**Library Database Management System**

In Unit 4, we developed a simple database system to manage a library, which included the following tables:

1. **Books**: ISBN, Title, Author, Genre, Quantity
2. **Members**: MemberID, Name, Email, Phone
3. **Loans**: LoanID, MemberID, ISBN, LoanDate, ReturnDate

For this week's task, we will extend the database by adding a new table for authors, populate the tables with sample data, and perform SQL queries for specified operations.

### Step 1: Extend the Database Schema

To further organize the authors, we will create an additional **Authors** table. This will help manage author-related information more effectively. The new table will have the following structure:

* **Authors**: AuthorID, Name, Nationality, BirthYear

Here is the SQL code to create the new **Authors** table:

CREATE TABLE Authors (

AuthorID INT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

Nationality VARCHAR(50),

BirthYear INT

);

### Step 2: Populate Tables with Sample Data

We will now insert sample data into the **Books**, **Authors**, **Members**, and **Loans** tables. Below is the SQL code to populate these tables.

**Sample Data for Authors**:

INSERT INTO Authors (AuthorID, Name, Nationality, BirthYear)

VALUES

(1, 'George Orwell', 'British', 1903),

(2, 'Haruki Murakami', 'Japanese', 1949),

(3, 'J.K. Rowling', 'British', 1965),

(4, 'Isaac Asimov', 'Russian', 1920),

(5, 'Agatha Christie', 'British', 1890);

**Sample Data for Books**:

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

VALUES

('9780451524935', '1984', 'George Orwell', 'Dystopian', 4),

('9780553293357', 'Norwegian Wood', 'Haruki Murakami', 'Romance', 3),

('9780439139601', 'Harry Potter and the Prisoner of Azkaban', 'J.K. Rowling', 'Fantasy', 5),

('9780316098120', 'Foundation', 'Isaac Asimov', 'Science Fiction', 2),

('9780062073488', 'Murder on the Orient Express', 'Agatha Christie', 'Mystery', 6),

('9780316769488', 'The Catcher in the Rye', 'J.D. Salinger', 'Fiction', 4),

('9780140283297', 'The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', 5),

('9780061120084', 'To Kill a Mockingbird', 'Harper Lee', 'Fiction', 7),

('9780140177398', 'Of Mice and Men', 'John Steinbeck', 'Fiction', 3),

('9780141442333', 'Crime and Punishment', 'Fyodor Dostoevsky', 'Classic', 2);

**Sample Data for Members**:

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

VALUES

(1, 'John Doe', 'johndoe@example.com', '555-1234'),

(2, 'Jane Smith', 'janesmith@example.com', '555-5678'),

(3, 'Alice Johnson', 'alicej@example.com', '555-2468'),

(4, 'Bob Brown', 'bobb@example.com', '555-1357'),

(5, 'Charlie Green', 'charlieg@example.com', '555-7890'),

(6, 'Diane White', 'dianew@example.com', '555-1122'),

(7, 'Eve Black', 'eveb@example.com', '555-3344'),

(8, 'Frank Red', 'frankr@example.com', '555-5566'),

(9, 'Grace Blue', 'graceb@example.com', '555-7788'),

(10, 'Hank Yellow', 'hanky@example.com', '555-9900'),

(11, 'Ivy Brown', 'ivyb@example.com', '555-1212'),

(12, 'Jack Silver', 'jacks@example.com', '555-1313'),

(13, 'Kathy Gold', 'kathyg@example.com', '555-1414'),

(14, 'Leo Copper', 'leoc@example.com', '555-1515'),

(15, 'Mia Bronze', 'miab@example.com', '555-1616'),

(16, 'Nate Iron', 'natei@example.com', '555-1717'),

(17, 'Oscar Steel', 'oscarsteel@example.com', '555-1818'),

(18, 'Pat Green', 'patg@example.com', '555-1919'),

(19, 'Quinn Pink', 'quinnp@example.com', '555-2020'),

(20, 'Rita Violet', 'ritav@example.com', '555-2121');

**Sample Data for Loans**:

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

VALUES

(1, 1, '9780451524935', '2024-09-01', '2024-09-15'),

(2, 2, '9780553293357', '2024-09-05', '2024-09-20'),

(3, 3, '9780439139601', '2024-09-10', '2024-09-24'),

(4, 4, '9780316098120', '2024-09-12', '2024-09-26'),

(5, 5, '9780062073488', '2024-09-15', '2024-09-30');

### Step 3: Write SQL Queries for Operations

1. **Retrieve all books written by a specific author**:

This query fetches all the books written by a specific author (e.g., 'George Orwell'):

SELECT \* FROM Books

WHERE Author = 'George Orwell';

1. **Drop the newly created Authors table**:

To remove the **Authors** table from the database:

DROP TABLE Authors;

1. **Identify the names of all members who have borrowed a specific book**:

This query identifies members who have borrowed a specific book (e.g., ISBN '9780451524935'):

SELECT Members.Name

FROM Members

JOIN Loans ON Members.MemberID = Loans.MemberID

WHERE Loans.ISBN = '9780451524935';

1. **Add a new attribute 'MembershipType' to the Members table**:

This query alters the **Members** table by adding a new column for membership type:

ALTER TABLE Members

ADD MembershipType VARCHAR(50);

### References

Kroenke, D. M., & Auer, D. J. (2013). Database concepts (6th ed.). Pearson.