# SQL Project

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

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

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.

```sql
create database library;
create table Authours_book_mapping
(author_book_id int primary key,
author_id int not null,
foreign key(author_id) references author(author_id),
Book_id int);
create table Reviews
(Review_id int primary key,
book_id int,
foreign key(book_id) references books(book_id),
member_id int,
foreign key (member_id) references member(member_id),
rating int,
review_text varchar(50),
review_date varchar(50));
create table member
(member_id int,
member_name varchar(10),
ph_number  int);
create table Author
(author_id int primary key,
author_name varchar(10));
```

```
alter table author add index(author_id);
create table transactions
(transaction_id int primary key,
member_id int,
foreign key (member_id) references member(member_id),
transaction_date varchar(10),
transaction_type varchar(10),
amount_paid int NOT NULL);
select * from transactions;
insert into authours_book_mapping values
(1,1,1),
(2,2,2),
(3,3,3),
(4,4,4),
(5,5,5),
(6,6,6);
insert into author values
(1,'xxx'),
(2,'yyy'),
(3,'zzz'),
(4,'aaa'),
(5,'bbb'),
```

```
45      (5,'bbb'),
46      (6,'ccc');
47  •   insert into book_copies_table values
48      (1,1,001,'good','A1'),
49      (2,1,002,'Fair','B3'),
50      (3,2,003,"Bad",'C1'),
51      (4,2,003,'Good','d2'),
52      (5,3,001,'Fair','E1'),
53      (6,5,002,'Bad','f5'),
54      (7,6,001,'Bad','A4');
55  •   insert into books values
56      (1,'The Secret',1),
57      (2,'Ikigagi',2),
58      (3,'The hope',3),
59      (4,'Healer',4),
60      (5,'Love',5),
61      (6,'Mind',6),
62      (7,'Money',7);
63  •   insert into member values
64      (101,'Deepa',90846526),
65      (102,'Raju',92397428),
66      (103,'John',87294714),
```

```
    (103,'John',87294714),
    (104,'Ajay',94914651),
    (105,'Rekha',89849413),
    (106,'Vidya',921470),
    (107,'Aman',9741414);
insert into publishers values
    (1,'sss','India'),
    (2,'ssd','Japan'),
    (4,'asas','USA'),
    (3,'sss','UK'),
    (5,'aad','India'),
    (6,'WAS','Japan'),
    (7,'sas','Italy');
insert into reviews values
    (1,1,101,3,'A classic masterpiece',2024-2-3),
    (2,2,103,4.8,'Mindblowing',2024-2-12),
    (3,1,104,5,'Must read book',2023-12-1),
    (4,5,105,2,'Not a good one',2023-3-3),
    (5,6,106,4,'Loved it',2023-3-12),
    (6,7,102,2.3,'Hard to undestad',2024-1-1),
    (7,4,107,5,'Masterpiece',2024-2-3),
    (8,3,107,3,'A fair one',2023-12-12);
```

```
86      (7,4,107,5,'Masterpiece',2024-2-3),
87      (8,3,107,3,'A fair one',2023-12-12);
88  •   insert into transactions values
89      (1,101,2024-2-3,'Borrow',0),
90      (2,103,2024-3-2,'Borrow',0),
91      (3,102,2024-4-5,'Purchased',199),
92      (4,104,2023-12-23,'Purchased',299),
93      (5,106,2023-11-12,'borrow',0),
94      (6,107,2023-12-23,'Purchased',159),
95      (7,105,2023-3-4,'Borrow',50);
96  •   alter table books add published_year int;
97
98
99
100 •   SELECT Genere, COUNT(*) AS Genere_Count
101     FROM books
102     GROUP BY Genere
103     ORDER BY Genere_Count DESC;
104 •   select max(rating) as hieghest_rating
105     from reviews;
106 •   SELECT b.book_name, AVG(t.rating) AS avg_rating
107     FROM reviews t
108     JOIN books b ON t.book_id = b.book_id
109     GROUP BY b.book_id, b.book_name
110     ORDER BY avg_rating DESC;
111 •   select member_name from
112     member join transactions on
113     member.member_id=transactions.member_id
```

```sql
member.member_id=transactions.member_id
group by member.member_id,member.member_name
 having count(*)>5;
select member_name from
member join transactions on
member.member_id=transactions.member_id
group by member.member_id,member.member_name
 having count(*)<5;
SELECT book_name, AVG(rating) AS average_rating, COUNT(review_id) AS review_count
FROM books
JOIN reviews ON books.book_id = reviews.book_id
GROUP BY books.book_id, book_name
HAVING COUNT(review_id) >= 5
ORDER BY average_rating DESC;
SELECT SUM(amount_paid) AS total_revenue
FROM transactions;
SELECT books.book_name, author.author_name, publishers.publisher_name
FROM books
JOIN author ON books.author_name = author.author_name
JOIN publishers ON books.publisher_id = publishers.publisher_id;
SELECT book_name
FROM books
WHERE availability = 'yes';
alter table books add availability varchar(10);
update  books
set published_year=case
when book_id=1 then '1999'
when book_id=2 then '2000'
when book_id=3 then '2001'
```

```sql
    when book_id=3 then '2001'
    when book_id=4 then "1987"
    when book_id=5 then '2002'
    when book_id=6 then '1999'
    when book_id=7 then '2000'
    else availability end;
set sql_safe_updates=0;
select*from books;
SELECT member_id, member_name
FROM transactions
JOIN member ON transactions.member_id = member.member_id
WHERE transaction_date < CURRENT_DATE
AND transaction_date IS NOT NULL;
SELECT books.book_id, books.book_name, COUNT(*) AS borrow_count
FROM books join transactions on
books.book_id=transactions.book_id
GROUP BY books.book_id, books.book_name
ORDER BY borrow_count DESC
LIMIT 10;
SELECT AVG(transaction_daye) AS avg_borrow_duration
FROM transactions
WHERE return_date IS NOT NULL;
SELECT published_year AS publication_year, COUNT(*) AS total_books_published
FROM books
GROUP BY published_year;
SELECT member_id, COUNT(*) AS borrow_count
FROM transactions
WHERE transaction_type = 'borrow'
GROUP BY member_id
```

```sql
    books.book_id=transactions.book_id
    GROUP BY books.book_id, books.book_name
    ORDER BY borrow_count DESC
    LIMIT 10;
    SELECT AVG(transaction_daye) AS avg_borrow_duration
    FROM transactions
    WHERE return_date IS NOT NULL;
    SELECT published_year AS publication_year, COUNT(*) AS total_books_published
    FROM books
    GROUP BY published_year;
    SELECT member_id, COUNT(*) AS borrow_count
    FROM transactions
    WHERE transaction_type = 'borrow'
    GROUP BY member_id
    HAVING COUNT(*) > 1;
    select books.book_name,books.author_name, avg(reviews.rating) as average_rating
    from books join reviews
    on books.book_id=reviews.book_id
    group by books.book_id,books.author_name,books.book_name;
    SELECT b.book_name, SUM(bct.copy_number) AS total_no_copies
    FROM books b
    JOIN book_copies_table bct ON b.book_id = bct.book_id
    GROUP BY b.book_name;
    CREATE USER 'myuser'@'localhost' IDENTIFIED WITH mysql_native_password BY '12345';
    GRANT SELECT ON transaction_view TO myuser@localhost;
    GRANT SELECT (member_id) ON transaction_view TO myuser@localhost;
```