**Create authors table**

CREATE TABLE Authors (

AuthorID int PRIMARY KEY,

Name varchar(255),

Country varchar(100)

);

**Create books table**

CREATE TABLE Books

( BookID int PRIMARY KEY,

Title varchar(255) NOT NULL,

AuthorID int,

Genre varchar(100),

Price decimal(10,2),

PublicationDate date,

FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID) );

**Create members table**

CREATE TABLE Members (

MemberID int PRIMARY KEY,

FirstName varchar(100) NOT NULL,

LastName varchar(100) NOT NULL,

Email varchar(255),

JoinDate date

);

**Create borrowedBooks table**

CREATE TABLE BorrowedBooks (

BorrowID int PRIMARY KEY,

BookID int,

MemberID int,

BorrowDate date,

ReturnDate date,

FOREIGN KEY (BookID) REFERENCES Books(BookID),

FOREIGN KEY (MemberID) REFERENCES Members(MemberID)

);

**Insert authors data**

INSERT INTO Authors (AuthorID, Name, Country)

VALUES

(1, 'Khaled Hosseini', 'Afghanistan'),

(2, 'J.K. Rowling', 'United Kingdom'),

(3, 'Agatha Christie', 'United Kingdom'),

(4, 'Haruki Murakami', 'Japan'),

(5, 'Jane Austen', 'United Kingdom'),

(6, 'George Orwell', 'United Kingdom'),

(7, 'Gabriel Garcia Marquez', 'Colombia'),

(8, 'Stephen King', 'United States'),

(9, 'Paulo Coelho', 'Brazil'),

(10, 'Leo Tolstoy', 'Russia');

**Insert books data**

INSERT INTO Books (BookID, Title, AuthorID, Genre, Price, PublicationDate)

VALUES

(1, 'The Kite Runner', 1, 'Fiction', 12.99, '2003-05-29'),

(2, 'Harry Potter and the Philosopher''s Stone', 2, 'Fantasy', 14.99, '1997-06-26'),

(3, 'Murder on the Orient Express', 3, 'Mystery', 10.99, '1934-01-01'),

(4, 'Norwegian Wood', 4, 'Fiction', 11.50, '1987-09-01'),

(5, 'Pride and Prejudice', 5, 'Romance', 9.99, '1813-01-28'),

(6, '1984', 6, 'Dystopian', 13.50, '1949-06-08'),

(7, 'One Hundred Years of Solitude', 7, 'Magical Realism', 12.75, '1967-05-30'),

(8, 'The Shining', 8, 'Horror', 11.25, '1977-01-28'),

(9, 'The Alchemist', 9, 'Philosophical Fiction', 10.00, '1988-01-01'),

(10, 'War and Peace', 10, 'Historical Fiction', 15.99, '1869-01-01');

**Insert borrowedBooks data**

INSERT INTO BorrowedBooks (BorrowID, BookID, MemberID, BorrowDate, ReturnDate)

VALUES

(1, 1, 1, '2023-01-10', '2023-01-20'),

(2, 2, 2, '2023-01-15', '2023-01-25'),

(3, 3, 3, '2023-01-12', '2023-01-22'),

(4, 4, 4, '2023-02-05', '2023-02-15'),

(5, 5, 5, '2023-02-10', '2023-02-20'),

(6, 6, 6, '2023-02-15', '2023-02-25'),

(7, 7, 7, '2023-03-01', '2023-03-11'),

(8, 8, 8, '2023-03-05', '2023-03-15'),

(9, 9, 9, '2023-03-10', '2023-03-20'),

(10, 10, 10, '2023-03-15', '2023-03-25');

**Insert members data**

-- Insert sample data into Members table

INSERT INTO Members (MemberID, FirstName, LastName, Email, JoinDate)

VALUES

(1, 'Ahmad', 'Khalil', 'ahmad.khalil@example.com', '2022-03-15'),

(2, 'Layla', 'Nasser', 'layla.nasser@example.com', '2022-04-20'),

(3, 'Omar', 'Mahmoud', 'omar.mahmoud@example.com', '2022-05-10'),

(4, 'Yara', 'Hassan', 'yara.hassan@example.com', '2022-06-05'),

(5, 'Sara', 'Amir', 'sara.amir@example.com', '2022-07-01'),

(6, 'Nour', 'Rami', 'nour.rami@example.com', '2022-08-15'),

(7, 'Ali', 'Salem', 'ali.salem@example.com', '2022-09-10'),

(8, 'Rana', 'Jawad', 'rana.jawad@example.com', '2022-10-05'),

(9, 'Mohammed', 'Farhan', 'mohammed.farhan@example.com', '2022-11-20'),

(10, 'Fatima', 'Abdullah', 'fatima.abdullah@example.com', '2022-12-15');

**Please give me the output for the following SQL statements: Note: Fore some tasks fill the data to ensure that the tasks run as SQL query / phpMyAdmin**

1. **SQL INSERT INTO Statement - Task: Add a new book to the `Books` table, With values ('The Great Gatsby', 1, 'Fiction', 15.99, '1925-04-10')**

INSERT INTO Books (Title, AuthorID, Genre, Price, PublicationDate)

VALUES ('The Great Gatsby', 1, 'Fiction', 15.99, '1925-04-10');

1. **SQL SELECT Statement - Task: Retrieve all the books written by the author with AuthorID = 1.**

SELECT \*

FROM Books

WHERE AuthorID = 1;

1. **SQL UPDATE Statement - Task: Update the price of the book with BookID = 2 to 20.99.**

UPDATE Books

SET Price = 20.99

WHERE BookID = 2;

1. **SQL DELETE Statement - Task: Remove the book with BookID = 3 from the database.**
2. **SQL WHERE Clause - Task: Find all books in the 'Science Fiction' genre.**

SELECT \*

FROM Books

WHERE Genre = ' Mystery';

1. **SQL AND, OR and NOT Operators - Task: Retrieve books that are either in the 'Fiction' genre and priced below 20, or not written by the author with AuthorID = 2.**

SELECT \*

FROM Books

WHERE (Genre = 'Mystery' AND Price < 13)

OR (AuthorID <> 2);

1. **SQL ORDER BY Keyword - Task: Get all books ordered by their publication date in descending order.**

SELECT \*

FROM Books

ORDER BY PublicationDate DESC;

1. **SQL MIN() and MAX() Functions - Task: Find the minimum and maximum price of books in the library. 9. SQL COUNT(), AVG() and SUM() Functions - Task: Get the total number of books, the average price, and the total price of all books in the library.**

SELECT MIN(Price) AS MinPrice, MAX(Price) AS MaxPrice

FROM Books;

SELECT

COUNT(\*) AS TotalBooks,

AVG(Price) AS AvgPrice,

SUM(Price) AS TotalPrice

FROM Books;

1. **SQL LIKE Operator - Task: Find all books with a title that starts with 'The'.**

SELECT \*

FROM Books

WHERE Title LIKE 'The%';

**11. SQL GROUP BY Statement - Task: Get the count of books for each genre.**

SELECT Genre, COUNT(\*) AS GenreCount

FROM Books

GROUP BY Genre;

**12. SQL INNER JOIN Keyword -**

**Task: Retrieve a list of books along with the names of their authors.**

SELECT books.Title, authers.Name AS AuthorName

FROM Books b

JOIN Authors authers ON books.AuthorID = authers.AuthorID;