1. List all books borrowed by a specific member:

```
select borrowing_id,member_id,books.book_id,books.title from borrowings
inner join books on borrowings.book_id=books.book_id;
```

2. Find the most popular genres:

```
select genre from books where book_id = (select count(*) from book_copies group by book_id order by book_id desc limit 1);
```

3. Identify books with the highest average rating:

```
select * from books where book_id= (select book_id from Reviews order by rating desc limit 1);
```

4. List all members who have borrowed more than 5 books:

```
select members.member_id,members.member_name,borrowings.borrowing_id,borrowings.book_id from members inner join borrowings on members.member_id=borrowings.member_id  where borrowing_id IN (select borrowing_id from borrowings group by borrowing_id having count(*) > 5);
```

5. List all members who have borrowed less than 5 books:

```
select members.member_id,members.member_name,borrowings.borrowing_id,borrowings.book_id from members inner join borrowings on members.member_id=borrowings.member_id  where borrowing_id IN (select borrowing_id from borrowings group by borrowing_id having count(*) < 5)
```

6. Retrieve the top-rated books with at least 5 reviews:

```
select title as top_rated_books from books where book_id IN( select book_id from  reviews where rating >= 5);
```

7. Calculate the total revenue generated from book purchases:

```
SELECT SUM(amount_paid) AS total_revenue FROM Transactions;
```

8. List all books with their respective authors and publishers:

```
SELECT books.book_id,books.title,authors.author_name,publisher_name FROM books

 inner join authors ON books.book_id = authors.author_id inner join publishers on
books.book_id=publishers.publisher_id;
```

9. Find books that are currently available for borrowing:

```
select title as currently_available from books inner join borrowings on books.book_id =
borrowings.book_id where return_date is not null;
```

10. Identify members who have overdue books:

```
select * from members where member_id=(select member_id from  borrowings where return_date
is null);
```

11. List the top 10 most borrowed books:

```
select * from books join borrowings on books.book_id = borrowings.book_id where

books.book_id  IN (select count(*) book_id from  borrowings group by book_id order by book_id
desc limit 10 );
```

12. Calculate the average number of days a book is borrowed for:

```
SELECT AVG(DATEDIFF(return_date,borrowing_date )) AS average_days

FROM Borrowings WHERE return_date IS NOT NULL;
```

13. Find the total number of books published in each year:

```
select count(*) ,publication_date from books group by publication_date;
```

14. Identify members who have borrowed books more than once:

```
select * from members where member_id IN(select  member_id from borrowings group by member_id having count(*) >1);
```

15. List all books with their respective authors and average ratings:

```
select books.book_id,books.title,authors.author_name,reviews.rating from books join authors on books.book_id = authors.author_id join reviews on books.book_id = reviews.book_id;
```

16. Calculate the total number of copies available for each book:

```
select count(*) ,book_id from book_copies group by book_id;
```

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.

```
create view transaction_view as select member_id,transaction_date from transactions;

create user "otheruser"@"localhost" identified by "password";

GRANT SELECT ON transaction_view TO "otheruser"@"localhost" ;
```