Library Management System

To build a project based on Library Management System. It keeps track of all information about books in the library, their cost, status and total number of books available in the library.

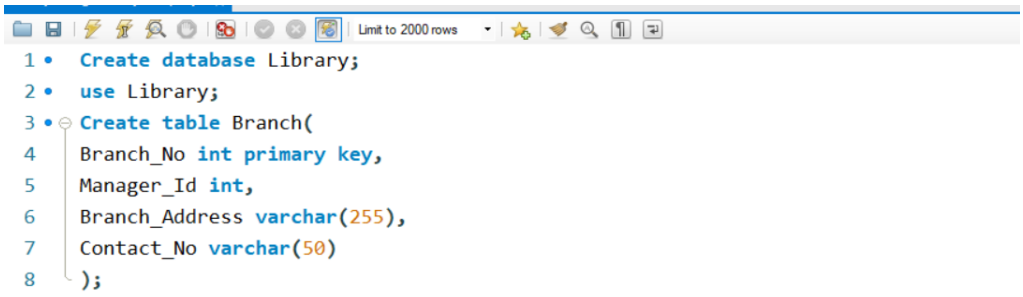
Create a database named library and following TABLES in the database:

1. Branch
2. Employee
3. Books
4. Customer
5. IssueStatus
6. ReturnStatus

Attributes for the tables:

1. **Branch**

- Branch_no - Set as PRIMARY KEY
- Manager_Id
- Branch_address
- Contact_no



2. **Employee**

- Emp_Id – Set as PRIMARY KEY
- Emp_name
- Position
- Salary
- Branch_no - Set as FOREIGN KEY and it refer Branch_no in Branch table

```sql
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)
);
```

3. **Books**

- ISBN - Set as PRIMARY KEY
- Book_title
- Category
- Rental_Price
- Status [Give yes if book available and no if book not available]
- Author
- Publisher

```sql
Create table Books(
ISBN varchar(25) primary key,
Book_Title varchar(50),
Category varchar(20),
Rental_Price decimal(5,2),
Status char(3),
Author varchar(20),
Publisher varchar(50)
);
```

4. **Customer**

- Customer_Id - Set as PRIMARY KEY
- Customer_name
- Customer_address
- Reg_date

```sql
Create table Customer(
Customer_Id int primary key,
Customer_Name varchar(50),
Customer_Address varchar(50),
Reg_Date date
);
```

5. IssueStatus

- Issue_Id - Set as PRIMARY KEY
- Issued_cust – Set as FOREIGN KEY and it refer customer_id in CUSTOMER table Issued_book_name
- Issue_date
- Isbn_book – Set as FOREIGN KEY and it should refer isbn in BOOKS table

```
124 • Create table IssueStatus(
125   Issue_Id int primary key,
126   Issued_Cust int,
127   Issued_Book_Name varchar(50),
128   Issue_Date date,
129   Isbn_Book varchar(50),
130   foreign key(Issued_Cust) references Customer(Customer_Id),
131   foreign key(Isbn_Book) references Books(ISBN)
132   );
```

6. ReturnStatus

- Return_Id - Set as PRIMARY KEY
- Return_cust
- Return_book_name
- Return_date
- Isbn_book2 - Set as FOREIGN KEY and it should refer isbn in BOOKS table

```
147 • Create table ReturnStatus(
148   Return_Id int primary key,
149   Return_Cust int,
150   Return_Book_Name varchar(50),
151   Return_Date date,
152   Isbn_Book2 varchar(25),
153   foreign key(Isbn_Book2) references Books(isbn)
154   );
```

Queries

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

```
Select Book_Title, Category, Rental_Price from Books;
```

```
180    -- ********************************************************************************
181
182    -- 1. Retrieve the book title, category, and rental price of all available books.
183 •  Select Book_Title, Category, Rental_Price from Books;
184
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_Title	Category	Rental_Price
To Kill a Mockingbird	Fiction	4.50
War and Peace	Historical	7.00
The Da Vinci Code	Thriller	5.50
The Hobbit	Fantasy	5.25
The Catcher in the Rye	Fiction	4.75
Pride and Prejudice	Romance	4.00

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

```
Select Emp_Name, Salary
From Employee
Order By salary desc;
```

```
184
185    -- 2. List the employee names and their respective salaries in descending order of salary.
186 •  Select Emp_Name, Salary
187    From Employee
188    Order By salary desc;
189
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Emp_Name	Salary
Kevin Hughes	62000.00
Hannah Wood	61000.00
Wendy Hall	60000.00
Oscar Rogers	60000.00
Rachel Turner	60000.00
Paul Adams	59000.00

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

```
Select IssueStatus.Issued_Book_Name, Customer.Customer_Name
from IssueStatus
JOIN Customer on Issuestatus.Issued_cust = customer.customer_id;
```

```
189
190    -- 3. Retrieve the book titles and the corresponding customers who have issued those books.
191 •  Select IssueStatus.Issued_Book_Name, Customer.Customer_Name
192    from IssueStatus
193    JOIN Customer on Issuestatus.Issued_cust = customer.customer_id;
194
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Issued_Book_Name	Customer_Name
The Great Gatsby	John Doe
1984	Jane Smith
The Catcher in the Rye	Alice James
Animal Farm	Bob Brown
The Hobbit	Carol White
To Kill a Mockingbird	David Green

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

```
Select Category, count(*) AS Total_Counts
from books
group by category;
```

```
194
195    -- 4. Display the total count of books in each category.
196 •  Select Category, count(*) AS Total_Counts
197    from books
198    group by category;
199
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Category	Total_Counts
Fiction	3
Historical	1
Thriller	1
Fantasy	1
Romance	2
Dystopian	2

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;
```

```
199
200    -- 5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
201 •  Select Emp_Name, Position
202    from Employee
203    Where Salary > 50000;
204
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Emp_Name	Position
Carol Davis	Librarian
Eva Brown	Librarian
Grace Lee	Librarian
Karen White	Librarian
Mona Black	Librarian
Paul Adams	Librarian

6. List the customer names who registered before 2023-04-01 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 < '2023-04-01';
```

```
204
205    -- 6. List the customer names who registered before 2023-04-01 and have not issued any books yet.
206 •  Select c.customer_name
207    from customer c
208    LEFT JOIN issuestatus i
209    on c.customer_id = i.issued_cust
210    where c.reg_date < '2023-04-01';
211
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_name
John Doe
Jane Smith
Alice James

7. Display the branch numbers and the total count of employees in each branch.

```
Select  Branch_No, count(*) As Total_Employee
from employee
group by branch_no;
```

```
211
212    -- 7. Display the branch numbers and the total count of employees in each branch.
213 •  Select  Branch_No, count(*) As Total_Employee
214    from employee
215    group by branch_no;
216
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Branch_No	Total_Employee
1	8
2	4
3	4
4	5
5	4
6	4

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

```
Select c.customer_name
from customer c
inner join issuestatus i on c.customer_id = i.issued_cust
where issue_date between '2023-06-01' and '2023-06-30';
```

```
216
217    -- 8. Display the names of customers who have issued books in the month of June 2023.
218 •  Select c.customer_name
219    from customer c
220    inner join issuestatus i on c.customer_id = i.issued_cust
221    where issue_date between '2023-06-01' and '2023-06-30';
222
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_name
John Doe

```
222
223    -- 9. Retrieve book_title from book table containing Romance.
224 •  Select Book_title
225    from Books
226    where Category Like '%Romance%';
227
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_title
Pride and Prejudice
Jane Eyre

9. Retrieve book_title from book table containing Romance.

```
Select Book_title
from Books
where Category Like '%Romance%';
```

```
227
228    -- 10.Retrieve the branch numbers along with the count of employees for branches having more than 5 employees
229 •  Select Branch_No, count(*) As Total_Employee
230    from employee
231    group by branch_no
232    Having (Total_Employee) > 5;
233
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Branch_No	Total_Employee
1	8
7	7

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

```
Select Branch_No, count(*) As Total_Employee
from employee
group by branch_no
Having (Total_Employee) > 5;
```

```
233
234     -- 11. Retrieve the names of employees who manage branches and their respective branch addresses.
235 •   Select e.Emp_name As Manager_Name, b.Branch_Address
236     from Branch b
237     join employee e
238     on b.manager_id = e.emp_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Manager_Name	Branch_Address
Alice Johnson	123 Main St, Cityville
Carol Davis	456 Elm St, Townsville
Eva Brown	789 Oak St, Villageville
Grace Lee	321 Pine St, Hamletville
Ivy Clark	654 Maple St, Boroughville
Karen White	987 Birch St, Suburbville

12. Display the names of customers who have issued books with a rental price higher than dollar 5.

```
Select 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 > 5;
```

```
40     -- 12.  Display the names of customers who have issued books with a rental price higher than dollar 5.
41 •   Select c.customer_name
42     from customer c
43     join IssueStatus i on c.customer_id = i.issued_cust
44     join Books b on i.isbn_book = b.isbn
45     where b.rental_price > 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_name
Grace Red
Henry Orange
Carol White
Jane Smith