, transaction date, and due date. However, the problematic area lies in the manual recording of transactions. At times, it is difficult to monitor which books are borrowed, who borrowed them, and whether they are returned on time. Delays, misplaced records, and lack of automated reminders for overdue books are common issues. (Marlindawati ., Misinem ., & Adhiatma, 2024)

Hence, the solution proposed in this study is intended to address the issues faced by a non-sectarian college—St. John Paul II College of Davao—that still processes book borrowing and returning manually. The researcher personally experienced this situation, having been a former student at the said college. In contrast, the University of Mindanao Library and Information Center (UM LIC) have already implemented an automated system. With the increase of book collections and the expansion of readers' scale in college libraries, how to improve the efficiency of borrowing and returning books has become an important topic in the operation and management of college libraries (Mengping, 2023).

The problem to be solved is the inefficient tracking of book borrowing and returning, which can lead to loss of books, delayed returns, and inaccurate inventory. The proposed database solution is a Library Management Database System that will:

- ✓ Store student and staff information.
- ✓ Record book details using barcode numbers.
- ✓ Manage checkout and return transactions.
- ✓ Track due dates and generate reports.

This system will improve accuracy, reduce errors, and make retrieval of information faster and more reliable (Song, 2018).

The scope of this system is limited to the physical borrowing and returning of books in the library. Based on an interview with one of the staff members at UM LIC, the book category in the borrowing process is not considered essential.

The primary objective of the borrowing process is to record which book a member wishes to borrow, along with the corresponding due date and any possible penalties that may arise from late returns (Andri, Sopiah, & Oktaviani, 2025).

B. Business Rule

- A **member** must be registered in the library before performing any borrow transaction.
- A **member** may borrow **zero, one, or many books** over time, but each **borrow transaction** must be linked to exactly **one member**.
- A **borrow transaction** must be processed by a **staff** (librarian). A **staff** may process many transactions, but a transaction is linked to only one **staff**.
- A **book** may be borrowed multiple times, but at any given transaction, it may only be linked once. Some **books** may never be borrowed, but if a **borrow detail** exists, it must be linked to one book.

- A **borrow transaction** may include **one or many books**, but it cannot exist without at least one book.
- Each **borrow details** must be linked to exactly **one borrow transaction** and exactly **one book**.
- A **borrow transaction** must record the following information:
    - Transaction ID
    - Member ID
    - Staff ID
    - Date Borrowed
    - Due Date
    - Date Returned

## Chapter 2

### Entity Relationship Diagram

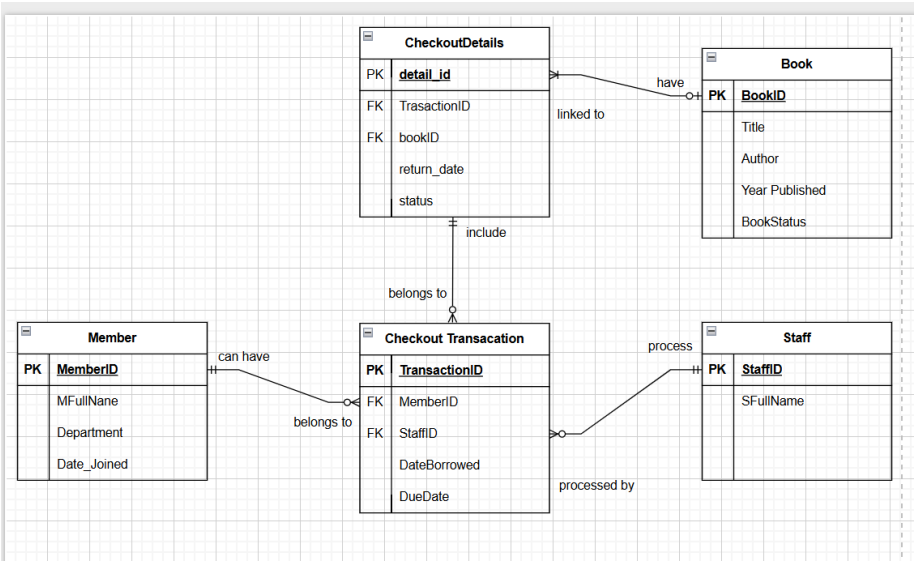



*Figure 1: Library borrowing and returning System ERD*

## Logic Design

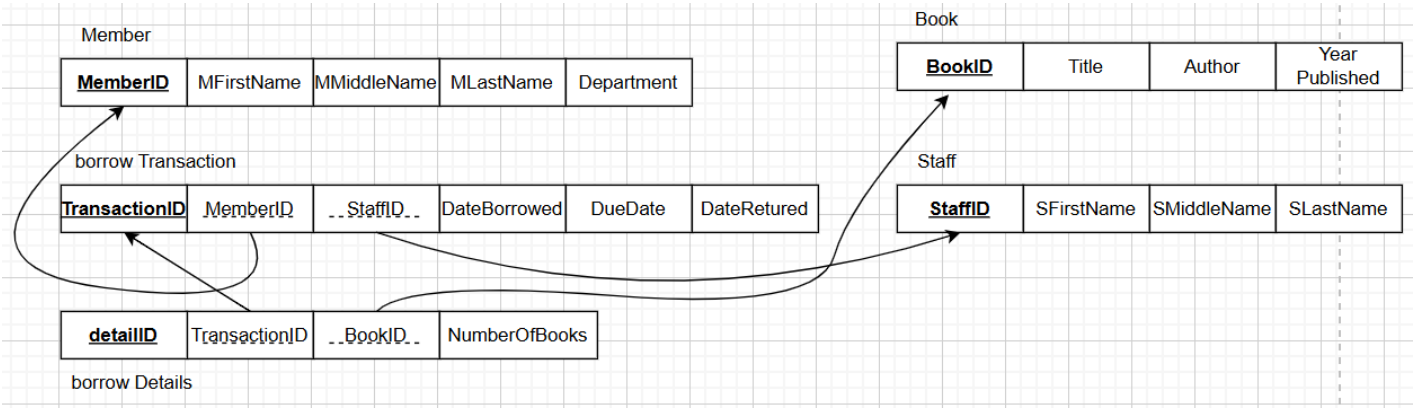



*Figure 2: Library borrowing and returning System Mapping Relations*

## Normalization

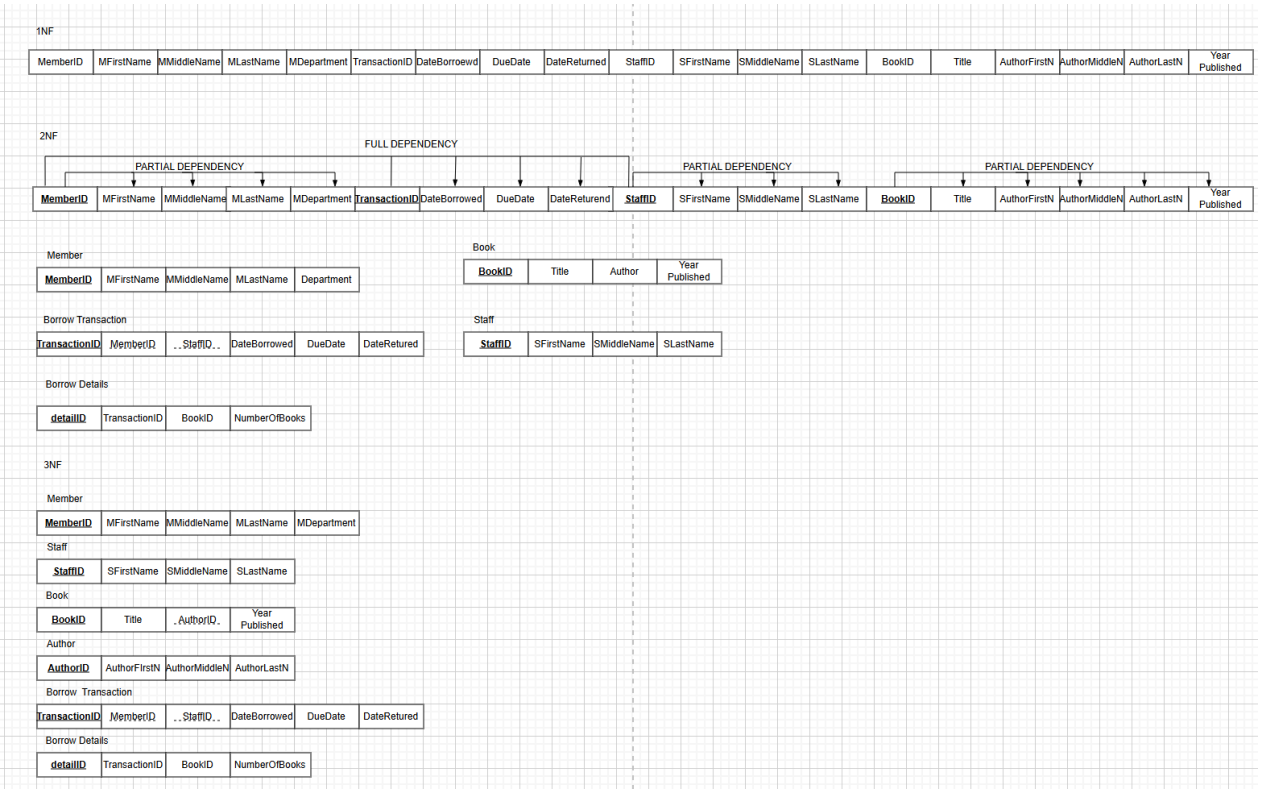



*Figure 3: Library borrowing and returning System Normalization*

# Chapter 3

## Physical Designs

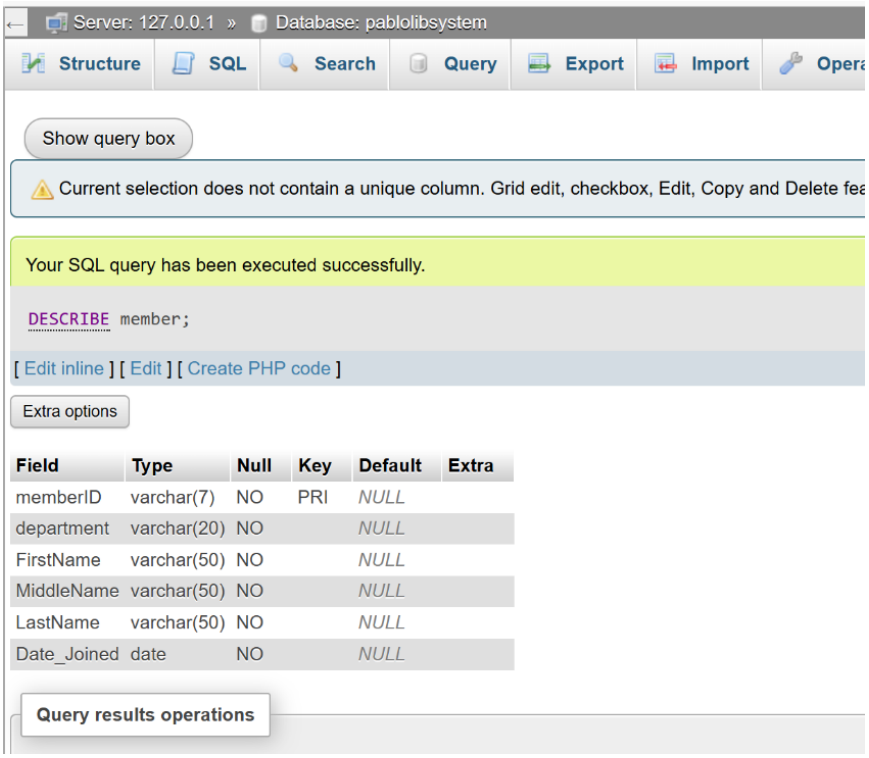
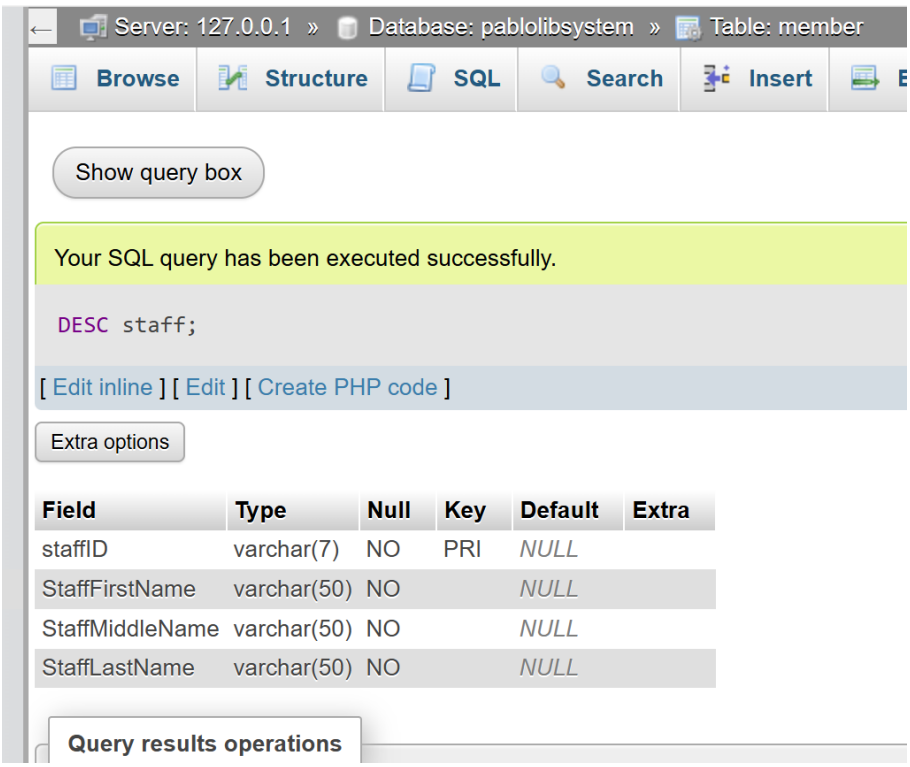



Figure 4: Member Table



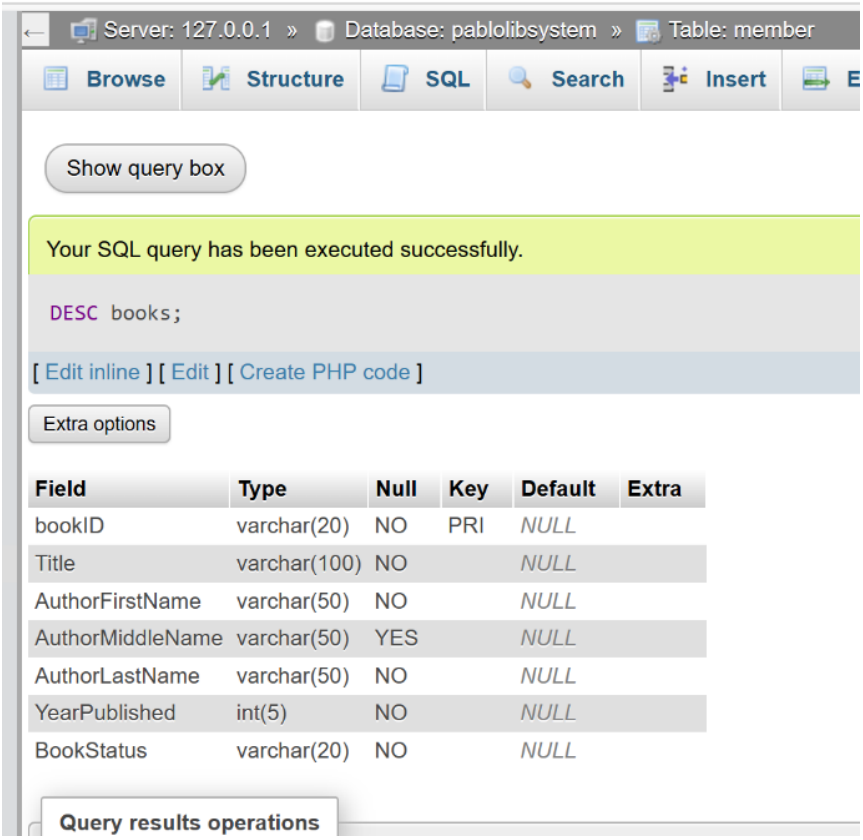



Figure 6: books Table

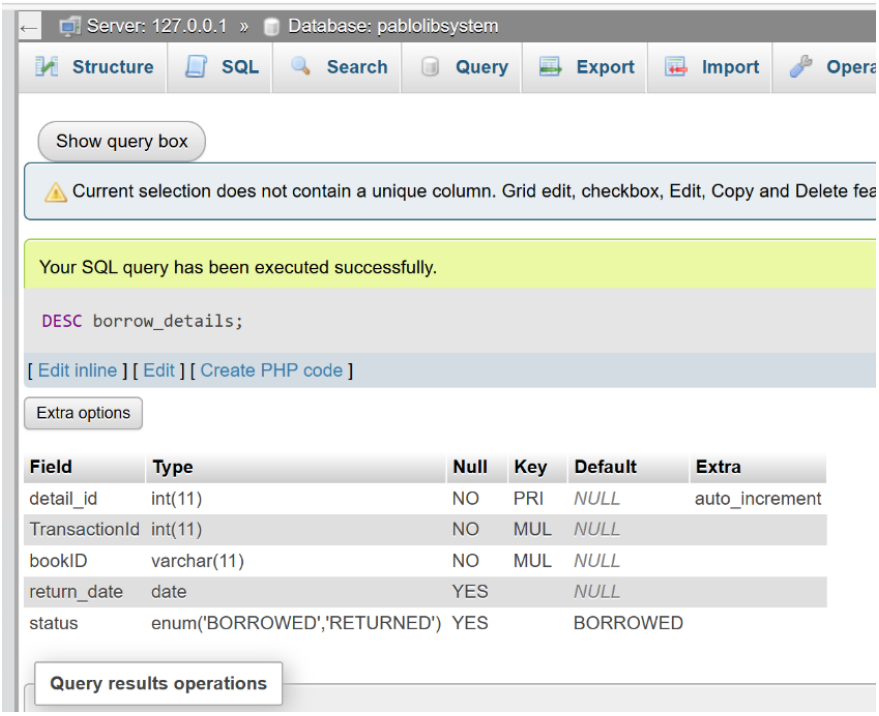


Figure 7: borrow_details Table

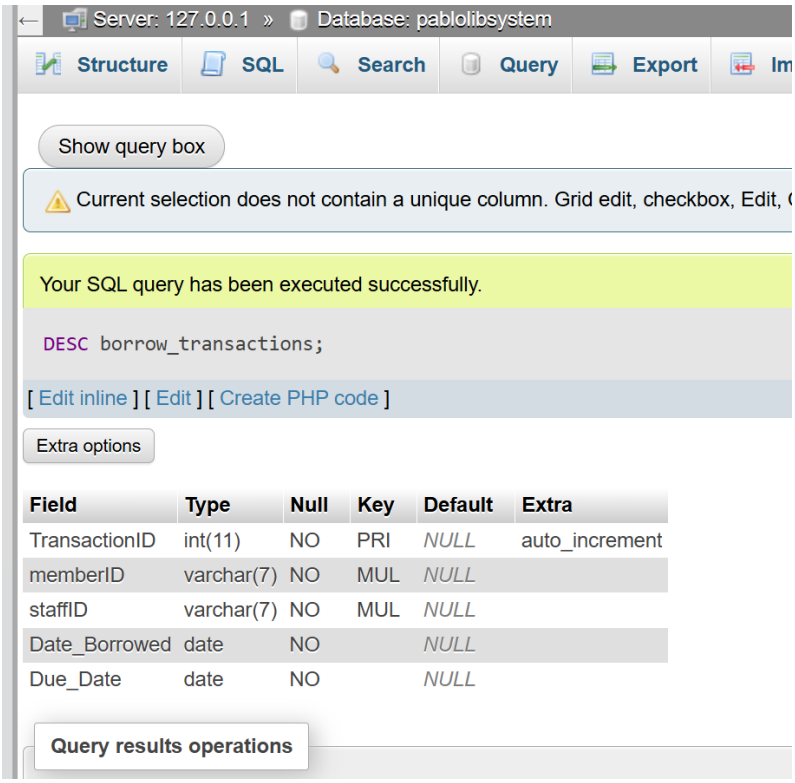



Figure 8: borrow_transactions Table

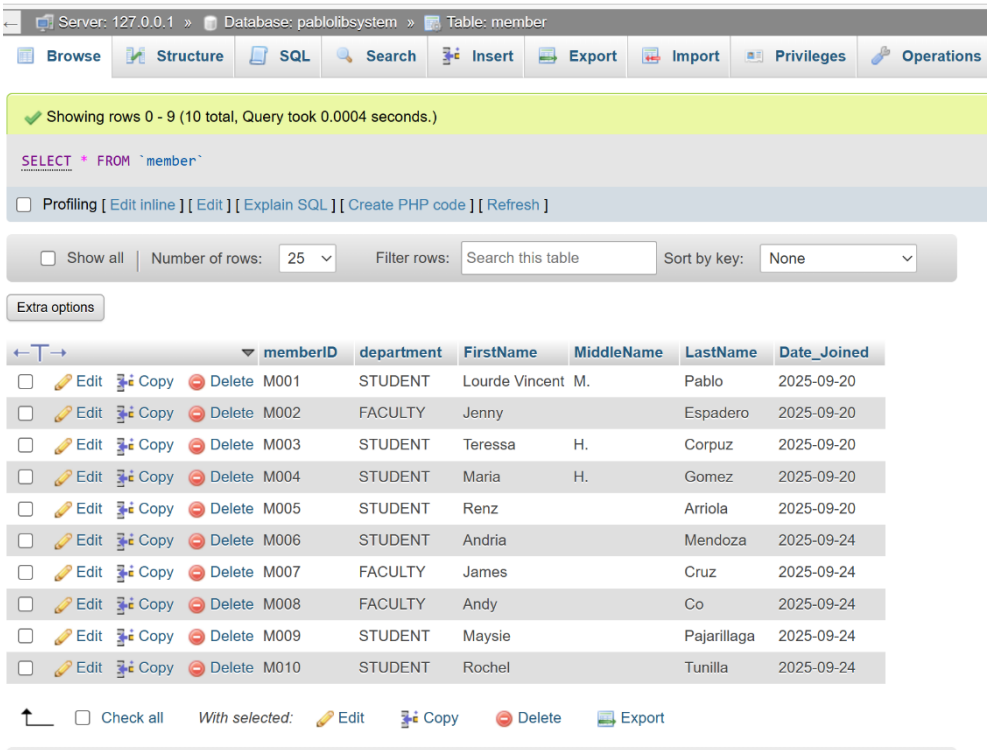



Figure 9: Member Records

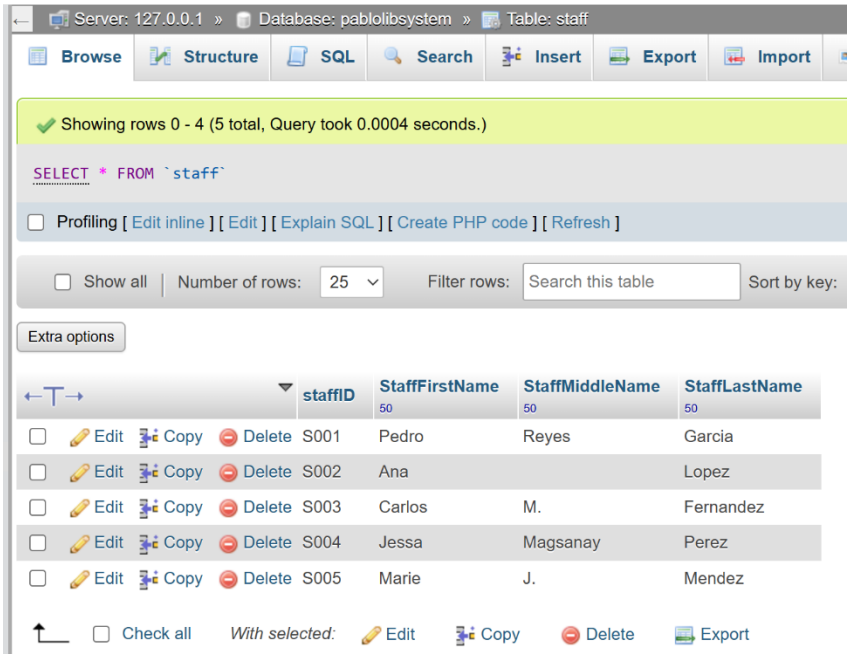



Figure 10: Staff Records

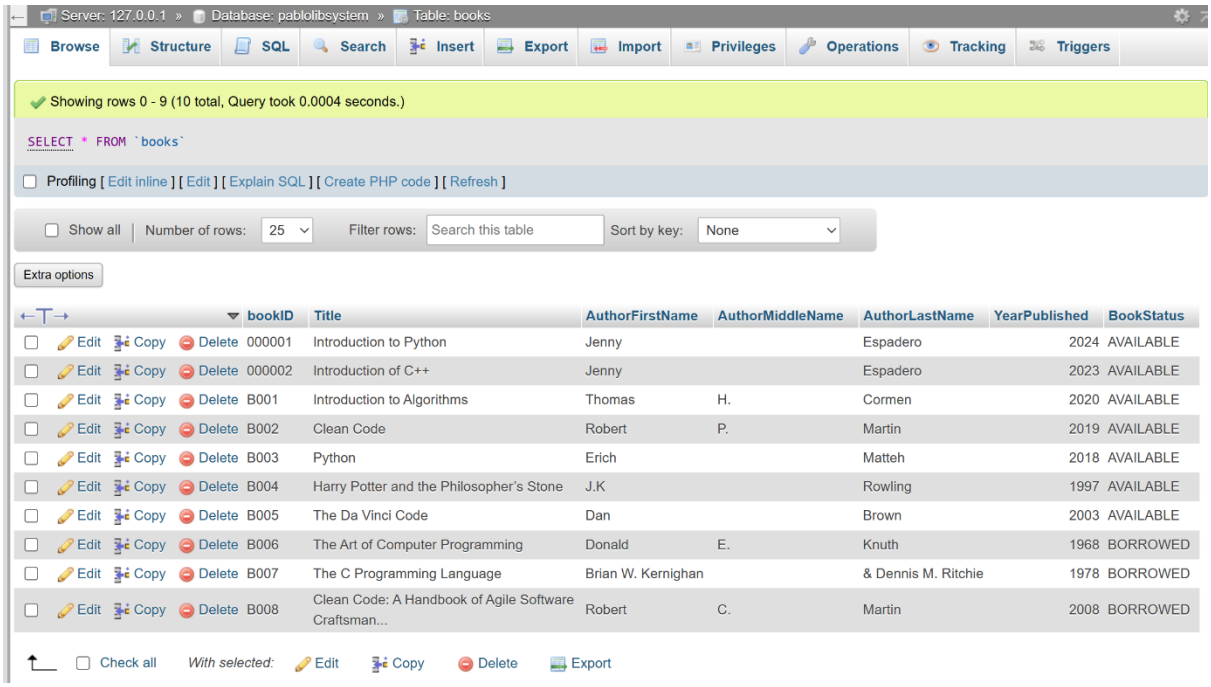



Figure 11: books Records

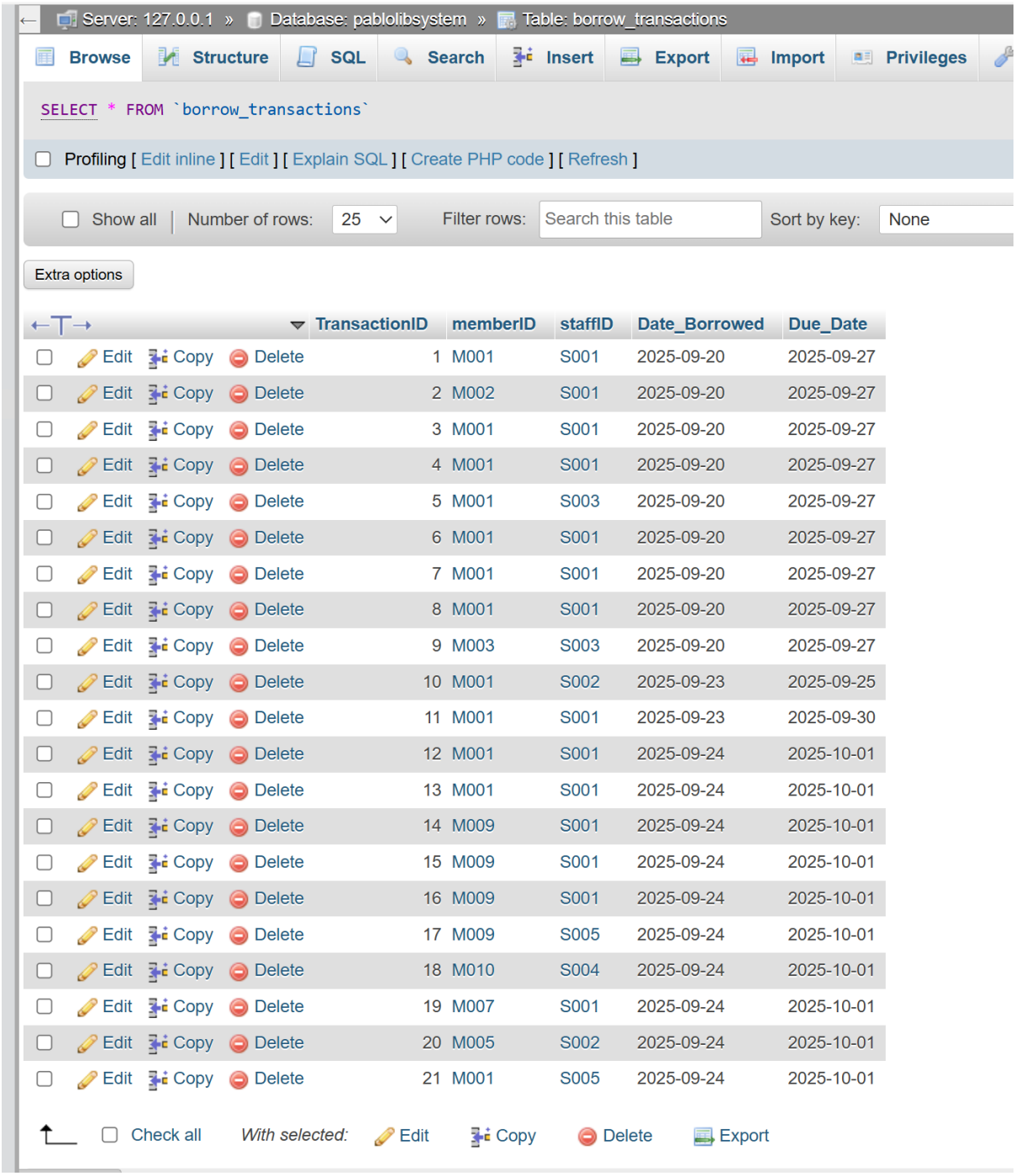



Figure 13: borrow_transactions Records

## Utilizing Functions

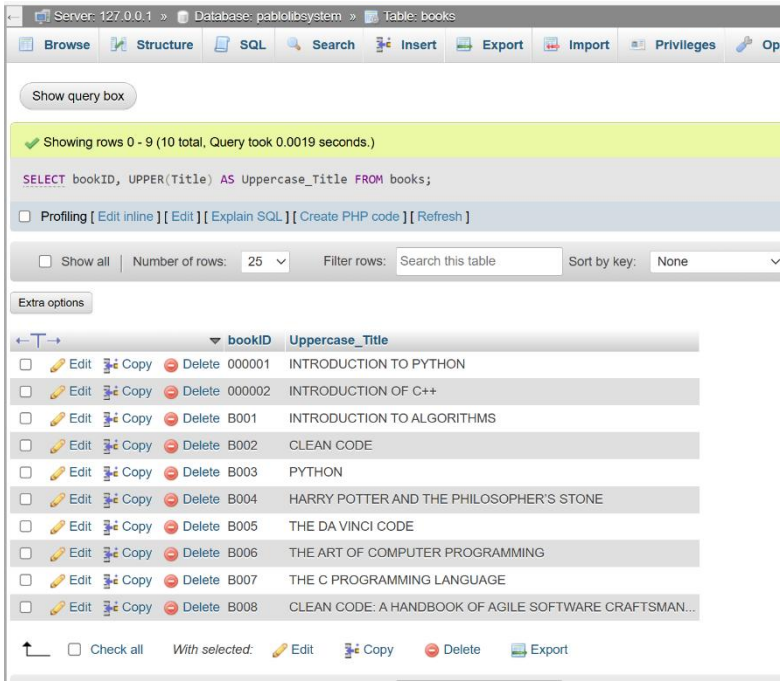



Figure 14: Function UPPER()

The UPPER() function converts text to uppercase. In the library system, it is used to standardize book titles for easier searching and reporting.

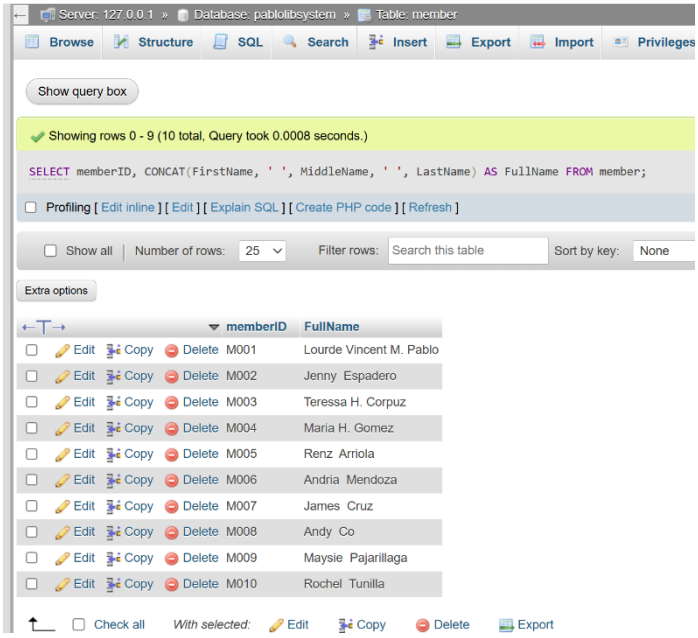



Figure 15: Function CONCAT()

The CONCAT() function combines multiple fields into one string. It is applied to display a member's full name by merging first, middle, and last names.

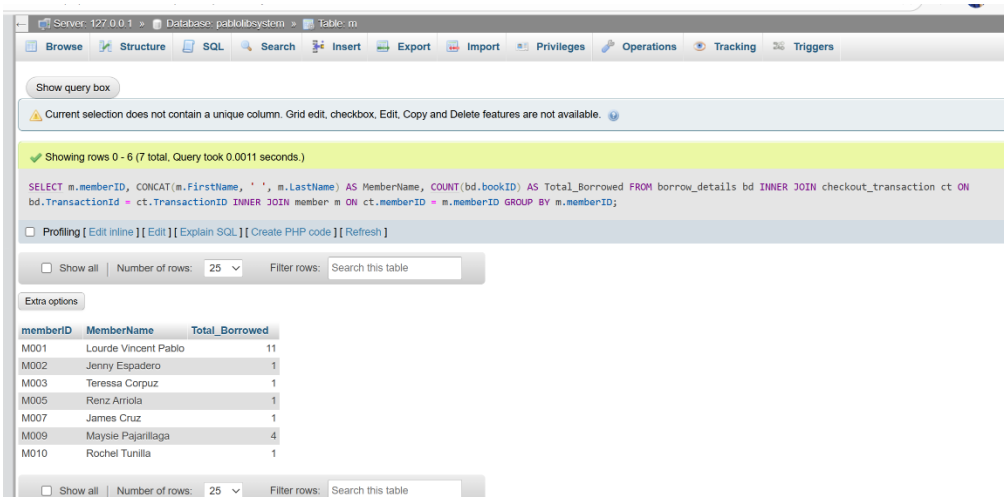



Figure 16: Function COUNT()

The COUNT() function returns the number of records that meet a condition. In the library system, it shows how many books each member has borrowed.

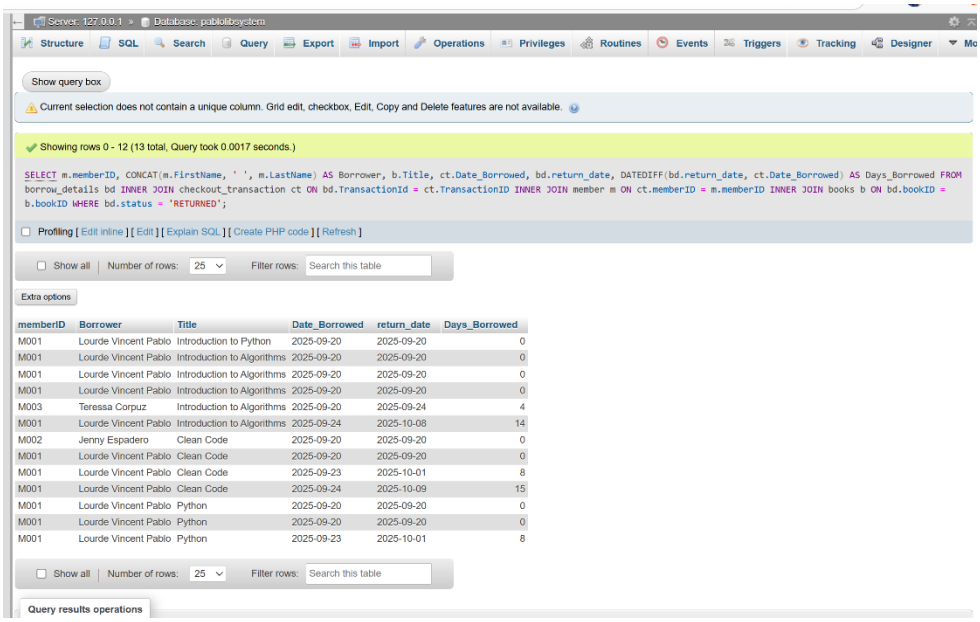



Figure 17: Function DATEDIFF()

The DATEDIFF() function calculates the number of days between two dates. It is useful for tracking borrowing duration and identifying overdue books.

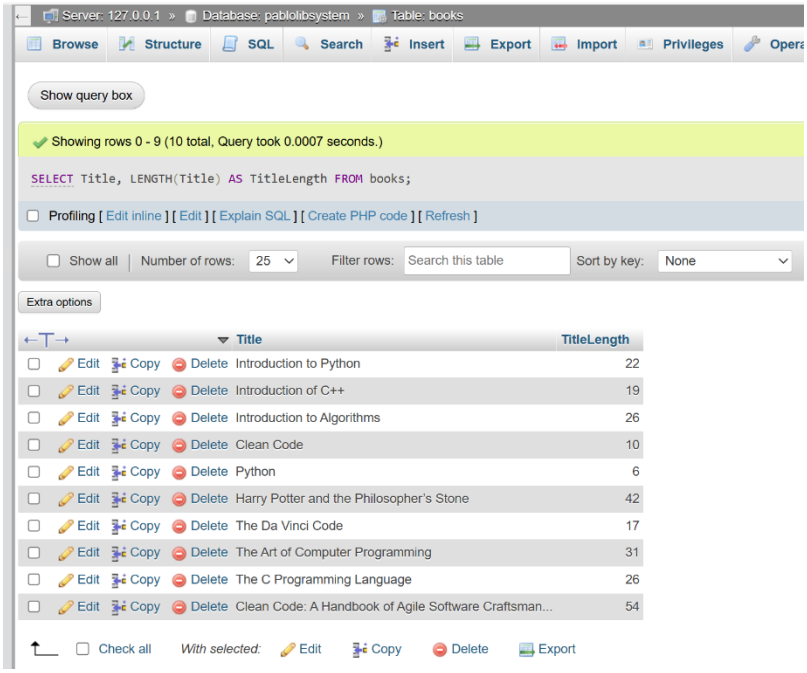



Figure 18: Function LENGTH()

The LENGTH() function returns the number of characters in a string. In the library system, it can be used to measure the length of book titles or member names, helping with data validation and formatting.

# Chapter 4

## Prototype

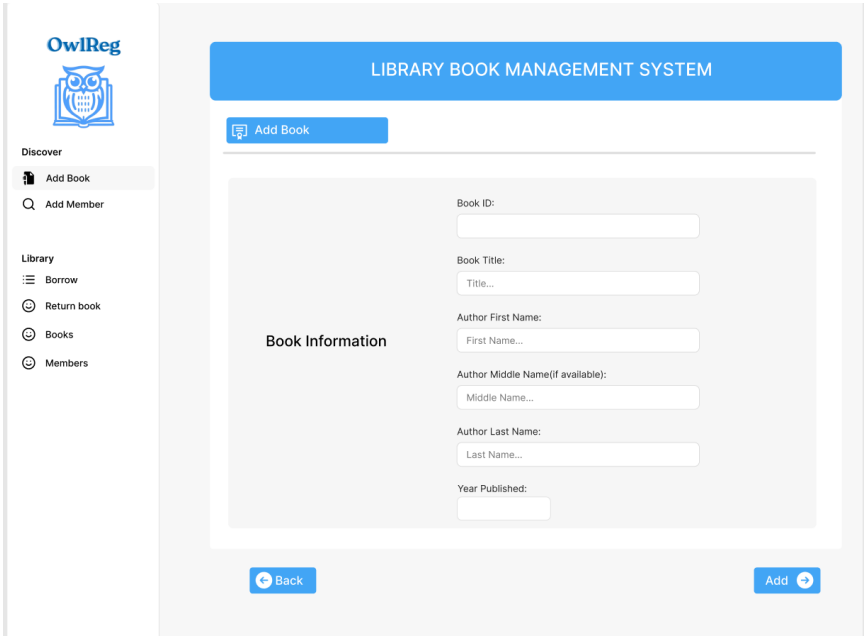



Figure 19: Add book navigation

```sql
INSERT INTO books (
    bookID, Title, AuthorFirstName, AuthorMiddleName,
    AuthorLastName, YearPublished, BookStatus
)
VALUES ('B001', 'Introduction to Python', 'Guido', '', 'Rossum', 1991, 'AVAILABLE');
```

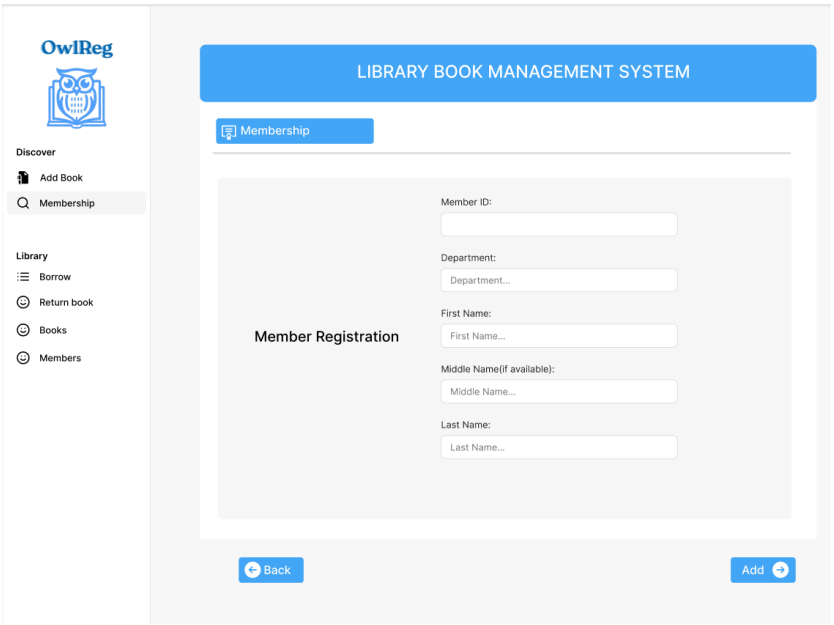



Figure 20: Add member navigation

```sql
INSERT INTO member (
    memberID, department, FirstName, MiddleName, LastName, Date_Joined
)
VALUES ('M001', 'STUDENT', 'Juan', 'Dela', 'Cruz', CURDATE());
```

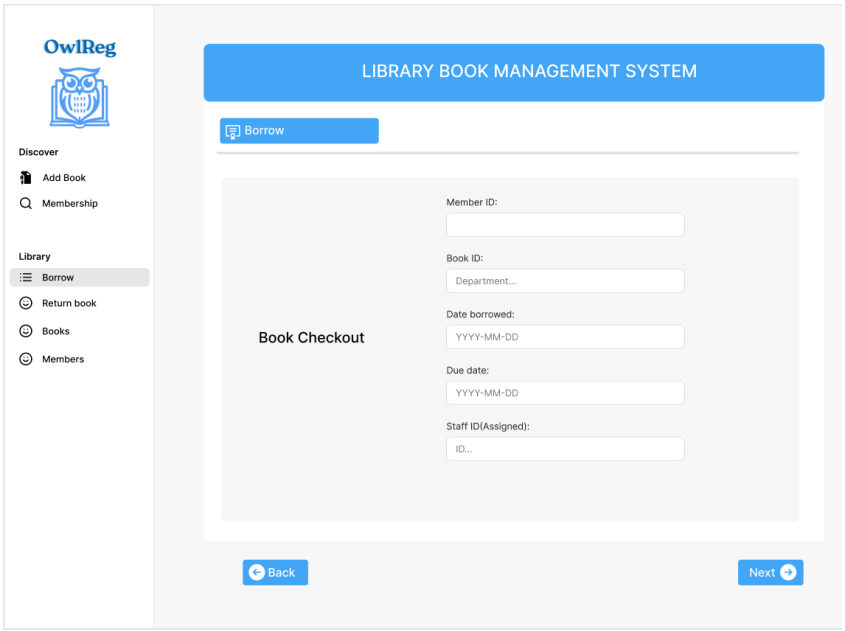



Figure 21: Borrow navigation

```sql
INSERT INTO borrow_transactions (memberID, staffID, Date_Borrowed, Due_Date)
VALUES ('M001', 'S001', '2025-09-30', '2025-10-07');

INSERT INTO borrow_details (TransactionId, bookID, status)
VALUES (LAST_INSERT_ID(), 'B001', 'BORROWED');

UPDATE books SET BookStatus='BORROWED' WHERE bookID='B001';
```

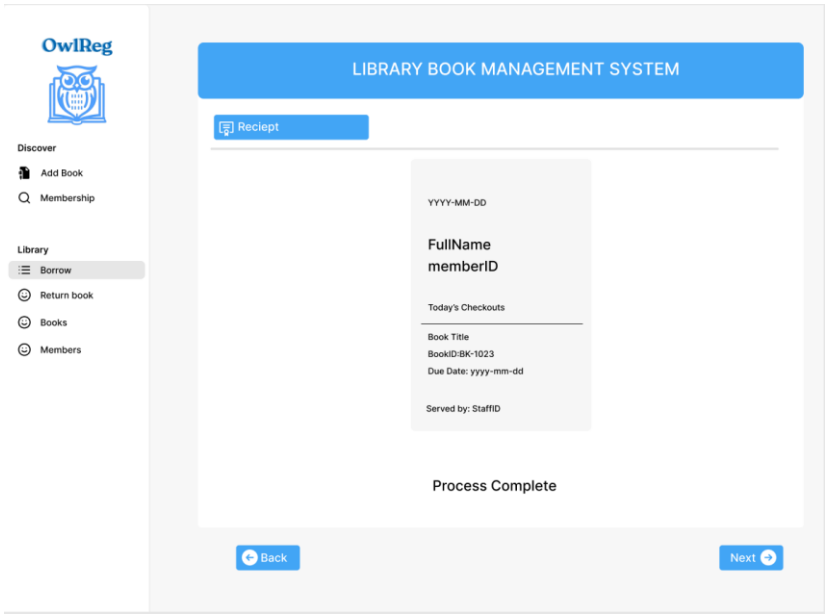



Figure 22: Borrow receipt

```sql
SELECT
    bt.TransactionID,
    m.FirstName || ' ' || m.LastName AS MemberName,
    s.FirstName || ' ' || s.LastName AS StaffName,
    b.Title,
    bt.DateBorrowed,
    bt.DueDate,
    bd.DateReturned
FROM borrowtransaction bt
JOIN member m ON bt.MemberID = m.MemberID
JOIN staff s ON bt.StaffID = s.StaffID
JOIN borrowdetails bd ON bt.TransactionID = bd.TransactionID
JOIN books b ON bd.BookID = b.BookID
WHERE bt.TransactionID = 1;
```

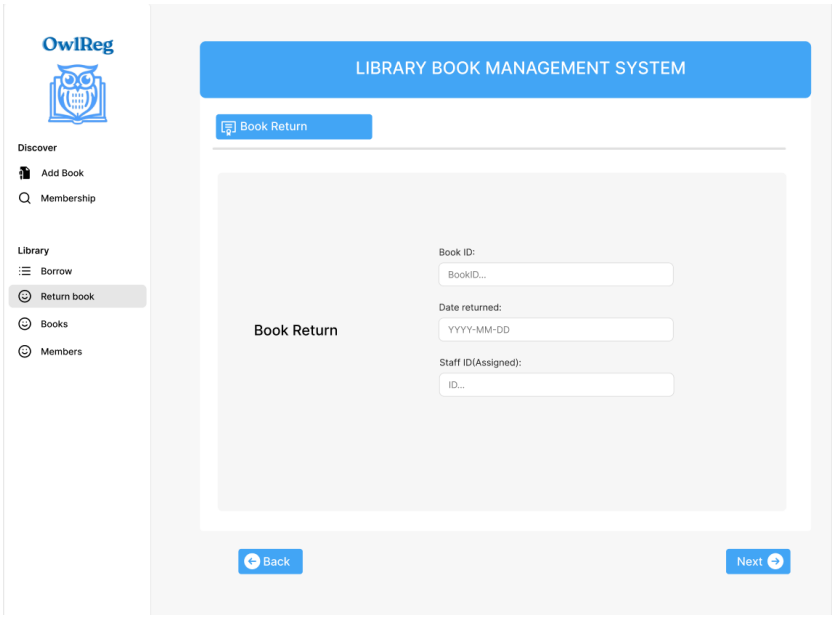



Figure 23: Book Return navigation

```sql
UPDATE borrow_details
SET return_date='2025-10-05', status='RETURNED'
WHERE detail_id=123;

UPDATE books SET BookStatus='AVAILABLE' WHERE bookID='B001';
```

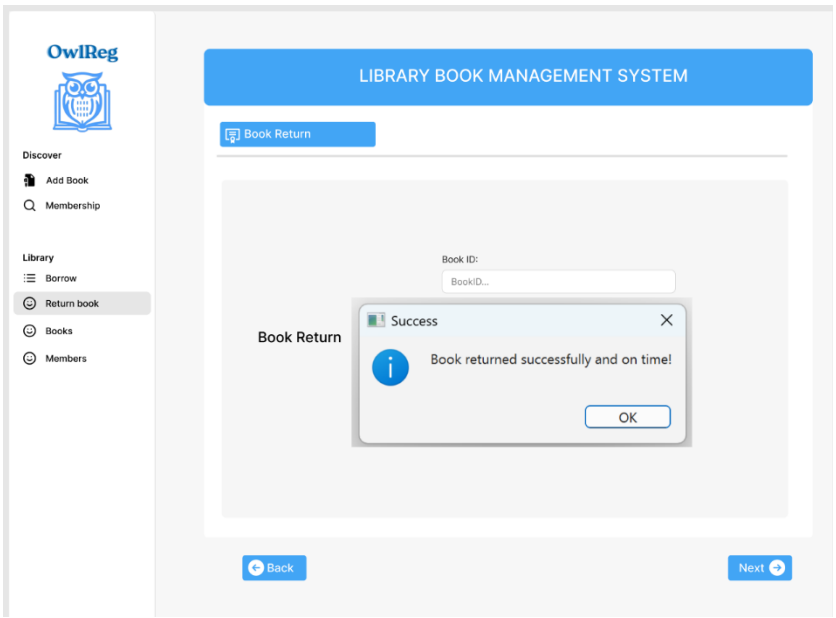



Figure 24: Book Return confirmation

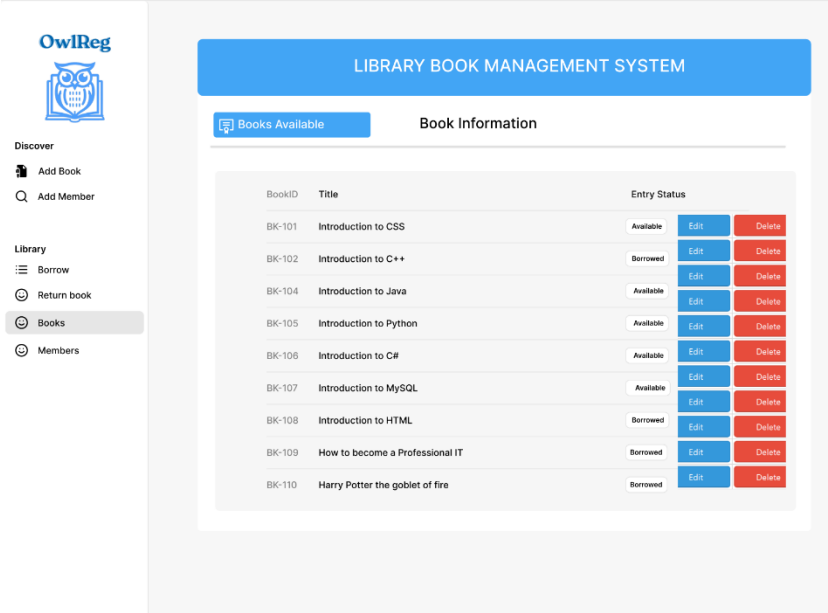



Figure 25: Books list table

```sql
SELECT
    bookID, Title, AuthorFirstName, AuthorMiddleName,
    AuthorLastName, YearPublished, BookStatus
FROM books
ORDER BY bookID;
```

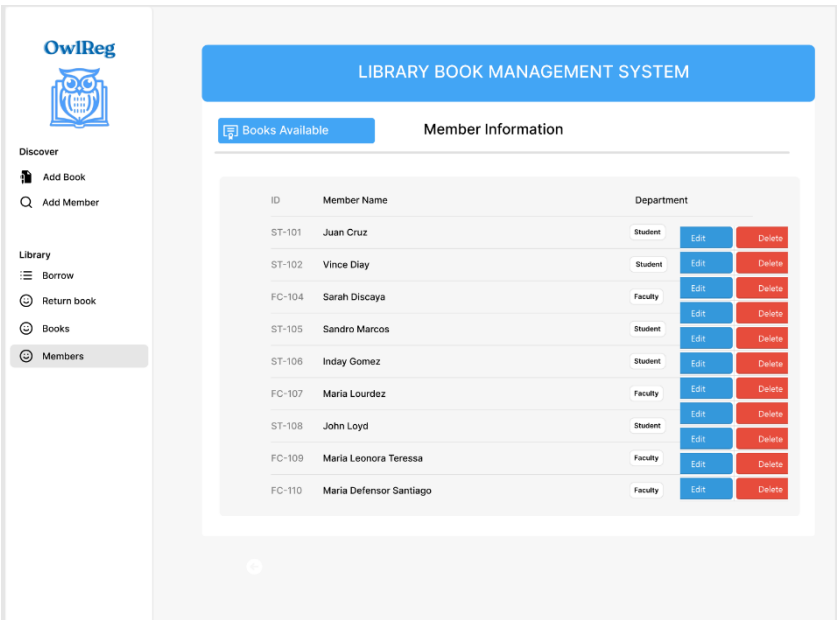



Figure 26: Members list table

```sql
SELECT
    memberID, department, FirstName, MiddleName, LastName
FROM member
ORDER BY memberID;
```

## Transaction/Processes Design

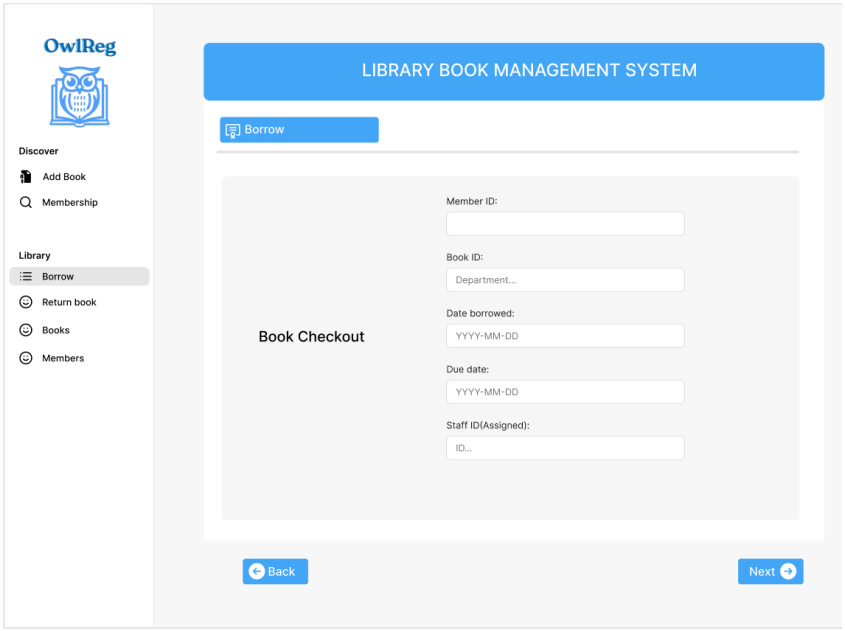



Figure 27: Borrow Transaction

```sql
-- Insert into borrow_transactions
INSERT INTO borrow_transactions (memberID, staffID, Date_Borrowed, Due_Date)
VALUES ('M001', 'S002', '2025-09-30', '2025-10-07');

-- Link the book into borrow_details
INSERT INTO borrow_details (TransactionId, bookID, status)
VALUES (LAST_INSERT_ID(), 'B011', 'BORROWED');

-- Update book status to BORROWED
UPDATE books
SET BookStatus = 'BORROWED'
WHERE bookID = 'B011';
```

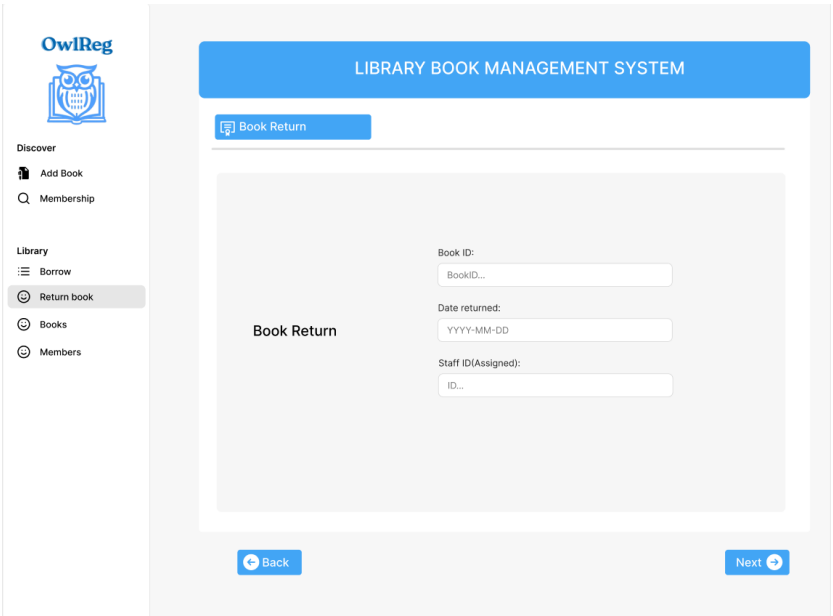



Figure 28: Return Transaction

```sql
-- Update borrow_details with return date
UPDATE borrow_details
SET return_date = '2025-10-05', status = 'RETURNED'
WHERE TransactionId = 21 AND bookID = 'B011';

-- Update book status back to AVAILABLE
UPDATE books
SET BookStatus = 'AVAILABLE'
WHERE bookID = 'B011';
```

## Reports

### List of All Borrowed Books

```sql
SELECT b.bookID, b.Title, m.FirstName, m.LastName, bt.Date_Borrowed, bt.Due_Date
FROM borrow_details bd
JOIN books b ON bd.bookID = b.bookID
JOIN borrow_transactions bt ON bd.TransactionId = bt.TransactionID
JOIN member m ON bt.memberID = m.memberID
WHERE bd.status = 'BORROWED';
```

### Borrowing History of a Member

```sql
SELECT m.memberID, m.FirstName, m.LastName, b.Title, bt.Date_Borrowed, bd.return_date, bd.status
FROM borrow_transactions bt
JOIN member m ON bt.memberID = m.memberID
JOIN borrow_details bd ON bt.TransactionID = bd.TransactionId
JOIN books b ON bd.bookID = b.bookID
WHERE m.memberID = 'M001';
```

### Most Borrowed Books

```sql
SELECT b.Title, COUNT(bd.bookID) AS TimesBorrowed
FROM borrow_details bd
JOIN books b ON bd.bookID = b.bookID
GROUP BY bd.bookID
ORDER BY TimesBorrowed DESC
LIMIT 5;
```

### Transactions Processed by Staff

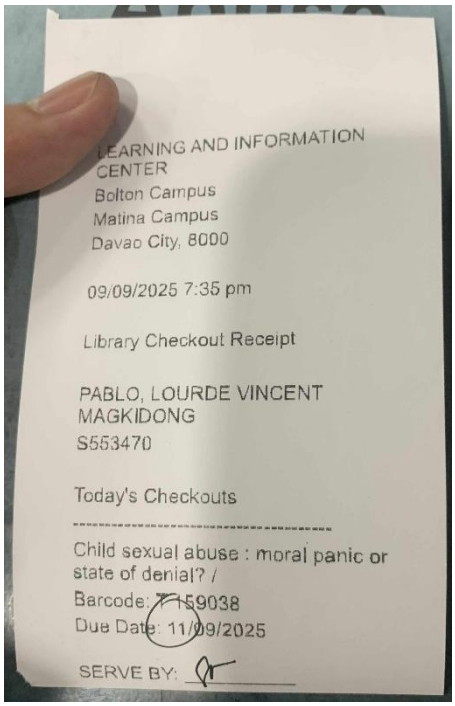
```sql
SELECT s.StaffFirstName, s.StaffLastName, COUNT(bt.TransactionID) AS TransactionsHandled
FROM borrow_transactions bt
JOIN staff s ON bt.staffID = s.staffID
GROUP BY bt.staffID
ORDER BY TransactionsHandled DESC;
```

### Daily Transaction summary

```sql
SELECT Date_Borrowed, COUNT(TransactionID) AS TotalTransactions
FROM borrow_transactions
GROUP BY Date_Borrowed
ORDER BY Date_Borrowed DESC;
```

# Appendix A

LIC Receipt



# References

Andri, A., Sopiah, N., & Oktaviani, N. (2025). Development of a book borrowing & returning system using the extreme programming method. *Journal of Information Systems and Application Innovation, 6(2)*. Retrieved from https://doi.org/10.26486/jisai.v6i2.239

Marlindawati ., Misinem ., & Adhiatma, P. (2024). Analyzing and Enhancing Data Management for the E-Library Transaction System. *Journal of Data Science, 2024*. Retrieved from https://iuojs.intimal.edu.my/index.php/jods/article/view/558

Mengping, C. (2023). Strategies and practices to enhance the efficiency of borrowing and returning books in college libraries. *Academic Journal of Humanities & Social Sciences*, 6(22), 53–58. . Retrieved from https://doi.org/10.25236/AJHSS.2023.062209

Song, W. (2018). Analysis of new library borrowing and returning mode based on book transfer system. *In Proceedings of the 2018 International Workshop on Advances in Social Sciences (IWASS 2018)*. Retrieved from https://www.webofproceedings.org/proceedings_series/ESSP/IWASS%202018/IWASS1231018.pdf