Library Management System

Create Database:

```
CREATE DATABASE Library;
USE Library;
```

Create Table:

```
CREATE TABLE Branch (
Branch_No INT PRIMARY KEY,
Manager_Id VARCHAR(50),
Branch_Address VARCHAR(100),
Contact_No VARCHAR(20) NOT NULL
);
```

Branch_No	Manager_Id	Branch_Address	Contact_No
1	101	123 Elm Street, City A	1234567890
2	102	456 Maple Avenue, City B	9876543210
3	103	789 Oak Boulevard, City C	1122334455
4	104	321 Pine Road, City D	9988776655
5	105	654 Birch Lane, City E	8877665544
6	106	987 Cedar Drive, City F	7766554433

Branch 3

```
CREATE TABLE Employee (
Emp_Id VARCHAR(50) PRIMARY KEY,
Emp_Name VARCHAR(50),
Position VARCHAR(50),
Salary DECIMAL(10,2),
Branch_No INT,
FOREIGN KEY (Branch_No) REFERENCES Branch(Branch_No)
);
```

Emp_Id	Emp_Name	Position	Salary	Branch_No
201	Alice	Librarian	35000.00	1
202	Bob	Assistant Librarian	30000.00	2
203	Charlie	Manager	45000.00	3
204	Diana	Clerk	25000.00	4
205	Eve	Manager	50000.00	5
206	Frank	Clerk	24000.00	6

Employee 4

```
CREATE TABLE Books (
ISBN VARCHAR(50) PRIMARY KEY,
Book_Title VARCHAR(100),
Category VARCHAR(100),
Rental_Price DECIMAL(10,2),
Status ENUM('Yes' , 'No') NOT NULL,
Author VARCHAR(100),
Publisher VARCHAR(100)
);
```

ISBN	Book_Title	Category	Rental_Price	Status	Author	Publisher
9780001	Book One	Fiction	15.00	Yes	Author A	Publisher X
9780002	Book Two	Non-Fiction	20.00	Yes	Author B	Publisher Y
9780003	Book Three	Science	25.00	No	Author C	Publisher Z
9780004	Book Four	History	18.00	Yes	Author D	Publisher X
9780005	Book Five	Biography	22.00	No	Author E	Publisher Y
9780006	Book Six	Fiction	17.00	Yes	Author F	Publisher Z

Books 5

```
CREATE TABLE Customer (
Customer_Id VARCHAR(50) PRIMARY KEY,
Customer_Name VARCHAR(50),
Customer_Address VARCHAR(100),
Reg_Date date);
```

Customer_Id	Customer_Name	Customer_Address	Reg_Date
301	Tom	789 Hill Road, City K	2024-01-01
302	Jerry	123 Sea Lane, City L	2024-01-05
303	Mickey	456 Valley Street, City M	2024-01-10
304	Minnie	789 River Boulevard, City N	2024-01-15
305	Donald	321 Forest Drive, City O	2024-01-20
306	Daisy	654 Desert Path, City P	2024-01-25

Customer 6

```
CREATE TABLE Issue_Status (
Issue_Id VARCHAR(50) PRIMARY KEY,
Issued_Cust VARCHAR(50),
Issued_Book_Name VARCHAR(100),
Issue_Date DATE,
ISBN_Book VARCHAR(50),
FOREIGN KEY (Issued_Cust) REFERENCES Customer(Customer_Id),
FOREIGN KEY (ISBN_Book) REFERENCES Books(ISBN)
);
```

Issue_Id	Issued_Cust	Issued_Book_Name	Issue_Date	ISBN_Book
401	301	Book One	2024-03-01	9780001
402	302	Book Two	2024-03-02	9780002
403	303	Book Three	2024-03-03	9780003
404	304	Book Four	2024-03-04	9780004
405	305	Book Five	2024-03-05	9780005
406	306	Book Six	2024-03-06	9780006

Issue_Status 7

```
CREATE TABLE Return_Status (
Return_Id VARCHAR(50) PRIMARY KEY,
Return_Cust VARCHAR(50),
Return_Book_Name VARCHAR(100),
Return_date DATE,
ISBN_Book2 VARCHAR(50),
FOREIGN KEY (ISBN_Book2) REFERENCES Books(ISBN),
FOREIGN KEY (Return_Cust) REFERENCES Customer(Customer_Id)
);
```

Return_Id	Return_Cust	Return_Book_Name	Return_date	ISBN_Book2
501	301	Book One	2024-03-15	9780001
502	302	Book Two	2024-03-16	9780002
503	303	Book Three	2024-03-17	9780003
504	304	Book Four	2024-03-18	9780004
505	305	Book Five	2024-03-19	9780005
506	306	Book Six	2024-03-20	9780006

Return_Status 8

1. Retrieve the book title, category, and rental price of all available books:

```
SELECT book_title, Category, Rental_Price FROM Books WHERE Status = 'Yes';
```

Output:

book_title	Category	Rental_Price
Book One	Fiction	15.00
Book Two	Non-Fiction	20.00
Book Four	History	18.00
Book Six	Fiction	17.00
Book Seven	Fiction	19.00
Book Nine	Science	23.00

Books 1

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

```
SELECT Emp_Name, Salary FROM Employee ORDER BY Salary DESC;
```

Output:

Emp_Name	Salary
Eve	50000.00
Judy	47000.00
Charlie	45000.00
Grace	38000.00
Alice	35000.00
Heidi	31000.00

Employee 2

3. Retrieve the book titles and the corresponding customers who have issued those books:

```
SELECT Books.Book_title AS Book_Title,
Customer.Customer_name AS Customer_Name
FROM
Issue_Status
JOIN
Books ON Issue_Status.Isbn_book = Books.ISBN
JOIN
Customer ON Issue_Status.Issued_cust = Customer.Customer_Id;
```

Output:

Book_Title	Customer_Name
Book One	Tom
Book Two	Jerry
Book Three	Mickey
Book Four	Minnie
Book Five	Donald
Book Six	Daisy
Book Seven	Goofy

Result 3 ×

4. Display the total count of books in each category:

```
SELECT Category, COUNT(*) AS Total_Books
FROM Books
GROUP BY Category;
```

Output:

Category	Total_Books
Fiction	3
Non-Fiction	2
Science	2
History	2
Biography	1

5. 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;
```

Output:

Emp_name	Position

Employee 6 × There is no Employee with salary above 50000.

6. 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 Issue_Status ON Customer.Customer_Id = Issue_Status.Issued_cust
WHERE Customer.Reg_date < '2022-01-01' AND Issue_Status.Issue_Id IS NULL;
```

Output:

Customer_name

Result 7 × There is no customer who Reg. Before 2022-01-01 and have not issued any books yet.

7. 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;
```

Output:

Branch_no	Total_Employees
1	1
2	1
3	1
4	1
5	1
6	1
7	1

Result 8 ×

8. Display the names of customers who have issued books in the month of June 2023:

```
SELECT DISTINCT Customer.Customer_name
FROM Issue_Status
JOIN Customer ON Issue_Status.Issued_cust = Customer.Customer_Id
WHERE Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
```

Output:

No customer exists with the above criteria.

9. Retrieve book_title from book table containing history:

```
SELECT Book_title
FROM Books
WHERE Book_title LIKE '%History%';
```

Output:

No books having word History in title.

10. Retrieve the branch numbers along with the count of employees for branches having more than 5 employees:

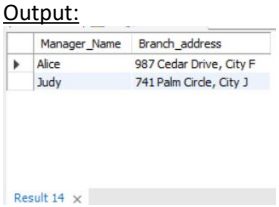
```
SELECT Branch_no, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Branch_no
HAVING COUNT(*) > 5;
```

Output:

11. Retrieve the names of employees who manage branches and their respective branch addresses:

```
SELECT Employee.Emp_name AS Manager_Name, Branch.Branch_address
FROM Branch
JOIN Employee ON Branch.Manager_Id = Employee.Emp_Id;
```

Output:



12. Display the names of customers who have issued books with a rental price higher than Rs. 25:

```
SELECT DISTINCT Customer.Customer_name,Books.Rental_price
FROM Issue_Status
JOIN Books ON Issue_Status.Isbn_book = Books.ISBN
JOIN Customer ON Issue_Status.Issued_cust = Customer.Customer_Id
WHERE Books.Rental_Price > 25;
```

Output: