SQL Master Class Project

Here's an example of a dataset you can use to create a library management system:

- Books Table:
    - book_id (Primary Key)
    - title
    - author
    - genre
    - publication_date
    - isbn
    - quantity_available
- Authors Table:
    - author_id (Primary Key)
    - author_name
- Members Table:
    - member_id (Primary Key)
    - member_name
    - member_email
    - member_phone
- Borrowings Table:
    - borrowing_id (Primary Key)
    - book_id (Foreign Key referencing Books table)
    - member_id (Foreign Key referencing Members table)
    - borrowing_date
    - return_date
    - is_returned

Books Table:

book_id	**title**	**author**	**genre**	**publication_date**	**isbn**	**quantity_available**

1	Harry Potter	J.K. Rowling	Fantasy	1997-06-26	9788700631625	5
2	To Kill a Mockingbird	Harper Lee	Fiction	1960-07-11	9780061120084	3
3	The Great Gatsby	F. Scott Fitzgerald	Classic	1925-04-10	9780743273565	2

Authors Table:

author_id	author_name
1	J.K. Rowling
2	Harper Lee
3	F. Scott Fitzgerald

Members Table:

member_id	member_name	member_email	member_phone
1	John Doe	john.doe@example.com	123-456-7890
2	Jane Smith	jane.smith@example.com	987-654-3210

Borrowings Table:

borrowing_id	book_id	member_id	borrowing_date	return_date	is_returned
1	1	1	2024-02-10	2024-02-20	true

2	2	2	2024-02-15	NULL	false

- Publishers Table:
    - publisher_id (Primary Key)
    - publisher_name
    - publisher_country
- Book Copies Table:
    - copy_id (Primary Key)
    - book_id (Foreign Key referencing Books table)
    - copy_number
    - condition
    - shelf_location
- Authors-Books Mapping Table:
    - author_book_id (Primary Key)
    - author_id (Foreign Key referencing Authors table)
    - book_id (Foreign Key referencing Books table)
- Reviews Table:
    - review_id (Primary Key)
    - book_id (Foreign Key referencing Books table)
    - member_id (Foreign Key referencing Members table)
    - rating
    - review_text
    - review_date
- Transactions Table:
    - transaction_id (Primary Key)
    - member_id (Foreign Key referencing Members table)
    - transaction_date
    - transaction_type (e.g., borrow, return, purchase)
    - amount_paid

Here's the expanded dataset:

Publishers Table:

publisher_id	publisher_name	publisher_country
1	Penguin Random House	United States
2	HarperCollins	United Kingdom
...	...	...

Book Copies Table:

copy_id	book_id	copy_number	condition	shelf_location
1	1	001	Good	A1
2	1	002	Fair	B3
...	...	...	...	...

Authors-Books Mapping Table:

author_book_id	author_id	book_id
1	1	1
2	2	2
...	...	...

Reviews Table:

review_id	book_id	member_id	rating	review_text	review_date
1	1	1	4.5	"A classic masterpiece."	2024-02-12
2	2	2	5.0	"Absolutely loved it!"	2024-02-18
...	...	...	...	...	...

Transactions Table:

transaction_id	member_id	transaction_date	transaction_type	amount_paid
1	1	2024-02-10	Borrow	0
2	2	2024-02-15	Borrow	0
...	...	...	...	...

After creating these tables perform the following queries

1. List all books borrowed by a specific member:
2. Find the most popular genres:
3. Identify books with the highest average rating:
4. List all members who have borrowed more than 5 books:
5. List all members who have borrowed less than 5 books:
6. Retrieve the top-rated books with at least 5 reviews:
7. Calculate the total revenue generated from book purchases:
8. List all books with their respective authors and publishers:
9. Find books that are currently available for borrowing:

10. Identify members who have overdue books:
11. List the top 10 most borrowed books:
12. Calculate the average number of days a book is borrowed for:
13. Find the total number of books published in each year:
14. Identify members who have borrowed books more than once:
15. List all books with their respective authors and average ratings:
16. Calculate the total number of copies available for each book:
17. Create a view of transaction table and provide privilege to another user. The user can view only member id and transaction date and privilege should be to select id who made transaction on any specific date.