

```

CREATE DATABASE LIBRARY;
USE LIBRARY;

-- Creating the branch table
CREATE TABLE Branch(
    Branch_No INT PRIMARY KEY,
    Manager_Id INT,
    Branch_address VARCHAR(250),
    Contact_No BIGINT
);

-- inserting 10 rows
INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES
(1, 101, '123 Main St, Springfield', '5553451234'),
(2, 102, '456 Elm St, Springfield', '5556785678'),
(3, 103, '789 Oak St, Springfield', '5556548765'),
(4, 104, '321 Pine St, Springfield', '5558094321'),
(5, 105, '654 Maple St, Springfield', '5556543456'),
(6, 106, '987 Birch St, Springfield', '5555676543'),
(7, 107, '159 Cedar St, Springfield', '5554327890'),
(8, 108, '753 Willow St, Springfield', '5553450987'),
(9, 109, '258 Walnut St, Springfield', '5556712345'),
(10, 110, '369 Chestnut St, Springfield', '5553096789');

-- creating the employee table
CREATE TABLE Employee(
    Emp_Id INT PRIMARY KEY,
    Emp_name VARCHAR(100),
    Position VARCHAR(100),
    Salary DECIMAL(10, 2),
    Branch_No INT,
    FOREIGN KEY (Branch_No) REFERENCES Branch(Branch_No)
);

-- inserting 10 rows
INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES
(1, 'Alice Smith', 'Librarian', 45000, 1),
(2, 'Bob Johnson', 'Assistant Librarian', 35000, 1),
(3, 'Cathy Lee', 'Branch Manager', 60000, 2),
(4, 'David Brown', 'Librarian', 42000, 2),
(5, 'Eva Green', 'Librarian', 43000, 3),
(6, 'Frank White', 'Assistant Librarian', 34000, 3),
(7, 'Grace Hall', 'Branch Manager', 62000, 4),
(8, 'Hank Black', 'Librarian', 46000, 4),
(9, 'Ivy Blue', 'Librarian', 44000, 5),
(10, 'Jack Red', 'Assistant Librarian', 36000, 5);

-- creating the books table

```

```

CREATE TABLE Books(
    ISBN VARCHAR(20) PRIMARY KEY,
    Book_title VARCHAR(200),
    Category VARCHAR(100),
    Rental_Price DECIMAL(10, 2),
    Status ENUM('Yes', 'No'),
    Author VARCHAR(100),
    Publisher VARCHAR(100)
);

-- inserting 10 rows
INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author,
Publisher) VALUES
('978-3-16-148410-0', 'A Brief History of Time', 'Science', 5.00, 'yes',
'Stephen Hawking', 'Bantam'),
('978-3-16-148411-7', 'To Kill a Mockingbird', 'Fiction', 7.50, 'yes', 'Harper
Lee', 'J.B. Lippincott & Co.'),
('978-3-16-148412-4', '1984', 'Dystopian', 6.00, 'no', 'George Orwell',
'Secker & Warburg'),
('978-3-16-148413-1', 'Pride and Prejudice', 'Romance', 8.00, 'yes', 'Jane
Austen', 'T. Egerton'),
('978-3-16-148414-8', 'The Catcher in the Rye', 'Fiction', 9.00, 'yes', 'J.D.
Salinger', 'Little, Brown and Company'),
('978-3-16-148415-5', 'The Hobbit', 'Fantasy', 4.50, 'yes', 'J.R.R. Tolkien',
'George Allen & Unwin'),
('978-3-16-148416-2', 'Brave New World', 'Dystopian', 10.00, 'no', 'Aldous
Huxley', 'Chatto & Windus'),
('978-3-16-148417-9', 'Moby Dick', 'Adventure', 12.00, 'yes', 'Herman
Melville', 'Harper & Brothers'),
('978-3-16-148418-6', 'War and Peace', 'Historical Fiction', 11.00, 'yes',
'Leo Tolstoy', 'The Russian Messenger'),
('978-3-16-148419-3', 'The Picture of Dorian Gray', 'Fiction', 13.00, 'no',
'Oscar Wilde', 'Lippincott's Monthly Magazine');

-- creating the customer table
CREATE TABLE Customer(
    Customer_Id INT PRIMARY KEY,
    Customer_name VARCHAR(100),
    Customer_address VARCHAR(250),
    Reg_date DATE
);

-- inserting 10 rows
INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date)
VALUES
(1, 'Michael Scott', '100 Business Park, Scranton', '2023-01-01'),
(2, 'Jim Halpert', '200 Business Park, Scranton', '2023-01-02'),
(3, 'Pam Beesly', '300 Business Park, Scranton', '2023-01-03'),

```

```

(4, 'Dwight Schrute', '400 Business Park, Scranton', '2023-01-04'),
(5, 'Ryan Howard', '500 Business Park, Scranton', '2023-01-05'),
(6, 'Angela Martin', '600 Business Park, Scranton', '2023-01-06'),
(7, 'Stanley Hudson', '700 Business Park, Scranton', '2023-01-07'),
(8, 'Phyllis Vance', '800 Business Park, Scranton', '2023-01-08'),
(9, 'Toby Flenderson', '900 Business Park, Scranton', '2021-01-09'),
(10, 'Kelly Kapoor', '1000 Business Park, Scranton', '2023-01-10');

-- creating the issueStatus table
CREATE TABLE IssueStatus(
    Issue_Id INT PRIMARY KEY,
    Issued_cust INT,
    Issued_book_name VARCHAR(250),
    Issue_date DATE,
    Isbn_book VARCHAR(20),
    FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
);

-- insering 10 rows
INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date,
Isbn_book) VALUES
(1, 1, '', '2023-01-01', '978-3-16-148410-0'),
(2, 2, 'To Kill a Mockingbird', '2023-02-02', '978-3-16-148411-7'),
(3, 3, '1984', '2023-02-03', '978-3-16-148412-4'),
(4, 4, 'Pride and Prejudice', '2023-02-04', '978-3-16-148413-1'),
(5, 5, 'The Catcher in the Rye', '2023-02-05', '978-3-16-148414-8'),
(6, 6, 'The Hobbit', '2023-06-06', '978-3-16-148415-5'),
(7, 7, 'Brave New World', '2023-06-07', '978-3-16-148416-2'),
(8, 8, 'Moby Dick', '2023-07-08', '978-3-16-148417-9'),
(9, 9, 'War and Peace', '2023-08-09', '978-3-16-148418-6'),
(10, 10, 'The Picture of Dorian Gray', '2023-09-10', '978-3-16-148419-3');

-- creating the ReturnStatus table
CREATE TABLE ReturnStatus(
    Return_Id INT PRIMARY KEY,
    Return_cust INT,
    Return_book_name VARCHAR(250),
    Return_date DATE,
    Isbn_book2 VARCHAR(20),
    FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
);

-- insering 10 rows
INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name,
Return_date, Isbn_book2) VALUES

```

```

(1, 1, 'A Brief History of Time', '2023-03-01', '978-3-16-148410-0'),
(2, 2, 'To Kill a Mockingbird', '2023-03-02', '978-3-16-148411-7'),
(3, 3, '1984', '2023-03-03', '978-3-16-148412-4'),
(4, 4, 'Pride and Prejudice', '2023-03-04', '978-3-16-148413-1'),
(5, 5, 'The Catcher in the Rye', '2023-03-05', '978-3-16-148414-8'),
(6, 6, 'The Hobbit', '2023-03-06', '978-3-16-148415-5'),
(7, 7, 'Brave New World', '2023-03-07', '978-3-16-148416-2'),
(8, 8, 'Moby Dick', '2023-03-08', '978-3-16-148417-9'),
(9, 9, 'War and Peace', '2023-03-09', '978-3-16-148418-6'),
(10, 10, 'The Picture of Dorian Gray', '2023-03-10', '978-3-16-148419-3');

-- Retrieve the book title, category, and rental price of all available books.
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
    Books.Book_title,
    Customer.Customer_name
FROM
    IssueStatus
JOIN
    Customer ON IssueStatus.Issued_cust = Customer.Customer_Id
JOIN
    Books ON IssueStatus.Isbn_book = Books.ISBN;

-- 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
    Customer.Customer_name
FROM
    Customer
LEFT JOIN
    IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
WHERE
    Customer.Reg_date < '2022-01-01'
    AND IssueStatus.Issue_Id IS NULL;

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

-- Display the name of customers who have issued books in the month of j 2023
SELECT DISTINCT C.Customer_name
FROM Customer C
JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust
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
WHERE Book_title LIKE '%history%';

-- Retrieve the branch numbers along with the count of employees for branches
having more than 1 employees

SELECT Branch_no, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Branch_no
HAVING COUNT(*) > 1;

-- 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.Branch_no = B.Branch_no
WHERE E.Position = 'Branch Manager';

```

-- Display the names of customers who have issued books with a rental price higher than Rs. 10.

```
SELECT DISTINCT C.Customer_name
FROM Customer C
JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust
JOIN Books B ON I.ISBN_book = B.ISBN
WHERE B.Rental_price > 10;
```