**Analyze the Books Table:**

1. **Count the Total Number of Books:**

SELECT COUNT(\*) AS total\_books FROM Books;

1. **Count the Number of Available Books:**

SELECT COUNT(\*) AS available\_books FROM Books WHERE availability = 1;

1. **Group Books by Genre:**

SELECT genre, COUNT(\*) AS count\_per\_genre FROM Books GROUP BY genre;

1. **List All Books by a Specific Author (e.g., 'George Orwell'):**

SELECT \* FROM Books WHERE author = 'George Orwell';

**Analyze the Members Table:**

1. **Count the Total Number of Members:**

SELECT COUNT(\*) AS total\_members FROM Members;

1. **List Members by Name:**

SELECT name, address, phone, email FROM Members ORDER BY name;

**Analyze the Transactions Table**

1. **Count the Total Number of Transactions:**

SELECT COUNT(\*) AS total\_transactions FROM Transactions;

1. **List Transactions by Borrow Date:**

SELECT \* FROM Transactions ORDER BY borrow\_date;

1. **Count Transactions Per Member:**

SELECT member\_id, COUNT(\*) AS transactions\_count FROM Transactions GROUP BY member\_id;

1. **List Overdue Books (assuming current date is '2023-06-01'):**

SELECT \* FROM Transactions WHERE due\_date < '2023-06-01' AND return\_date IS NULL;

**Complex Queries**

1. **List Books Currently Borrowed and Their Borrowers:**

SELECT b.title, m.name, t.borrow\_date, t.due\_date FROM Transactions t JOIN Books b ON t.book\_id = b.book\_id JOIN Members m ON t.member\_id = m.member\_id WHERE t.return\_date IS NULL;

1. **Find the Most Borrowed Book:**

SELECT b.title, COUNT(t.transaction\_id) AS borrow\_count FROM Transactions t JOIN Books b ON t.book\_id = b.book\_id GROUP BY b.title ORDER BY borrow\_count DESC LIMIT 1;

**Analysis made:**

A total of 3 books in the library.

All 3 books are currently available.

Books are evenly distributed across genres (Fiction, Dystopian, Classic).

2 members registered in the library.

2 transactions recorded.