Project 1

Title: Library Management System (using SQL)

Task 1 Database Creation:

a) Create Books table with columns BOOK_ID, TITLE, AUTHOR, GENRE, YEAR_PUBLISHED, AVAILABLE_COPIES.

```
CREATE TABLE Books (

BOOK_ID INT PRIMARY KEY AUTO_INCREMENT,

TITLE VARCHAR(150) NOT NULL,

AUTHOR VARCHAR(150) NOT NULL,

GENRE VARCHAR(50),

YEAR_PUBLISHED INT,

AVAILABLE_COPIES INT DEFAULT 0

);
```

b) Create Members table with columns MEMBER_ID, NAME, EMAIL, PHONE_NO, ADDRESS, MEMBERSHIP_DATE.

```
CREATE TABLE Members (
MEMBER_ID INT AUTO_INCREMENT PRIMARY KEY,
NAME VARCHAR(255) NOT NULL,
EMAIL VARCHAR(255) UNIQUE NOT NULL,
PHONE_NO INT(15),
ADDRESS TEXT,
MEMBERSHIP_DATE DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

c) Create BorrowingRecords table with columns BORROW_ID, MEMBER_ID, BOOK_ID, BORROW_DATE, RETURN_DATE. Set foreign key constraints linking MEMBER_ID to Members and BOOK_ID to Books.

```
CREATE TABLE BorrowingRecords (
BORROW_ID INT PRIMARY KEY AUTO_INCREMENT,
MEMBER_ID INT NOT NULL,
BOOK_ID INT NOT NULL,
BORROW_DATE DATETIME DEFAULT CURRENT_TIMESTAMP,
RETURN_DATE DATETIME DEFAULT NULL,

FOREIGN KEY (MEMBER_ID) REFERENCES Members(MEMBER_ID),
FOREIGN KEY (BOOK_ID) REFERENCES Books(BOOK_ID)
);
```

Task 2 Data Creation:

Question:

Insert sample into Books table

Query:

```
INSERT INTO Books (TITLE, AUTHOR, GENRE, YEAR_PUBLISHED, AVAILABLE_COPIES)
VALUES
('400 DAYS', 'Chethan Bhagat', 'Mystery', 2021, 4),
('Lazarus', 'Lars Kepler', 'Thriller', 2018, 5),
('Part of the Family', 'Charlotte Philby', 'Thriller', 2020, 3),
('The Night Circus', 'Erin Morgenstern', 'Fantasy', 2011, 2),
('The Road', 'Cormac McCarthy', 'Fiction', 2006, 6);
```

Question:

Insert sample into Members table

Query:

```
INSERT INTO Members (NAME, EMAIL, PHONE_NO, ADDRESS, MEMBERSHIP_DATE)
VALUES
('Allan', 'alan@gmail.com', '9874567890', '123 Maple St', '2023-01-15'),
('Robin', 'robin@gmail.com', '9125678901', '456 Oak St', '2023-03-10'),
('Davis', 'davis@gmail.com', '7336789012', '789 Pine St', '2023-05-05'),
('Preethi', 'preethi@gmail.com', '7377890123', '135 Elm St', '2023-06-20'),
('Charles', 'charles@gmail.com', '9548901234', '246 Cedar St', '2023-07-11'),
('Sethu', 'sethu@gmail.com', '9481234567', '999 Sunset Blvd', '2025-08-20');
```

Question:

Insert sample into BorrowingRecords table

Query:

```
INSERT INTO BorrowingRecords (MEMBER_ID, BOOK_ID, BORROW_DATE, RETURN_DATE)
VALUES
(1, 1, '2025-07-01 10:00:00', NULL),
(1, 2, '2025-06-01 09:00:00', '2025-06-20 12:00:00'),
(2, 3, '2025-07-15 14:00:00', NULL),
(3, 4, '2025-05-10 08:00:00', '2025-05-25 16:00:00'),
(4, 5, '2025-06-10 11:00:00', NULL),
(5, 1, '2025-07-20 15:00:00', NULL),
(1, 4, '2025-08-22 14:30:00', NULL),
(2, 1, '2025-08-01 10:00:00', NULL),
(3, 1, '2025-08-02 11:00:00', NULL),
(4, 1, '2025-08-03 12:00:00', NULL),
(5, 1, '2025-08-04 13:00:00', NULL),
(1, 1, '2025-08-05 14:00:00', NULL),
(2, 1, '2025-08-06 15:00:00', NULL),
(3, 1, '2025-08-07 16:00:00', NULL),
(4, 1, '2025-08-08 17:00:00', NULL);
```

Task 3 Information Retrieval:

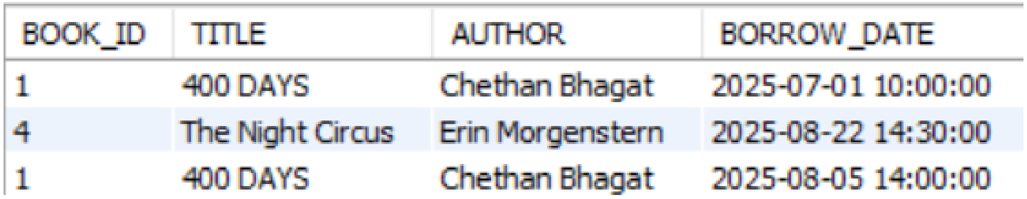
Question:

a) Retrieve a list of books currently borrowed by a specific member

Query:

```
SELECT b.BOOK_ID, b.TITLE, b.AUTHOR, br.BORROW_DATE

FROM BorrowingRecords br

JOIN Books b ON br.BOOK_ID = b.BOOK_ID

WHERE br.MEMBER_ID = 1

  AND br.RETURN_DATE IS NULL;
```

OUTPUT:



BOOK_ID	TITLE	AUTHOR	BORROW_DATE
1	400 DAYS	Chethan Bhagat	2025-07-01 10:00:00
4	The Night Circus	Erin Morgenstern	2025-08-22 14:30:00
1	400 DAYS	Chethan Bhagat	2025-08-05 14:00:00

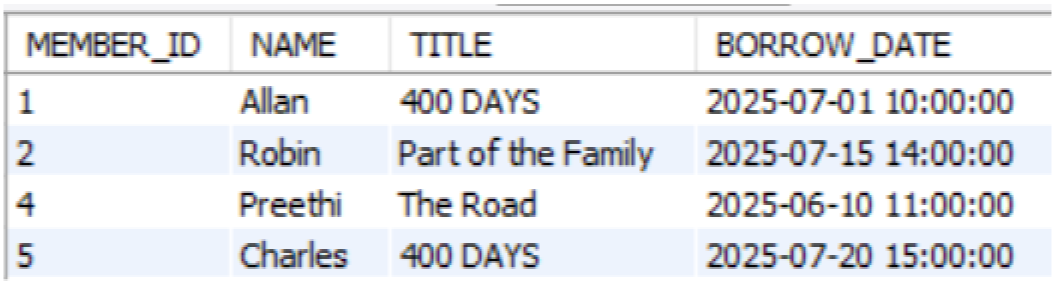
Question:

b) Find members who have overdue books (borrowed more than 30 days ago, not returned).

Query:

```
SELECT DISTINCT m.MEMBER_ID, m.NAME, b.TITLE, br.BORROW_DATE

FROM BorrowingRecords br

JOIN Members m ON br.MEMBER_ID = m.MEMBER_ID

JOIN Books b ON br.BOOK_ID = b.BOOK_ID

WHERE br.RETURN_DATE IS NULL

AND br.BORROW_DATE < NOW() - INTERVAL 30 DAY;
```

OUTPUT:



MEMBER_ID	NAME	TITLE	BORROW_DATE
1	Allan	400 DAYS	2025-07-01 10:00:00
2	Robin	Part of the Family	2025-07-15 14:00:00
4	Preethi	The Road	2025-06-10 11:00:00
5	Charles	400 DAYS	2025-07-20 15:00:00

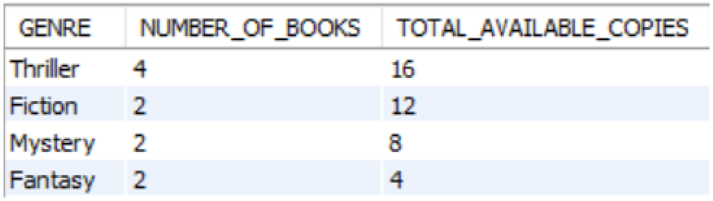
Question:

c) Retrieve books by genre along with the count of available copies.

Query:

```
SELECT GENRE,
COUNT(*) AS NUMBER_OF_BOOKS,
SUM(AVAILABLE_COPIES) AS TOTAL_AVAILABLE_COPIES
FROM Books
GROUP BY GENRE
ORDER BY 3 DESC;
```

OUTPUT:



GENRE	NUMBER_OF_BOOKS	TOTAL_AVAILABLE_COPIES
Thriller	4	16
Fiction	2	12
Mystery	2	8
Fantasy	2	4

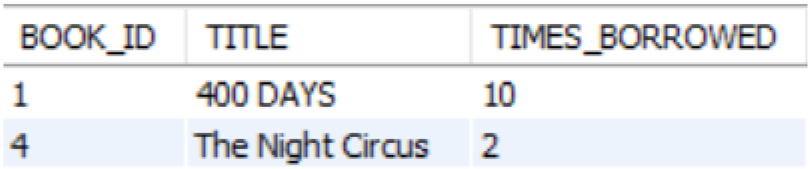
Question:

d) Find the most borrowed book(s) overall

Query:

```
SELECT b.BOOK_ID, b.TITLE, COUNT(*) AS TIMES_BORROWED
FROM BorrowingRecords br
JOIN Books b ON br.BOOK_ID = b.BOOK_ID
GROUP BY b.BOOK_ID, b.TITLE
ORDER BY TIMES_BORROWED DESC
LIMIT 2;
```

OUTPUT:



BOOK_ID	TITLE	TIMES_BORROWED
1	400 DAYS	10
4	The Night Circus	2

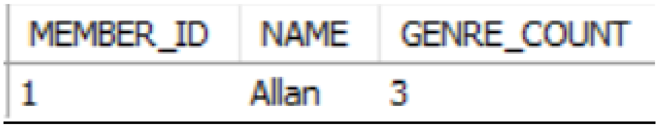
Question:

e) Retrieve members who have borrowed books from at least three different genres.

Query:

```
SELECT
m.MEMBER_ID,
m.NAME,
COUNT(DISTINCT b.GENRE) AS GENRE_COUNT
FROM
BorrowingRecords br
JOIN
Books b ON br.BOOK_ID = b.BOOK_ID
JOIN
Members m ON br.MEMBER_ID = m.MEMBER_ID
GROUP BY
m.MEMBER_ID, m.NAME
HAVING
COUNT(DISTINCT b.GENRE) >= 3;
```

OUTPUT:



MEMBER_ID	NAME	GENRE_COUNT
1	Allan	3

Task 4: Reporting and Analytics:

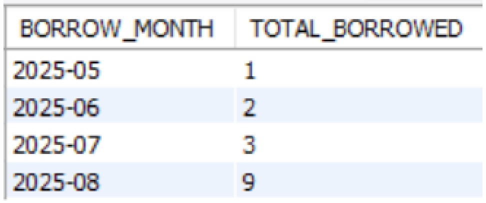
Question:

a) Calculate the total number of books borrowed per month.

Query:

```
SELECT DATE_FORMAT(BORROW_DATE, '%Y-%m') AS BORROW_MONTH, COUNT(*) AS
TOTAL_BORROWED
FROM BorrowingRecords
GROUP BY BORROW_MONTH
ORDER BY BORROW_MONTH;
```

OUTPUT:



BORROW_MONTH	TOTAL_BORROWED
2025-05	1
2025-06	2
2025-07	3
2025-08	9

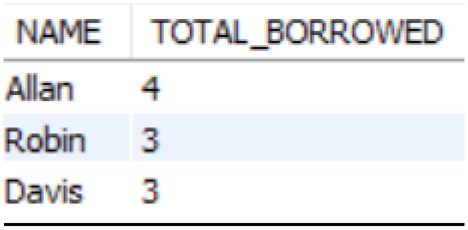
Question:

b) Find the top three most active members based on the number of books borrowed.

Query:

```
SELECT M.NAME, COUNT(*) AS TOTAL_BORROWED

FROM BorrowingRecords BR

JOIN Members M ON BR.MEMBER_ID = M.MEMBER_ID

GROUP BY M.MEMBER_ID, M.NAME

ORDER BY TOTAL_BORROWED DESC

LIMIT 3;
```

OUTPUT:



NAME	TOTAL_BORROWED
Allan	4
Robin	3
Davis	3

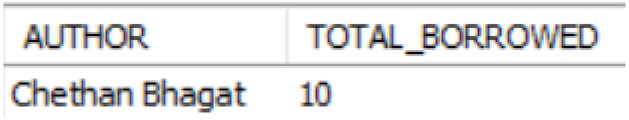
Question:

c) Retrieve authors whose books have been borrowed at least 10 times.

Query:

```
SELECT B.AUTHOR, COUNT(*) AS TOTAL_BORROWED
FROM BorrowingRecords BR
JOIN Books B ON BR.BOOK_ID = B.BOOK_ID
GROUP BY B.AUTHOR
HAVING COUNT(*) >= 10;
```

OUTPUT:



AUTHOR	TOTAL_BORROWED
Chethan Bhagat	10

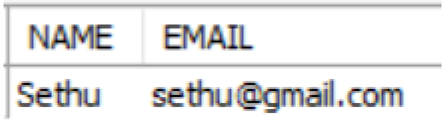
Question:

d) Identify members who have never borrowed a book

Query:

```
SELECT M.NAME, M.EMAIL
FROM Members M
LEFT JOIN BorrowingRecords BR ON M.MEMBER_ID = BR.MEMBER_ID
WHERE BR.BORROW_ID IS NULL;
```

OUTPUT:



NAME	EMAIL
Sethu	sethu@gmail.com