Library Management System
1. Members (MemberID, Name, Email, Phone, MembershipDate)
2. Books (BookID, Title, Author, Genre, CopiesAvailable)
3. BorrowedBooks (BorrowID, MemberID, BookID, BorrowDate, ReturnDate)
Write queries for the following questions:
1. CREATE TABLE – Create a Fines table with FineID, MemberID (FK), Amount, Status, and
FineDate.
2. UPDATE – Update CopiesAvailable when a book is borrowed or returned.
3. SELECT with JOIN & Operators – Retrieve Member Name, Book Title, and Borrow Date for
books borrowed in the last month.
4. GROUP BY & Aggregate Function – Find the number of books borrowed per genre.
5. Joins & Aggregate Functions – Find the top 5 members who borrowed the most books.

```
CREATE TABLE Members (
    MemberID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    MembershipDate DATE
);

CREATE TABLE Books (
    BookID INT PRIMARY KEY AUTO_INCREMENT,
    Title VARCHAR(150),
    Author VARCHAR(100),
    Genre VARCHAR(50),
    CopiesAvailable INT
);

CREATE TABLE BorrowedBooks (
    BorrowID INT PRIMARY KEY AUTO_INCREMENT,
    MemberID INT,
    BookID INT,
    BorrowDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
);

-- Members
INSERT INTO Members (Name, Email, Phone, MembershipDate) VALUES
('Aarav Mehta', 'aarav@example.com', '9876543210', '2023-01-10'),
('Diya Sharma', 'diya@example.com', '9876543211', '2023-02-15'),
('Raj Patel', 'raj@example.com', '9876543212', '2023-03-20'),
```

```
('Sneha Reddy', 'sneha@example.com', '9876543213', '2023-04-25');

-- Books
INSERT INTO Books (Title, Author, Genre, CopiesAvailable) VALUES
('Wings of Fire', 'A.P.J. Abdul Kalam', 'Biography', 5),
('The Alchemist', 'Paulo Coelho', 'Fiction', 3),
('Python Programming', 'John Zelle', 'Education', 4),
('Ikigai', 'Francesc Miralles', 'Self-help', 2),
('Atomic Habits', 'James Clear', 'Self-help', 1);

-- BorrowedBooks
INSERT INTO BorrowedBooks (MemberID, BookID, BorrowDate, ReturnDate) VALUES
(1, 1, '2024-03-10', '2024-03-20'),
(2, 2, '2024-03-15', '2024-03-25'),
(3, 3, '2024-04-01', NULL),
(1, 4, '2024-04-05', NULL),
(4, 5, '2024-04-10', NULL);

-- 1. CREATE TABLE – Create a Fines table with FineID, MemberID (FK), Amount,
Status, and FineDate.
CREATE TABLE Fines (
    FineID INT PRIMARY KEY AUTO_INCREMENT,
    MemberID INT,
    Amount DECIMAL(10,2),
    Status VARCHAR(20),
    FineDate DATE,
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);

-- Fines
INSERT INTO Fines (MemberID, Amount, Status, FineDate) VALUES
(1, 50.00, 'Unpaid', '2024-03-22'),
(2, 30.00, 'Paid', '2024-03-27');

-- 2. Update CopiesAvailable When a Book is Borrowed or Returned
-- Book borrowed (e.g., BookID = 3)
UPDATE Books
SET CopiesAvailable = CopiesAvailable - 1
WHERE BookID = 3;

-- Book returned (e.g., BookID = 3)
UPDATE Books
SET CopiesAvailable = CopiesAvailable + 1
WHERE BookID = 3;

-- 3. Retrieve Member Name, Book Title, and Borrow Date for books borrowed in the
last month
SELECT m.Name, b.Title, bb.BorrowDate
FROM BorrowedBooks bb
JOIN Members m ON bb.MemberID = m.MemberID
```

```
JOIN Books b ON bb.BookID = b.BookID
WHERE bb.BorrowDate >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH);

-- 4. Find the Number of Books Borrowed Per Genre
SELECT bk.Genre, COUNT(bb.BorrowID) AS BooksBorrowed
FROM BorrowedBooks bb
JOIN Books bk ON bb.BookID = bk.BookID
GROUP BY bk.Genre;

-- 5. Find the Top 5 Members Who Borrowed the Most Books
SELECT m.Name, COUNT(bb.BorrowID) AS BooksBorrowed
FROM BorrowedBooks bb
JOIN Members m ON bb.MemberID = m.MemberID
GROUP BY m.MemberID
ORDER BY BooksBorrowed DESC
LIMIT 5;
```