```
-- Create the database
CREATE DATABASE librarry ;

-- Use the database
USE librarry;

-- Create the Branch table
CREATE TABLE Branch (
    Branch_no INT PRIMARY KEY,
    Manager_Id INT,
    Branch_address VARCHAR(255),
    Contact_no VARCHAR(15)
);

-- Insert data into Branch table
INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES
(1, 101, ' Main St Ernakulam' , '04844567890'),
(2, 102, ' Sub St Calicut ', '0498765432'),
(3, 103, ' Sub st trivandram ', ' 0471820820 ' );


-- Select all data from the BRANCH table
SELECT * FROM BRANCH;

-- Create the Employee table
CREATE TABLE Employee (
    Emp_Id INT PRIMARY KEY,
    Emp_name VARCHAR(50),
    Position VARCHAR(50),
    Salary DECIMAL(10, 2),
    Branch_no INT,
    FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
);

-- Select all data from the EMPLOYEE table
SELECT * FROM EMPLOYEE;

insert into employee ( Emp_Id ,
    Emp_name ,
    Position ,
    Salary ,
    Branch_no
 ) value (101, 'Ram kumar' , 'manager', 100000, 1),
 (102, 'Stefen Devid' , 'manager', 90000, 2) ,
  (103, 'Gracy Thomus' , 'manager', 90000, 3),
 (104, 'Sreejith Sasi' , 'Accountant', 60000, 1),
  (105, 'Rajesh kumar' , 'Accountant', 60000, 2),
 (106, 'Ramya kumari' , 'Accountant', 60000, 3),
 (107, 'Syam Suresh' , 'sales man', 50000, 1) ,
  (108, 'Greeshma Thomus' , 'sales girl', 45000, 2),
 (109, 'Sreeja Sasi' , 'sales girl', 45000, 3),
  (110, 'Ratheesh kumar' , 'sales man', 45000, 1),
 (111, 'Roshni Devid' , 'sales girl', 45000, 2),(112, 'Roshan Devis' , 'sales man', 45000, 2);



-- Create the Books table
CREATE TABLE Books (
    ISBN INT PRIMARY KEY,
    Book_title VARCHAR(255),

    Category VARCHAR(50),

    Rental_Price DECIMAL(10, 2),

    Status VARCHAR(3),

    Author VARCHAR(50),

    Publisher VARCHAR(50)

);



-- Insert 20 books into the Books table

INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES

('1001000', 'Tharakans Grandhavari', 'Novel', 50.00, 'yes', 'Benyamin', 'DC Books'),

('1002000', '1984', 'Dystopian', 45.00, 'yes', 'George Orwell', 'Penguin Books'),

('1003000', 'Neuro Area', 'Crimr Thriller', 55.00, 'no', 'Sivan Edamana', 'DC Books'),

('1004000', 'Pride and Prejudice', 'Romance', 40.00, 'yes', 'Jane Austen', 'Penguin Books'),

('1005000', 'Brave New World', 'Dystopian', 50.00, 'no', 'Aldous Huxley', 'DC Books'),

('1006000', 'The Catcher in the Rye', 'Fiction', 45.00, 'yes', 'T D Antony', 'Little Books'),

('1007000', 'Crime and Punishment', 'Classic', 60.00, 'yes', 'Fyodor Dostoevsky', ' Classics Books'),

('1008000', 'Moby-Dick', 'Adventure', 55.00, 'no', 'Herman Melville', ' Classics Books'),

('1009000', 'War and Peace', 'Historical', 70.00, 'yes', 'Leo ', ' Classics Books'),

('2001000', 'The Brothers Karamazov', 'Classic', 65.00, 'no', 'A Grop of Author', 'DC Books'),

('2002000', 'The Grapes of Wrath', 'Fiction', 50.00, 'yes', 'John Steinbeck', 'Penguin Books'),

('2003000', 'Great Expectations', 'Classic', 45.00, 'yes', 'Charles Dickens', 'Penguin Books'),

('2004000', 'Anna Karenina', 'Romance', 60.00, 'no', 'Leo Tolstoy', 'Penguin Books'),

('2005000', 'Of Mice and Men', 'Fiction', 40.00, 'yes', 'John Steinbeck', 'Penguin Books'),

('2006000', 'Les Misérables', 'Historical', 70.00, 'no', 'Victor Hugo', 'Classics Books'),

('2007000', 'East of Eden', 'Fiction', 55.00, 'yes', 'John Steinbeck', 'Penguin Books'),

('2008000', 'The Odyssey', 'Epic', 65.00, 'yes', 'Homer', 'DC Books'),

('2009000', 'The Iliad', 'Epic', 60.00, 'no', 'Homer', ' DC Books'),
```

```sql
('3001000', 'Don Quixote', 'Adventure', 50.00, 'yes', 'Miguel de Cervantes', ' Classics Books'),
('3002000', 'The Divine Comedy', 'Epic', 70.00, 'no', 'Dante Alighieri', 'Classics Books');



-- Select all data from the BOOKS table
SELECT * FROM BOOKS;

-- Create the Customer table
CREATE TABLE Customer (
    Customer_Id INT PRIMARY KEY,
    Customer_name VARCHAR(50),
    Customer_address VARCHAR(255),
    Reg_date DATE
);

INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) VALUES
(1, 'John Devid', '123 Ernamkulam Street', '2021-05-15'),
(2, 'Jaims Smitha', '456 calicut Avenue', '2020-11-22'),
(3, 'Michael Johnson', '789 mavoor Road', '2022-03-10'),
(4, 'Emar Davis', '101 Map Lane trivandram', '2021-07-30'),
(5, 'Sheeja Mohan', '202 Beach road', '2019-12-05'),
(6, 'Sarah Muhammed', '303 Center Street', '2023-01-18'),
(7, 'James Tony', '404 Sqare Drive', '2020-09-25'),
(8, 'Jessica Munthaz', '505 mall roads', '2022-06-14'),
(9, 'Daniel parly', '606  Court road', '2021-04-02'),
(10, 'Lara Thomas', '707  beeach Circle', '2023-08-09');
```

```
-- Select all data from the CUCTOMER table
SELECT * FROM CUSTOMER;


-- Create the IssueStatus table
CREATE TABLE IssueStatus (
    Issue_Id INT PRIMARY KEY,
    Issued_cust INT,
    Issued_book_name VARCHAR(200),
    Issue_date DATE,
    Isbn_book INT,
    FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
);


-- Select all data from the ISSUESTATUS table
SELECT * FROM ISSUESTATUS;



INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book) VALUES
(1, 1, 'Tharakans Grandhavari', '2020-09-25', '1001000'),
(2, 2, '1984', '2022-06-14', '1002000'),
(3, 3, 'Neuro Area', '2021-04-02', '1003000'),
(4, 4, 'Pride and Prejudice', '2023-08-09', '1004000');
```

```
-- Create the ReturnStatus table

CREATE TABLE ReturnStatus (

    Return_Id INT PRIMARY KEY,

    Return_cust INT,

    Return_book_name VARCHAR(200),

    Return_date DATE,

    Isbn_book2 INT,

    FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)

);


INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)
VALUES

(1, 1, 'Tharakans Grandhavari', '2021-10-01', '1001000'),

(2, 2, '1984', '2022-07-01', '1002000');


-- Select all data from the RETURNSTATUS table

SELECT * FROM RETURNSTATUS;


-- Retrieve the book title, category, and rental price of all available book

SELECT Book_title, Category, Rental_Price

FROM Books

WHERE Status = 'yes';


-- List the employee names and their respective salaries in descending order of salary.

SELECT Emp_name, Salary

FROM Employee

ORDER BY Salary DESC;


-- Retrieve the book titles and the corresponding customers who have issued those books
```

```
SELECT B.Book_title, C.Customer_name
FROM IssueStatus I
JOIN Books B ON I.Isbn_book = B.ISBN
JOIN Customer C ON I.Issued_cust = C.Customer_Id;

-- Display the total count of books in each category.
SELECT Category, COUNT(*) AS Total_Books
FROM Books
GROUP BY Category;

-- Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000
SELECT Emp_name, Position
FROM Employee
WHERE Salary > 50000;

--  List the customer names who registered before 2022-01-01 and have not issued any books yet
SELECT C.Customer_name
FROM Customer C
LEFT JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust
WHERE C.Reg_date < '2022-01-01' AND I.Issue_Id IS NULL;

-- Display the branch numbers and the total count of employees in each branch.
SELECT Branch_no, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Branch_no;

-- Display the names of customers who have issued books in the month of June 2023.
SELECT DISTINCT C.Customer_name
```

```
FROM IssueStatus I
JOIN Customer C ON I.Issued_cust = C.Customer_Id
WHERE I.Issue_date BETWEEN '2023-06-01' AND '2023-06-30';


-- Retrieve book_title from book table containing history.
SELECT Book_title
FROM Books;


-- Retrieve the branch numbers along with the count of employees for branches having more than 5 employees
SELECT Branch_no, COUNT(*) AS Employee_Count
FROM Employee
GROUP BY Branch_no
HAVING COUNT(*) > 5;

-- Retrieve the branch numbers along with the count of employees for branches having more than 3 employees
SELECT Branch_no, COUNT(*) AS Employee_Count
FROM Employee
GROUP BY Branch_no
HAVING COUNT(*) > 3;

-- Retrieve the names of employees who manage branches and their respective branch addresses
SELECT E.Emp_name, B.Branch_address
FROM Employee E
JOIN Branch B ON E.Emp_Id = B.Manager_Id;
```

```
-- Display the names of customers who have issued books with a rental price higher than Rs. 25
SELECT DISTINCT C.Customer_name
FROM IssueStatus I
JOIN Books B ON I.Isbn_book = B.ISBN
JOIN Customer C ON I.Issued_cust = C.Customer_Id
WHERE B.Rental_Price > 25;
```