```
create database libarari;
use libarari;

CREATE TABLE Book (
    Book_No INT PRIMARY KEY,
    Book_Name VARCHAR(100),
    Author VARCHAR(100),
    Category VARCHAR(50),
    Price DECIMAL(8,2)
);

CREATE TABLE Member (
    Member_Id INT PRIMARY KEY,
    Member_Name VARCHAR(100),
    Member_Type VARCHAR(50),
    Membership_Fees DECIMAL(8,2),
    Max_Books_Allowed INT,
    Acc_Open_Date DATE,
    Penalty_Amount DECIMAL(8,2)
);

CREATE TABLE Issue (
    Lib_Issue_Id INT PRIMARY KEY,
    Member_Id INT,
    Book_No INT,
    Issue_Date DATE,
    Return_Date DATE,
    FOREIGN KEY (Member_Id) REFERENCES Member(Member_Id),
    FOREIGN KEY (Book_No) REFERENCES Book(Book_No)
);

INSERT INTO Book VALUES
```

```
(101, 'SQL Essentials', 'Loni', 'Database', 550),
(102, 'Advanced SQL', 'Scott Urman', 'Database', 750),
(103, 'PL/SQL Programming', 'Loni', 'RDBMS', 680),
(104, 'Data Structures', 'Narasimha', 'Science', 400),
(105, 'Fictional World', 'J.K. Rowling', 'Fiction', 620),
(106, 'DBMS Concepts', 'Loni', 'Database', 580),
(107, 'Management Basics', 'Khan', 'Management', 700),
(108, 'C Programming', 'Yashwant Kanetkar', 'Science', 350),
(109, 'Operating Systems', 'Galvin', 'Science', 620),
(110, 'SQL for Beginners', 'Loni', 'Database', 500);

table Book;

INSERT INTO Member VALUES
(201, 'Garima Sen', 'Lifetime', 1200, 5, '2006-01-10', 50),
(202, 'Richa Sharma', 'Regular', 800, 3, '2006-02-15', 0),
(203, 'Ajay Mehra', 'Regular', 800, 3, '2006-03-12', 20),
(204, 'Sunita Rao', 'Lifetime', 1200, 5, '2006-04-22', 10),
(205, 'Deepak Kumar', 'Regular', 700, 2, '2006-05-05', 0),
(206, 'Meena Jain', 'Lifetime', 1300, 5, '2006-06-25', 0),
(207, 'Raj Patel', 'Regular', 750, 3, '2006-07-20', 5),
(208, 'Kavita Desai', 'Regular', 850, 3, '2006-08-15', 0),
(209, 'Pooja Nair', 'Lifetime', 1400, 5, '2006-09-12', 0),
(210, 'Arun Joshi', 'Regular', 800, 3, '2006-10-10', 15);
table Member;

INSERT INTO Issue VALUES
(7001, 201, 101, '2006-12-01', NULL),
(7002, 202, 102, '2006-12-10', '2007-01-15'),
(7003, 203, 103, '2006-11-20', NULL),
(7004, 204, 104, '2006-07-18', '2006-08-10'),
(7005, 205, 105, '2006-12-22', NULL),
(7006, 206, 106, '2006-12-15', NULL),
(7007, 207, 107, '2006-10-05', '2006-10-25'),
(7008, 208, 108, '2006-08-14', '2006-09-01'),
(7009, 209, 109, '2006-06-10', '2006-07-12'),
(7010, 210, 110, '2006-12-25', NULL);
table Issue;

-- 1)List all the books that are written by Author Loni and has price  less then 600.
select * from Book where Author='Loni' and price <600;

-- 2)List the Issue details for the books that are not returned yet.
select *  from Issue where Return_Date  is null;

-- 3)Update blank Return_Date to '2006-12-31', excluding Book_No 7005 and 7006
update Issue set Return_Date="2006-12-31"
where Return_Date is null and Book_No not in(7005,7006);

-- 4)Issue details for books issued for more than 30 days
select *,DATEDIFF(Return_Date, Issue_Date) from Issue where DATEDIFF(Return_Date,
Issue_Date)>30;

-- 5)Books with price between 500 and 750 and category 'Database'
select * from Book where Price between 500 and 700  and Category='Database';

-- 6)Books from categories Science, Database, Fiction, Management
select * from Book where Category in('Database', 'Science', 'Fiction', 'Management');

-- 7)Members in descending order of penalty due
select * from Member order by Penalty_Amount desc;



-- 8) Update book 103's price to 300 and category to 'RDBMS'
update Book
```

```
set Price=300 , Category='RDBMS'
where Book_No=103;

-- 9)Books ordered by ascending category and descending price
select * from Book order by Category asc, Price desc;

-- 10)  Books with 'SQL' in the name
select * from Book where Book_Name like '% SQL%';

-- 11)Lib_Issue_Id, Issue_Date, Return_Date, No. of days book was issued
select Lib_Issue_Id, Issue_Date, Return_Date,DATEDIFF(Return_Date, Issue_Date) AS Days_Issued
from Issue;

-- 12)Members ordered by joining date
select * from Member order by Acc_Open_Date asc ;

-- 13)Total books issued to Member 101
select count(*) from Issue where Member_Id =101;


-- 14)Total penalty due for all members
select sum(Penalty_Amount) as total_penalty from Member;


use firstbit;
show tables;
table emp
-- create table users;
```