```
-- By Rohit kumar
Create database Library_Management_system;
use Library_Management_system;
#drop database Library_Management_system;


CREATE TABLE Authors (
      Authorid int primary key,
    AuthorName VARCHAR(255) NOT NULL,
    BirthDate varchar(20) CHECK (BirthDate < "2024-09-08")
);


CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    Genre VARCHAR(255),
    PublishedYear varchar(20) CHECK (PublishedYear <= "2024"),
    ISBN VARCHAR(13) UNIQUE,
    AuthorID INT,
    FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)
);


CREATE TABLE Members (
    MemberID INT PRIMARY KEY,
    MemberName VARCHAR(255) NOT NULL,
    JoinDate varchar(20) DEFAULT "2024-09-08",
    Email VARCHAR(255) UNIQUE NOT NULL
);
CREATE TABLE Loans (
    LoanID INT PRIMARY KEY,
    BookID INT,
    MemberID INT,
    LoanDate varchar(20) DEFAULT "2024-09-08",
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);
Insert into Authors values
(1, "Ram kumar", "1987-09-09"),
(2,"Ravi prasad", "1990- 10-21");
select * from Authors;
insert into Books
values
(1, "Data science ", "Textbook", "1987", "12345678", 1),
(2, "Sql","TextBook", "1990", "45678345", 2);
select * from Books;
insert into Members
values
(111, "Ram ","2020-09-10", "ram@gmail.com"),
(112, "Ravi", "2021-08-21", "ravi@gmail.com");
select * from Members;
insert into Loans
values
```

```
(21, 1, 111, "2021-09-08"),
(22, 2, 112, "2022-10-09");
select * from Loans;
select s1.MemberName as "Member Name", s3.Title as "Book Title" from
Members as s1  inner join Loans as s2 on
s1.MemberID = s2.MemberID inner
join Books as s3 on s2.BookID = s3.BookID;
select s1.*  from Members as s1 left join
Loans as s2 on s1.MemberID = s2.MemberID left
join Books as s3 on s2.BookID = s3.BookID;
select s1.* from Books as s1 right join
Loans as s2 on s1.BookID = s2.BookID;
select s1.* , s3.* from Members as s1 left join
Loans as s2 on s1.MemberID = s2.MemberID left
join Books as s3 on s2.BookID = s3.BookID union
select s1.* , s3.* from Members as s1 right join
Loans as s2 on s1.MemberID = s2.MemberID right
join Books as s3 on s2.BookID = s3.BookID ;
select * from Authors;
select a.Authorid, b.AuthorName from
Authors as a, Authors as b where
a.  Authorid = b.Authorid;
create view BorrowedBooks as
select s1.Title as "Book Title", s3.MemberName as "Member Name",
s2.LoanDate as "Loan Date" from Books as s1
inner join Loans as s2 on s1.BookID = s2.BookID
inner join Members as s3 on s2.MemberID = s3.MemberID;
#drop view BorrowedBooks;
#Delimiter $$
#create procedure InsertBook(find_BookID int,
#find_Title varchar(20),find_Genre varchar(20),
#find_PublishedYear varchar(20),find_ISBN varchar(20))
#begin
#insert into Books(BookID,Title,
#Genre, PublishedYear, ISBN) values(find_BookID,
#find_Title , find_Genre, find_PublishedYear, find_ISBN);
#end ;
#drop procedure InsertBook;
select * from Books;
#Delimiter $$
#create procedure UpdateBook
#(find_BookID int,find_title varchar(20))
#begin
#update  Books set Title = find_title
#where BookID = find_BookID;
#end;
select * from Books;
#Delimiter $$
#create procedure GetBookByNameAndID(find_BookName varchar(20),
find_BookId int)
#begin
#select BookID,  Title from Books where
#Title = find_BookName
#or BookID = find_BookId;
```

```
#end;
#drop procedure GetBookByNameAndID;
#select AuthorID, count(BookID) as "Number of Book"
#from Books group by AuthorID having count(BookID) < 2 ;
#select Title from Books where PublishedYear > "2000"  order by Title;
#select * from Books;
#select * from Authors;
update Authors set AuthorName = "J.K. Rowling"
where Authorid = 1;
update Books set Title = "Human being"
where Authorid = 1;
select Title from Books where Authorid in (select
Authorid from Authors where AuthorName like "J.K. Rowling");
#function member_function(member_name varchar(20), member_email
varchar(20))
#returns varchar(20)
#As
#begin
#SELECT
    #CONCAT(member_name, ' - ', member_email) AS member_info,
    #DATE_FORMAT(join_date, '%M %d, %Y') AS formatted_join_date
#FROM
    #members
#end;
create index genre_index on
Books(genre);
#drop index genre_index on Books;
select * from Books;
```