DATABASE FINAL PROJECT
EL-GEM: LIBRARY BORROWING TRACKER

FINAL REPORT

CSA Database // MII212501

28/12/2025

Raihan Yudhistira Hartawan | 24/532727/PA/22529
I Ketut Ezra Hesperos | 24/546552/PA/23205
Aditya Evan Rahmadani | 24/536755/PA/22771

I. Introduction

A. Problem & Background

Libraries that rely on manual record-keeping or loosely structured digital systems often face issues such as data inconsistency, difficulty tracking borrowed books, and inefficiencies in managing users and inventory. As the number of books and users grows, these problems become more pronounced, leading to delays, errors, and reduced service quality. A well-designed database-backed web application is therefore essential to ensure accurate data storage, efficient retrieval, and reliable transaction handling

This project addresses these issues by developing a web-based library management system supported by a relational database. The system centralizes data related to books, categories, users (students and administrators), and borrowing transactions, enabling structured management and reporting

B. Objectives

The objectives of this project are:

1. To design a normalized relational database for a library management system
2. To implement key database constraints to maintain data integrity
3. To develop a web-based application that interacts with the database for daily library operations
4. To demonstrate correct database usage through testing and sample queries

C. Users

The system is designed for the following users:

Administrators: Manage books, categories, and oversee borrowing transactions

Students: View available books and perform borrowing-related actions

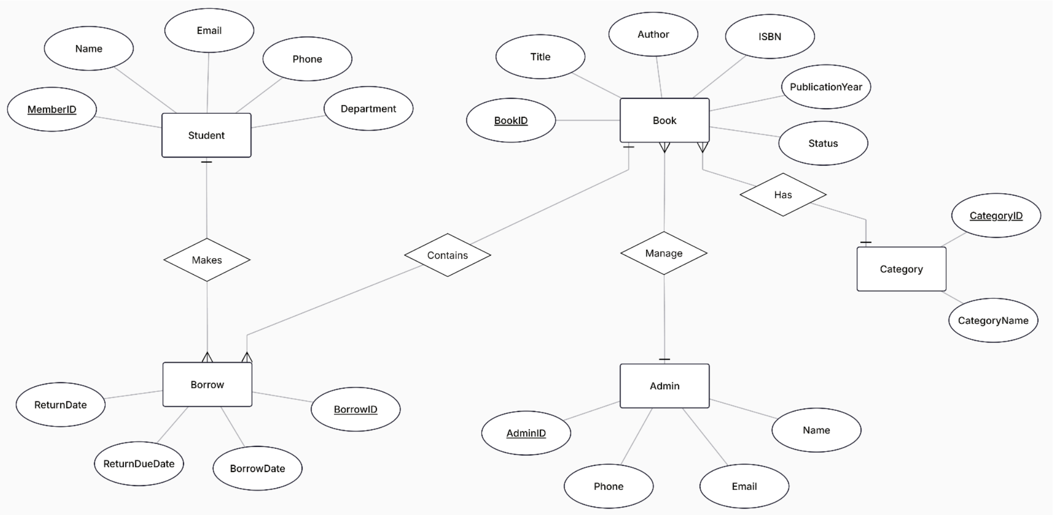
D. Use Cases

Key use cases supported by the system include:

1. Administrator login and management of library data
2. Book and category management
3. Student registration and identification
4. Borrowing and returning books

II. Database Design

A. Entity-Relation Diagram (ERD)



B. Entity Explanation

1. Student

This entity stores data for all registered library members.

- MemberID (Primary Key): A unique identifier for each student
- Name, Email, Phone: Attributes used for identification and communication regarding loan statuses
- Department: Categorizes students by their academic field

2. Book

The central repository for all items in the library collection

- BookID (Primary Key): A unique identifier for every digital copy
- Title, Author, ISBN: Metadata for bibliographic tracking
- PublicationYear: Records the release date of the title
- Status: A flag indicating availability (e.g., "Available", "Borrowed")

3. Borrow

An associative entity that records the transaction when a student borrows a book.

- BorrowID (Primary Key): A unique transaction serial number
- BorrowDate, ReturnDueDate: Tracks when the loan started and when it is expected back
- ReturnDate: Records the actual date of return to monitor late submissions

4. **Admin**

   Stores information for library staff who manage the system.

   - AdminID (Primary Key): Unique staff identifier
   - Name, Email, Phone: Professional contact details

5. **Category**

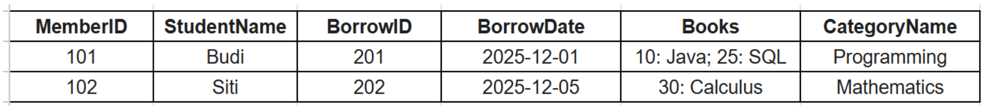
   Used to organize the library catalog into manageable sections

   - CategoryID (Primary Key): Unique code for the genre/subject
   - CategoryName: The descriptive title of the category (e.g., "History")

C. Normalization Steps

1. **Unnormalized Form (UNF)**

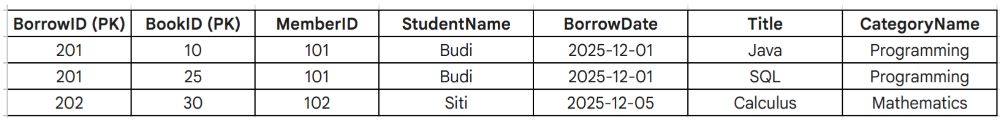
   In this stage, data is captured in a single flat file. It contains repeating groups and redundant information that would lead to data anomalies.



MemberID	StudentName	BorrowID	BorrowDate	Books	CategoryName
101	Budi	201	2025-12-01	10: Java; 25: SQL	Programming
102	Siti	202	2025-12-05	30: Calculus	Mathematics

2. **First Normal Form (1NF)**

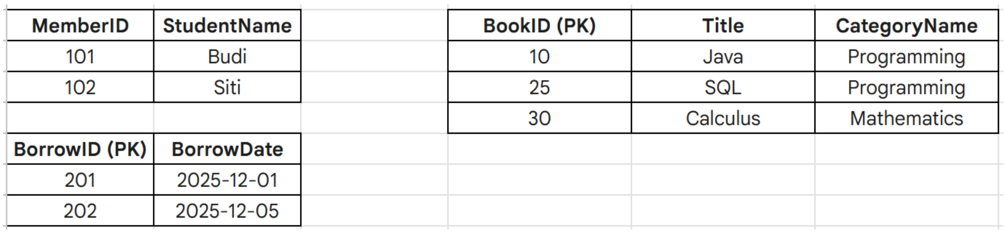
   Remove repeating groups and ensure atomicity. Every row now contains a single book record. We identify the composite Primary Key (BorrowID + BookID).



BorrowID (PK)	BookID (PK)	MemberID	StudentName	BorrowDate	Title	CategoryName
201	10	101	Budi	2025-12-01	Java	Programming
201	25	101	Budi	2025-12-01	SQL	Programming
202	30	102	Siti	2025-12-05	Calculus	Mathematics

3. **Second Normal Form (2NF)**

   Remove Partial Functional Dependencies. Non-key attributes must depend on the entire Primary Key. Student and Book details are moved to their own tables.



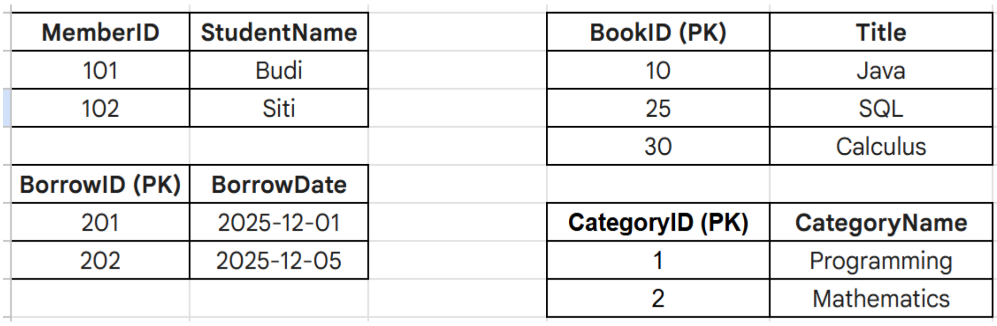
MemberID	StudentName
101	Budi
102	Siti

BorrowID (PK)	BorrowDate
201	2025-12-01
202	2025-12-05

BookID (PK)	Title	CategoryName
10	Java	Programming
25	SQL	Programming
30	Calculus	Mathematics

4. **Third Normal Form (3NF)**

   Remove Transitive Dependencies. CategoryName depends on CategoryID, which depends on BookID. We move category details to a dedicated table.



MemberID	StudentName
101	Budi
102	Siti

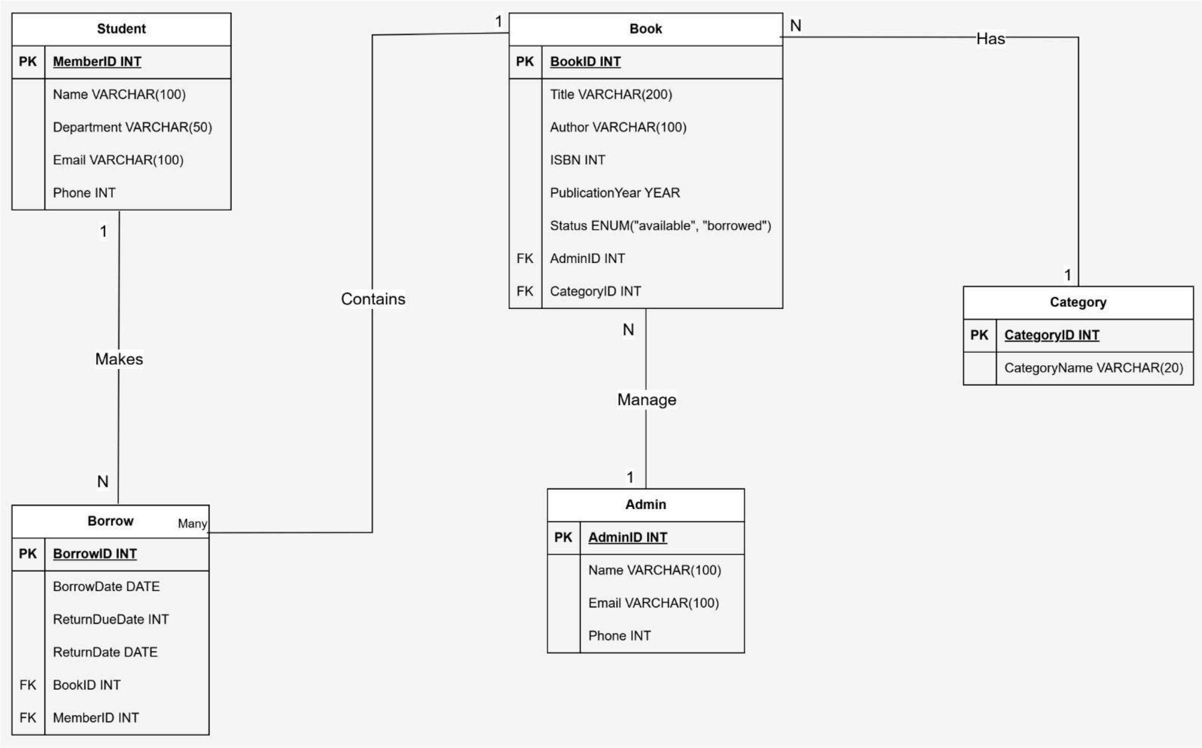
BorrowID (PK)	BorrowDate
201	2025-12-01
202	2025-12-05

BookID (PK)	Title
10	Java
25	SQL
30	Calculus

CategoryID (PK)	CategoryName
1	Programming
2	Mathematics

III. Database Implementation

A. Relational Schema



B. Key Constraints

PK = MemberID, BookID, BorrowID, AdminID, CategoryID

FK = - (BookID, MemberID), (AdminID, CategoryID)

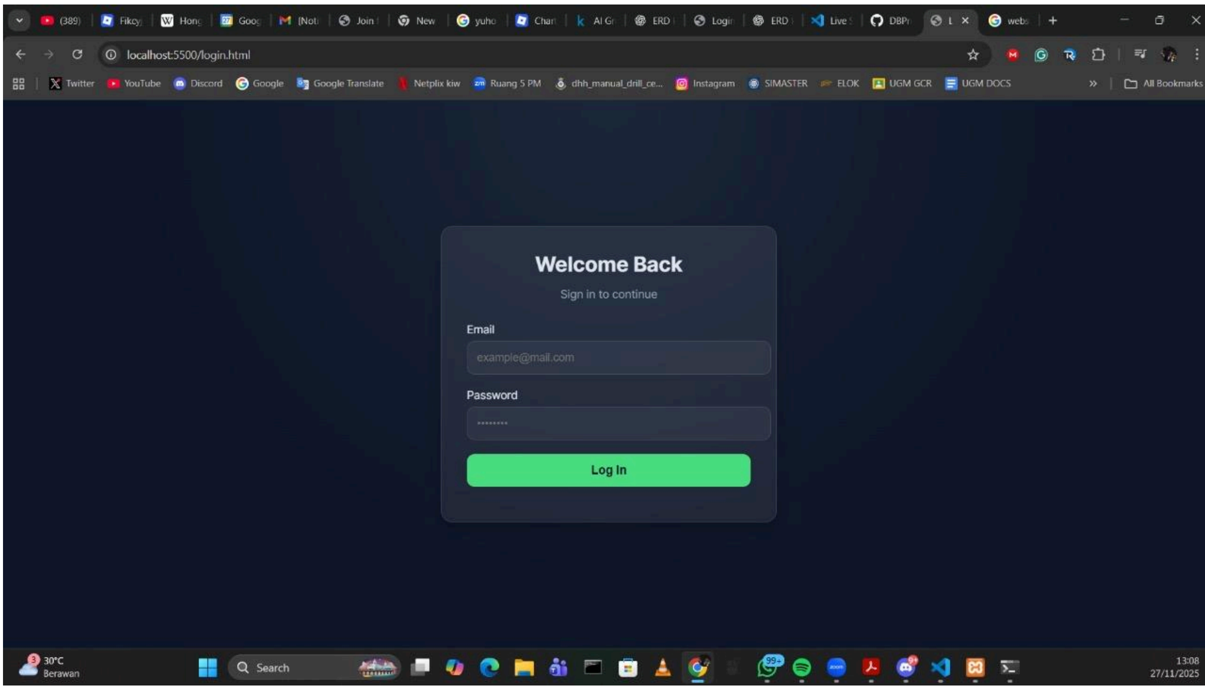
C. Example SQL Statements:

```
SELECT BookID, Title, Author, PublicationYear, Status
FROM Book
WHERE Status = 'available';
```

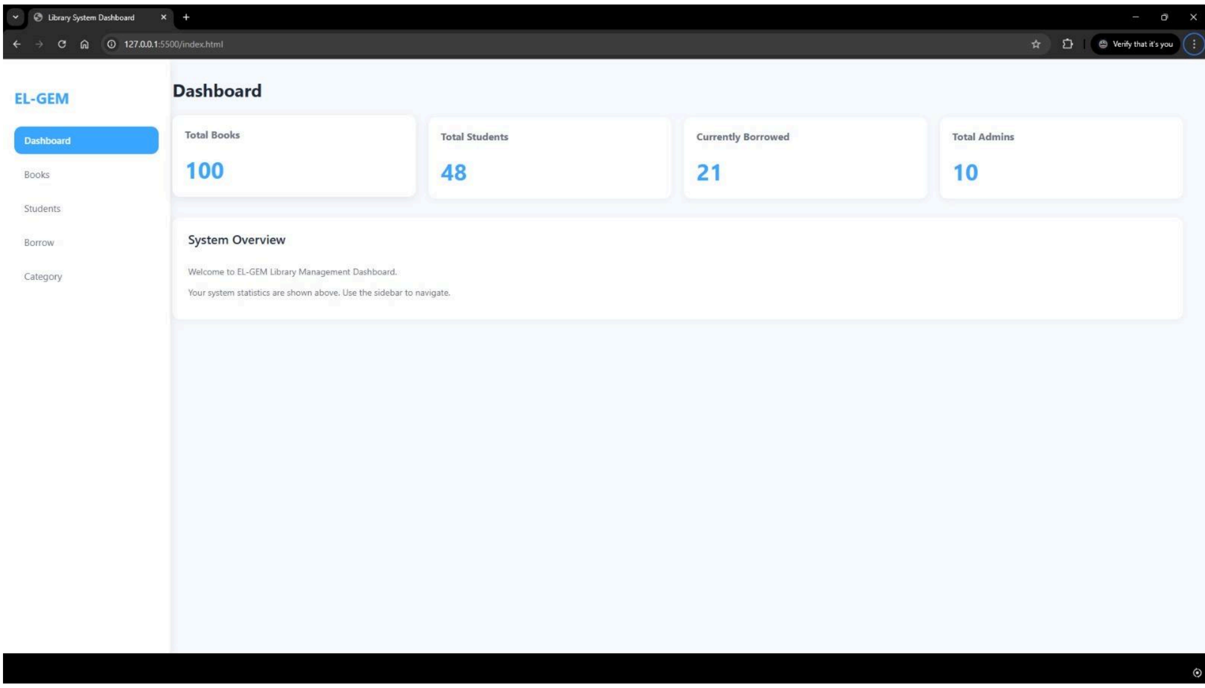
This query retrieves all books that are currently available for borrowing. It is used on the student and admin dashboard to show which books can still be borrowed

IV. Application Implementation

A. Screenshots :



Login page : everyone will go through this login page first and will enter their email and password (either they are admin or user) to access the correct page



Dashboard

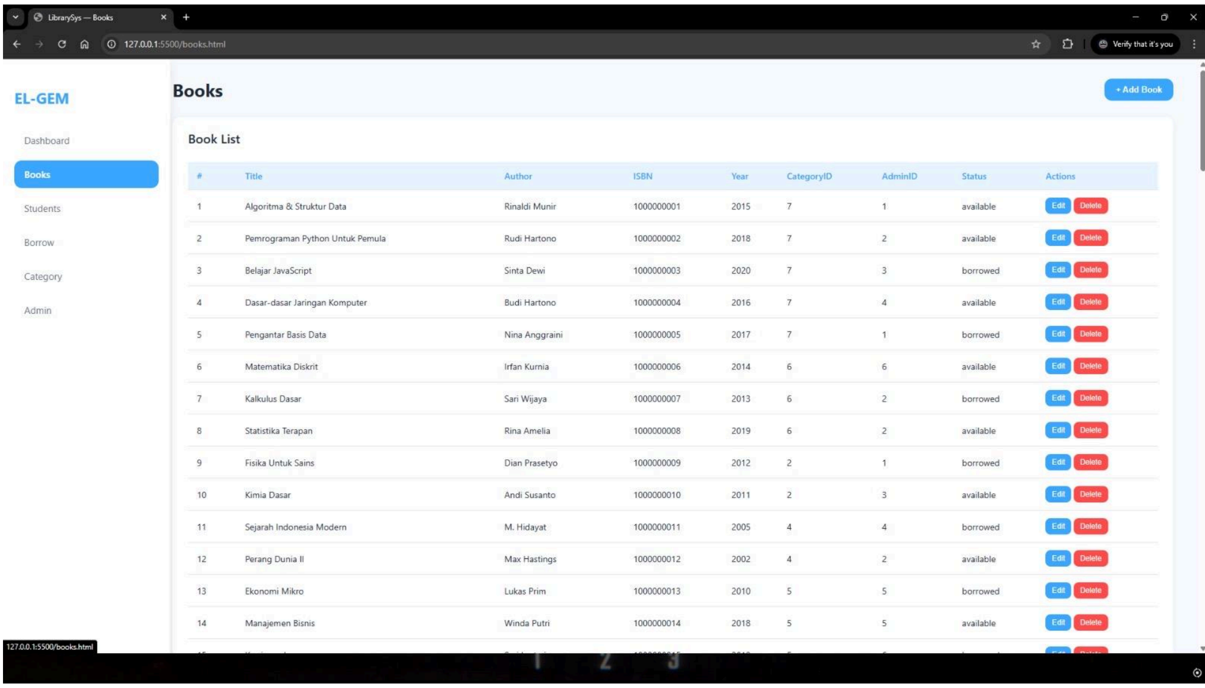
Total Books	Total Students	Currently Borrowed	Total Admins
100	48	21	10

System Overview

Welcome to EL-GEM Library Management Dashboard.

Your system statistics are shown above. Use the sidebar to navigate.

This is the Main Dashboard (admin), where total books, students, admin and currently borrowed are displayed in the first page. From the main dashboard we can access all the sections from Books, Students, Books, and Category



Books

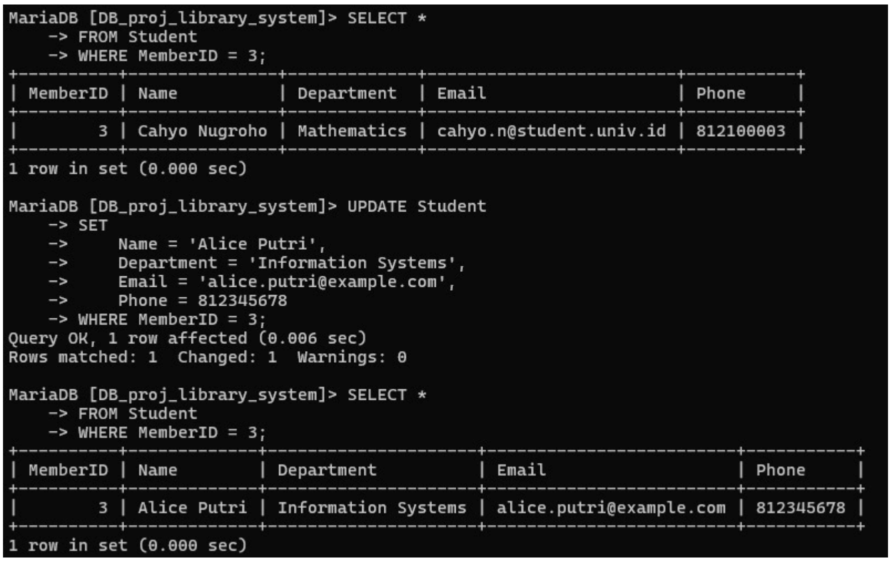
Book List

#	Title	Author	ISBN	Year	CategoryID	AdminID	Status	Actions
1	Algoritma & Struktur Data	Rinaldi Munir	1000000001	2015	7	1	available	Edit Delete
2	Pemrograman Python Untuk Pemula	Rudi Hartono	1000000002	2018	7	2	available	Edit Delete
3	Belajar JavaScript	Sinta Dewi	1000000003	2020	7	3	borrowed	Edit Delete
4	Dasar-dasar Jaringan Komputer	Budi Hartono	1000000004	2016	7	4	available	Edit Delete
5	Pengantar Basis Data	Nina Anggraini	1000000005	2017	7	1	borrowed	Edit Delete
6	Matematika Diskrit	Irfan Kurnia	1000000006	2014	6	6	available	Edit Delete
7	Kalkulus Dasar	Sari Wijaya	1000000007	2013	6	2	borrowed	Edit Delete
8	Statistika Terapan	Rina Amelia	1000000008	2019	6	2	available	Edit Delete
9	Fisika Untuk Sains	Dian Prasetyo	1000000009	2012	2	1	borrowed	Edit Delete
10	Kimia Dasar	Andi Susanto	1000000010	2011	2	3	available	Edit Delete
11	Sejarah Indonesia Modern	M. Hidayat	1000000011	2005	4	4	borrowed	Edit Delete
12	Perang Dunia II	Max Hastings	1000000012	2002	4	2	available	Edit Delete
13	Ekonomi Mikro	Lukas Prim	1000000013	2010	5	5	borrowed	Edit Delete
14	Manajemen Bisnis	Winda Putri	1000000014	2018	5	5	available	Edit Delete

Books Section (same as other sections), where we as admins can search, add, remove, or edit the data from every category to modify and handle things. And all the data is displayed in order

V. Testing & Results

A. Sample Queries



```
MariaDB [DB_proj_library_system]> SELECT *
    -> FROM Student
    -> WHERE MemberID = 3;
+----------+---------------+-------------+--------------------------+-----------+
| MemberID | Name          | Department  | Email                    | Phone     |
+----------+---------------+-------------+--------------------------+-----------+
|        3 | Cahyo Nugroho | Mathematics | cahyo.n@student.univ.id  | 812100003 |
+----------+---------------+-------------+--------------------------+-----------+
1 row in set (0.000 sec)

MariaDB [DB_proj_library_system]> UPDATE Student
    -> SET
    ->     Name = 'Alice Putri',
    ->     Department = 'Information Systems',
    ->     Email = 'alice.putri@example.com',
    ->     Phone = 812345678
    -> WHERE MemberID = 3;
Query OK, 1 row affected (0.006 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [DB_proj_library_system]> SELECT *
    -> FROM Student
    -> WHERE MemberID = 3;
+----------+-------------+---------------------+-------------------------+-----------+
| MemberID | Name        | Department          | Email                   | Phone     |
+----------+-------------+---------------------+-------------------------+-----------+
|        3 | Alice Putri | Information Systems | alice.putri@example.com | 812345678 |
+----------+-------------+---------------------+-------------------------+-----------+
1 row in set (0.000 sec)
```

B. Results

The test demonstrates correct record retrieval and update functionality within the Student table. An initial SELECT query successfully retrieved the student record with MemberID = 3, confirming that the primary key constraint enables precise tuple identification. The subsequent UPDATE statement modified multiple attributes (Name, Department, Email, and Phone) for the same MemberID, and the database reported that exactly one row was affected, indicating correct targeting and execution. A follow-up SELECT query verified that the updated values were persistently stored in the database. These results confirm that data manipulation operations (SELECT and UPDATE) function correctly and that the primary key constraint maintains data integrity during update operations

VI. Conclusion & Reflection

A. Conclusion

This project successfully demonstrates the design and implementation of a relational database-backed library management system. The system meets its objectives by providing structured data management, enforcing integrity constraints, and supporting essential library operations

B. Struggles

Difficulties on connecting login page to the main dashboard and separating admin and user login

C. Reflection

Through this project, we gained practical experience in database normalization, relational schema design, and integrating databases with web applications. Challenges included aligning database design with application logic and ensuring referential integrity. Future improvements may include role-based access control, advanced reporting features, and performance optimization

VII. Appendix

(https://github.com/ketutezraugm/DBProject_Group5_LibraryBorrowingTracker/tree/0271787ae29a6ff16c87b278ca763e9539627054/week2_schema_SQL)