Library Management System

In Library Management System, I took five tables.
1.Librarian (who issue the books from library to member).
2.Member (who borrow books from library by librarian).
3.Book(In these table consists book details such as book name, author name, publisher date etc)
4.book_copies (stores book copy ids for every book in book table, available status)
5. Book_transactions (stores book borrow transaction from member by book copies)

Table Creation for all five Tables

```
CREATE TABLE librarian (
  librarian_id INT PRIMARY KEY,
  librarian_name VARCHAR(50) NOT NULL,
  librarian_email VARCHAR(50) NOT NULL,
  librarian_phone VARCHAR(20) NOT NULL
);

CREATE TABLE member (
  member_id INT PRIMARY KEY,
  member_name VARCHAR(50) NOT NULL,
  member_email VARCHAR(50) NOT NULL,
  member_phone VARCHAR(20) NOT NULL
);

CREATE TABLE book (
  book_id INT PRIMARY KEY,
  book_title VARCHAR(100) NOT NULL,
  book_author VARCHAR(50) NOT NULL,
  book_publisher VARCHAR(50) NOT NULL,
  book_published_date DATE NOT NULL,
  book_price decimal(10,2) NOT NULL DEFAULT 0.00
);

CREATE TABLE book_copies (
  book_copy_id INT PRIMARY KEY,
  book_id INT NOT NULL,
  is_available BOOLEAN NOT NULL DEFAULT true,
  FOREIGN KEY (book_id) REFERENCES book (book_id)
```

```
);

CREATE TABLE book_transactions (
  transaction_id INT PRIMARY KEY,
  book_copy_id INT NOT NULL,
  member_id INT NOT NULL,
  librarian_id INT NOT NULL,
  transaction_date DATE NOT NULL,
  due_date DATE NOT NULL,
  return_date DATE,
  due_fee DECIMAL(10,2) NOT NULL DEFAULT 0.00,
  due_fee_type  enum('lost','late_fee'),
  FOREIGN KEY (book_copy_id) REFERENCES book_copies (book_copy_id),
  FOREIGN KEY (member_id) REFERENCES member (member_id),
  FOREIGN KEY (librarian_id) REFERENCES librarian (librarian_id)
);
```

Insert Records in Tables

Inserting records in book table

```
INSERT INTO book (book_id, book_title, book_author, book_publisher, book_published_date, book_category)
VALUES
  (1, 'Harry Potter and the Philosopher''s Stone', 'J.K. Rowling', 'Bloomsbury Publishing', '1997-06-26', 'Fantasy'),
  (2, 'Harry Potter and the Chamber of Secrets', 'J.K. Rowling', 'Bloomsbury Publishing', '1998-07-02', 'Fantasy'),
  (3, 'Harry Potter and the Prisoner of Azkaban', 'J.K. Rowling', 'Bloomsbury Publishing', '1999-07-08', 'Fantasy'),
  (4, 'Harry Potter and the Goblet of Fire', 'J.K. Rowling', 'Bloomsbury Publishing', '2000-07-08', 'Fantasy'),
  (5, 'Harry Potter and the Order of Phoenix', 'J.K. Rowling', 'Bloomsbury Publishing', '2003-06-21', 'Fantasy'),
  (6, 'Harry Potter and the Half-Blood Prince', 'J.K. Rowling', 'Bloomsbury Publishing', '2005-07-16', 'Fantasy'),
  (7, 'Harry Potter and the Deathly Hallows', 'J.K. Rowling', 'Bloomsbury Publishing', '2007-07-21', 'Fantasy');
```

Inserting records in book_copies table

```
insert into book_copies( book_copy_id,book_id, is_available)
   values
   (1,1,1),
   (2,1,1),
```

```
(3,1,1),
(4,1,1),
(5,1,1),
(6,2,1),
(7,2,1),
(8,2,1),
(9,2,1),
(10,2,1),
(11,3,1),
(12,3,1),
(13,3,1),
(14,4,1),
(15,4,1),
(16,5,1),
(17,6,1),
(18,6,1),
(19,7,1),
(20,7,1);
```

Inserting records in librarian table

```
insert into librarian(librarian_id,librarian_name,librarian_email,librarian_phone)
  values
  (101,"Y Vasu" ,"yvasu@gmail.com","8919234678"),
  (102,"K Rajesh","Rajesh123@gmail.com","9912367543");
```

Inserting records in member table

```
insert into member(member_id,member_name,member_email,member_phone)
  values
  (1,"ysanathkumar","ysanathkumar@gmail.com","8769431209"),
  (2,"ksushant","kurraSushant@gmail.com","9987654321"),
  (3,"Ysairam","ysai1234@gmail.com","8790321567"),
  (4,"Yvenkat","venkatYadavilli@gmail.com","7890654321"),
  (5,"Nsanjay","sanjaySai@gmail.com","6789012345");
```

Different Queries perform on table

1. **Altering librarian and member table for adding constraint to phone number ?**

```
alter table librarian add constraint chk_phone_number_len check(length(librarian_phone) =10);

alter table member add constraint chk_member_phone_number_len check(length(member_phone)
=10);
```

2.Altering table by modifying ids in all tables to auto_increment ?

alter table librarian modify librarian_id varchar(10) AUTO_INCREMENT ;

alter table member modify member_id varchar(10) AUTO_INCREMENT;

alter table book modify book_id varchar(10) AUTO_INCREMENT;

alter table book_copies modify book_copy_id varchar(10) AUTO_INCREMENT;

alter table book_transactions modify transaction_id varchar (10)AUTO_INCREMENT;

3.check available copies for each book?

select b.book_title, count(bc.book_id) from book as b , book_copies as bc where b.book_id=bc.book_id and is_available=true group by bc.book_id;

4.Inserting records in book transactions table

insert into book_transactions(book_copy_id,member_id,librarian_id,transaction_date,due_date)
values(1,1,101,'2023-03-01', DATE_ADD('2023-03-01', INTERVAL 15 DAY));

insert into book_transactions(book_copy_id,member_id,librarian_id,transaction_date,due_date)
values(2,2,102,'2023-02-01', DATE_ADD('2023-02-01', INTERVAL 15 DAY));

5. update is_available column in book_copies after inserting records in book_transactions table?

Update book_copies set is_available=false where book_copies_id=1

Update book_copies set is_available=false where book_copies_id=2

6. check available book in copies tables along with book name from book table

select cb.book_id ,b.book_title,count(cb.is_available) as available_books from book b join book_copies cb on b.book_id = cb.book_id where cb.is_available = true group by cb.book_id;

7. check available bookcopies in copies tables by book_id colum in book_copies table

select book_copy_id from book_copies where book_id=1 and is_available=true;

8. check due date less than or under the present date ?

select * from book_transactions where due_date <now();

9. Calculating due_fee 10/-per day if member having due_date ?

update book_transactions set due_fee = DATEDIFF(curdate(),due_date)* 10,return_date=curdate() where due_date <now();

10.Display no.of books taken and total due_fee for member ?

select m.member_id,m.member_name,count(t.book_copy_id) as No_Of_Book_Taken,sum(t.due_fee) as Total_due from member as m join book_transactions as t on m.member_id =t.member_id group by member_id;

11. Display no of book issued by librarian ?

select l.librarian_name, count(t.librarian_id) as No_Of_book_Issued from librarian as l join book_transactions as t where l.librarian_id=t.librarian_id group by l.librarian_id ;

12. displaying all books borrowed by particular member ?

SELECT book.book_title

FROM book_transactions

JOIN book_copies ON book_transactions.book_copy_id = book_copies.book_copy_id

JOIN book ON book_copies.book_id = book.book_id

WHERE book_transactions.member_id = 2 AND book_transactions.return_date IS NULL;

13. delete particular book from book table by book_id ?

Delete from book where book_id =1;

14.delete particular book copy in book_copies table by id?

Delete from book_copies where book_copies_id=1;

15. delete record in book if foreign key is there ?

set foreign_key_checks=0;

delete from book where book_id =7;

set foreign_key_checks=1;

16. if member lost the copy book then set due_fee_type status to 'lost' and record delete in book_copie table ?

UPDATE book_transactions
SET due_fee_type = 'lost', due_fee = (select b.book_price from book as b join book_copies as bc on b.book_id =bc.book_id where bc. book_copy_id =1)
WHERE book_copy_id = 1;

```
DELETE FROM book_copies WHERE book_id = 1;
```