Programming Assignment Unit 4

Department of Computer Science, UoPeople

CS 2203-01 - AY2025-T1

Instructor Ada Ajunwa

**2nd October 2023**

### Developing a Library Management Database System

#### Introduction

The aim of this assignment is to develop a simple database management system for a library using SQL. The database will consist of three main entities: **Books**, **Members**, and **Loans**. Each entity will be represented as a table with attributes that capture specific information needed to manage library operations efficiently. The system will allow for basic operations such as inserting new records, retrieving specific data, updating existing records, and deleting records. This document provides an overview of the schema, the SQL scripts, and an explanation of the queries used.

#### Database Schema

The database schema is designed to facilitate the management of library resources and member activities. Below are the descriptions of the tables and their attributes:

1. **Books Table**: This table stores information about the books available in the library.
   * **ISBN**: A unique identifier for each book (Primary Key).
   * **Title**: The title of the book.
   * **Author**: The name of the book’s author.
   * **Genre**: The genre or category of the book (e.g., Fiction, Science, History).
   * **Quantity**: The number of copies available in the library.
2. **Members Table**: This table keeps track of all registered library members.
   * **MemberID**: A unique identifier for each member (Primary Key).
   * **Name**: The full name of the member.
   * **Email**: The member's email address for communication.
   * **Phone**: The phone number of the member.
3. **Loans Table**: This table records the borrowing activities of members.
   * **LoanID**: A unique identifier for each loan record (Primary Key).
   * **MemberID**: The identifier of the member who borrowed a book (Foreign Key referencing Members table).
   * **ISBN**: The identifier of the borrowed book (Foreign Key referencing Books table).
   * **LoanDate**: The date the book was borrowed.
   * **ReturnDate**: The date the book was returned.

The relationships between the tables are defined using foreign keys to ensure referential integrity. The Loans table connects the Members and Books tables, representing the association between members and the books they borrow.

#### SQL Implementation

Below are the SQL scripts for creating the database schema, inserting records, and performing the required operations.

**1. Creating the Tables**

**-- Create Books table**

**CREATE TABLE Books (**

**ISBN VARCHAR(13) PRIMARY KEY,**

**Title VARCHAR(100),**

**Author VARCHAR(100),**

**Genre VARCHAR(50),**

**Quantity INT**

**);**

**-- Create Members table**

**CREATE TABLE Members (**

**MemberID INT PRIMARY KEY,**

**Name VARCHAR(100),**

**Email VARCHAR(100),**

**Phone VARCHAR(15)**

**);**

**-- Create Loans table**

**CREATE TABLE Loans (**

**LoanID INT PRIMARY KEY,**

**MemberID INT,**

**ISBN VARCHAR(13),**

**LoanDate DATE,**

**ReturnDate DATE,**

**FOREIGN KEY (MemberID) REFERENCES Members(MemberID),**

**FOREIGN KEY (ISBN) REFERENCES Books(ISBN)**

**);**

**These scripts create the three tables with their respective attributes and relationships.**

**2. Inserting Records**

**-- Insert records into Books table**

**INSERT INTO Books (ISBN, Title, Author, Genre, Quantity) VALUES**

**('978-3-16-148410-0', 'The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', 10),**

**('978-1-86197-876-9', 'Sapiens', 'Yuval Noah Harari', 'History', 5);**

**-- Insert records into Members table**

**INSERT INTO Members (MemberID, Name, Email, Phone) VALUES**

**(1, 'Alice Johnson', 'alice.johnson@example.com', '123-456-7890'),**

**(2, 'Bob Smith', 'bob.smith@example.com', '987-654-3210');**

**-- Insert records into Loans table**

**INSERT INTO Loans (LoanID, MemberID, ISBN, LoanDate, ReturnDate) VALUES**

**(1, 1, '978-3-16-148410-0', '2024-10-01', NULL),**

**(2, 2, '978-1-86197-876-9', '2024-10-01', '2024-10-15');**

**These statements add sample records to the** Books**,** Members**, and** Loans **tables.**

**3. Retrieve All Information About Books Borrowed by a Specific Member**

-- Retrieve all books borrowed by a specific member (e.g., MemberID = 1)

SELECT Books.ISBN, Books.Title, Books.Author, Books.Genre, Loans.LoanDate, Loans.ReturnDate

FROM Books

JOIN Loans ON Books.ISBN = Loans.ISBN

WHERE Loans.MemberID = 1;

**This query joins the** Books **and** Loans **tables to retrieve information about the books borrowed by a specific member identified by** MemberID**.**

**4. Update the Quantity of a Particular Book**

-- Update the quantity of a particular book (e.g., ISBN = '978-3-16-148410-0')

UPDATE Books

SET Quantity = Quantity - 1

WHERE ISBN = '978-3-16-148410-0';

This statement updates the quantity of a book after it has been borrowed.

**5. Delete a Member Record**

**-- Delete a member record from the Members table (e.g., MemberID = 2)**

**DELETE FROM Members**

**WHERE MemberID = 2;**

**This query removes a member's record from the** Members **table.**

#### Explanation of SQL Scripts

The provided SQL scripts include detailed comments for each statement, ensuring clarity regarding the purpose and function of the operations. The creation scripts define the structure of the database, including the primary and foreign keys that establish relationships between tables. The insertion scripts populate the tables with sample data for testing and demonstration purposes. The retrieval, update, and deletion scripts illustrate typical operations that may be performed on the database, showcasing the system's ability to handle real-world scenarios.

#### Conclusion

The library management database system developed in this assignment provides a foundational model for managing library resources and member activities. Through the use of SQL, we have created a schema that captures essential data, implemented operations for inserting, retrieving, updating, and deleting records, and ensured referential integrity using foreign keys. This project demonstrates the fundamental capabilities of a relational database system in supporting the administrative tasks of a library.