You are going 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

```
1      -- Create the database
2 •    CREATE DATABASE library;
3 •    USE library;
```

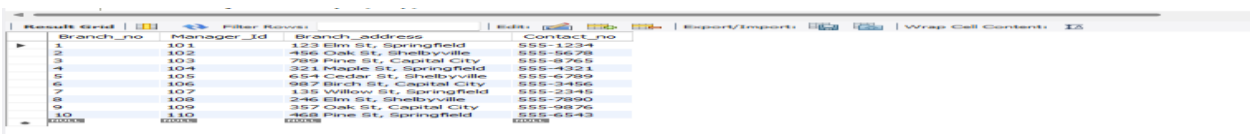
Attributes for the tables:

1. Branch

• Branch_no

- Set as PRIMARY KEY
  • Manager_Id
  • Branch_address
  • Contact_no

```
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, '123 Elm St, Springfield', '555-1234'),
(2, 102, '456 Oak St, Shelbyville', '555-5678'),
(3, 103, '789 Pine St, Capital City', '555-8765'),
(4, 104, '321 Maple St, Springfield', '555-4321'),
(5, 105, '654 Cedar St, Shelbyville', '555-6789'),
(6, 106, '987 Birch St, Capital City', '555-3456'),
(7, 107, '135 Willow St, Springfield', '555-2345'),
(8, 108, '246 Elm St, Shelbyville', '555-7890'),
(9, 109, '357 Oak St, Capital City', '555-9876'),
(10, 110, '468 Pine St, Springfield', '555-6543');
```



Branch_no	Manager_Id	Branch_address	Contact_no
1	101	123 Elm St, Springfield	555-1234
2	102	456 Oak St, Shelbyville	555-5678
3	103	789 Pine St, Capital City	555-8765
4	104	321 Maple St, Springfield	555-4321
5	105	654 Cedar St, Shelbyville	555-6789
6	106	987 Birch St, Capital City	555-3456
7	107	135 Willow St, Springfield	555-2345
8	108	246 Elm St, Shelbyville	555-7890
9	109	357 Oak St, Capital City	555-9876
10	110	468 Pine St, Springfield	555-6543

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

```
module2- Library Management...
Limit to 1000 rows
25  CREATE TABLE Employee (
26      Emp_Id INT PRIMARY KEY,
27      Emp_name VARCHAR(100),
28      Position VARCHAR(50),
29      Salary DECIMAL(10, 2),
30      Branch_no INT,
31      FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
32  );
33  -- Insert data into Employee table
34  INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES
35  (1, 'Alice Johnson', 'Manager', 60000.00, 1),
36  (2, 'Bob Smith', 'Assistant', 45000.00, 1),
37  (3, 'Charlie Brown', 'Manager', 65000.00, 2),
38  (4, 'Daisy Wilson', 'Assistant', 42000.00, 2),
39  (5, 'Eva Adams', 'Manager', 70000.00, 3),
40  (6, 'Frank Moore', 'Assistant', 46000.00, 3),
41  (7, 'Grace Lee', 'Manager', 62000.00, 4),
42  (8, 'Henry Clark', 'Assistant', 43000.00, 4),
43  (9, 'Ivy Davis', 'Manager', 68000.00, 5),
44  (10, 'Jack Walker', 'Assistant', 44000.00, 5);
45
```

```
Limit to 1000 rows
41  (7, 'Grace Lee', 'Manager', 62000.00, 4),
42  (8, 'Henry Clark', 'Assistant', 43000.00, 4),
43  (9, 'Ivy Davis', 'Manager', 68000.00, 5),
44  (10, 'Jack Walker', 'Assistant', 44000.00, 5);
45  select * from employee;
```

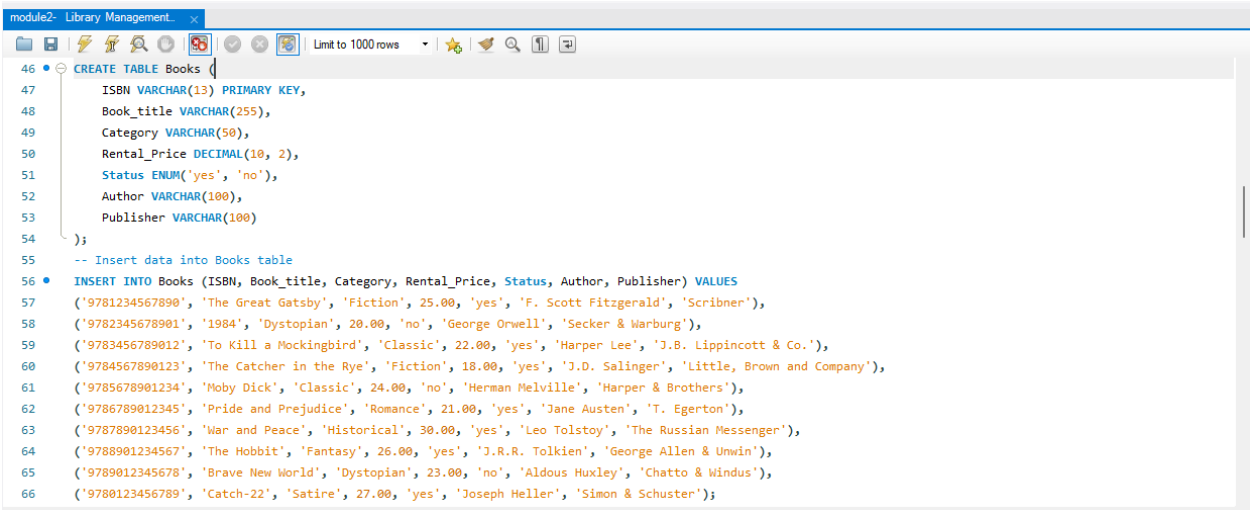
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Emp_Id	Emp_name	Position	Salary	Branch_no
1	Alice Johnson	Manager	60000.00	1
2	Bob Smith	Assistant	45000.00	1
3	Charlie Brown	Manager	65000.00	2
4	Daisy Wilson	Assistant	42000.00	2
5	Eva Adams	Manager	70000.00	3
6	Frank Moore	Assistant	46000.00	3
7	Grace Lee	Manager	62000.00	4
8	Henry Clark	Assistant	43000.00	4
9	Ivy Davis	Manager	68000.00	5
10	Jack Walker	Assistant	44000.00	5
NULL	NULL	NULL	NULL	NULL

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



4. Customer

• Customer_Id

- Set as PRIMARY KEY
  • Customer_name
  • Customer_address
  • Reg_date

```
('9789012345678', 'Brave New World', 'Dystopian', 23.00, 'no', 'Aldous Huxley', 'Chatto & Windus'),
('9780123456789', 'Catch-22', 'Satire', 27.00, 'yes', 'Joseph Heller', 'Simon & Schuster');
select * from books;
CREATE TABLE Customer (
    Customer_Id INT PRIMARY KEY,
    Customer_name VARCHAR(100),
    Customer_address VARCHAR(255),
    Reg_date DATE
);
-- Insert data into Customer table
INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) VALUES
(1, 'John Doe', '123 Maple St, Springfield', '2021-06-15'),
(2, 'Jane Smith', '456 Oak St, Shelbyville', '2020-12-01'),
(3, 'Alice Brown', '789 Pine St, Capital City', '2019-03-22'),
(4, 'Bob Johnson', '321 Elm St, Springfield', '2022-01-10'),
(5, 'Carol White', '654 Cedar St, Shelbyville', '2021-09-15'),
(6, 'David Green', '987 Birch St, Capital City', '2022-07-01'),
(7, 'Eva Black', '135 Willow St, Springfield', '2020-10-23'),
(8, 'Frank Blue', '246 Elm St, Shelbyville', '2021-11-30'),
(9, 'Grace Red', '357 Oak St, Capital City', '2021-05-05'),
(10, 'Hank Purple', '468 Pine St, Springfield', '2022-04-18');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Customer_Id	Customer_name	Customer_address	Reg_date
1	John Doe	123 Maple St, Springfield	2021-06-15
2	Jane Smith	456 Oak St, Shelbyville	2020-12-01
3	Alice Brown	789 Pine St, Capital City	2019-03-22
4	Bob Johnson	321 Elm St, Springfield	2022-01-10
5	Carol White	654 Cedar St, Shelbyville	2021-09-15
6	David Green	987 Birch St, Capital City	2022-07-01
7	Eva Black	135 Willow St, Springfield	2020-10-23
8	Frank Blue	246 Elm St, Shelbyville	2021-11-30
9	Grace Red	357 Oak St, Capital City	2021-05-05
10	Hank Purple	468 Pine St, Springfield	2022-04-18
NULL	NULL	NULL	NULL

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

```
CREATE TABLE IssueStatus (
    Issue_Id INT PRIMARY KEY,
    Issued_cust INT,
    Issue_date DATE,
    Isbn_book VARCHAR(13),
    FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
);
-- Insert data into IssueStatus table
INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issue_date, Isbn_book) VALUES
(1, 1, '2023-06-15', '9781234567890'),
(2, 2, '2023-07-01', '9782345678901'),
(3, 3, '2023-05-20', '9783456789012'),
(4, 4, '2023-08-10', '9784567890123'),
(5, 5, '2023-06-25', '9785678901234'),
(6, 6, '2023-07-30', '9786789012345'),
(7, 7, '2023-06-10', '9787890123456'),
(8, 8, '2023-05-15', '9788901234567'),
(9, 9, '2023-07-22', '9789012345678'),
(10, 10, '2023-06-05', '9780123456789');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Issue_Id	Issued_cust	Issue_date	Isbn_book
1	1	2023-06-15	9781234567890
2	2	2023-07-01	9782345678901
3	3	2023-05-20	9783456789012
4	4	2023-08-10	9784567890123
5	5	2023-06-25	9785678901234
6	6	2023-07-30	9786789012345
7	7	2023-06-10	9787890123456
8	8	2023-05-15	9788901234567
9	9	2023-07-22	9789012345678
10	10	2023-06-05	9780123456789
NULL	NULL	NULL	NULL

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

```
CREATE TABLE ReturnStatus (
    Return_Id INT PRIMARY KEY,
    Return_cust INT,
    Return_book_name VARCHAR(255),
    Return_date DATE,
    Isbn_book2 VARCHAR(13),
    FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
);
-- Insert data into ReturnStatus table
INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2) VALUES
(1, 1, 'The Great Gatsby', '2023-07-01', '9781234567890'),
(2, 2, '1984', '2023-07-20', '9782345678901'),
(3, 3, 'To Kill a Mockingbird', '2023-06-30', '9783456789012'),
(4, 4, 'The Catcher in the Rye', '2023-08-15', '9784567890123'),
(5, 5, 'Moby Dick', '2023-07-10', '9785678901234'),
(6, 6, 'Pride and Prejudice', '2023-08-01', '9786789012345'),
(7, 7, 'War and Peace', '2023-06-25', '9787890123456'),
(8, 8, 'The Hobbit', '2023-06-15', '9788901234567'),
(9, 9, 'Brave New World', '2023-07-10', '9789012345678'),
(10, 10, 'Catch-22', '2023-06-12', '9780123456789');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
1	1	The Great Gatsby	2023-07-01	9781234567890
2	2	1984	2023-07-20	9782345678901
3	3	To Kill a Mockingbird	2023-06-30	9783456789012
4	4	The Catcher in the Rye	2023-08-15	9784567890123
5	5	Moby Dick	2023-07-10	9785678901234
6	6	Pride and Prejudice	2023-08-01	9786789012345
7	7	War and Peace	2023-06-25	9787890123456
8	8	The Hobbit	2023-06-15	9788901234567
9	9	Brave New World	2023-07-10	9789012345678
10	10	Catch-22	2023-06-12	9780123456789
NULL	NULL	NULL	NULL	NULL

QUESTIONS

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

```
131     -- 1.Retrieve the book title, category, and rental price of all available books.
132 •   SELECT Book_title, Category, Rental_Price FROM Books WHERE Status = 'yes';
```

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

Book_title	Category	Rental_Price
Catch-22	Satire	27.00
The Great Gatsby	Fiction	25.00
To Kill a Mockingbird	Classic	22.00
The Catcher in the Rye	Fiction	18.00
Pride and Prejudice	Romance	21.00
War and Peace	Historical	30.00
The Hobbit	Fantasy	26.00

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

```
133     -- 2.List the employee names and their respective salaries in descending order of salary.
134 •   SELECT Emp_name, Salary FROM Employee ORDER BY Salary DESC;
```

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

Emp_name	Salary
Eva Adams	70000.00
Ivy Davis	68000.00
Charlie Brown	65000.00
Grace Lee	62000.00
Alice Johnson	60000.00
Frank Moore	46000.00
Bob Smith	45000.00
Jack Walker	44000.00
Henry Clark	43000.00
Daisy Wilson	42000.00

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

```
135     -- 3.Retrieve the book titles and the corresponding customers who have issued those books.
136 •   SELECT b.Book_title, c.Customer_name FROM IssueStatus i
137     JOIN Books b ON i.Isbn_book = b.ISBN
138     JOIN Customer c ON i.Issued_cust = c.Customer_Id;
```

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

Book_title	Customer_name
The Great Gatsby	John Doe
1984	Jane Smith
To Kill a Mockingbird	Alice Brown
The Catcher in the Rye	Bob Johnson
Moby Dick	Carol White
Pride and Prejudice	David Green
War and Peace	Eva Black
The Hobbit	Frank Blue
Brave New World	Grace Red
Catch-22	Hank Purple

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

```
139     -- 4.Display the total count of books in each category.
140 •   SELECT Category, COUNT(*) AS Total_Count FROM Books GROUP BY Category;
```

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

Category	Total_Count
Satire	1
Fiction	2
Dystopian	2
Classic	2
Romance	1
Historical	1
Fantasy	1

5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.

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

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

Emp_name	Position
Alice Johnson	Manager
Charlie Brown	Manager
Eva Adams	Manager
Grace Lee	Manager
Ivy Davis	Manager

6. List the customer names who registered before 2022-01-01 and have not issued any books yet.

```
144 •   SELECT c.Customer_name FROM Customer c LEFT JOIN IssueStatus i ON c.Customer_Id = i.Issued_cust
145     WHERE c.Reg_date < '2022-01-01' AND i.Issued_cust IS NULL;
146     -- 7.Display the branch numbers and the total count of employees in each branch.
```

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

Customer_name

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

```
146     -- 7.Display the branch numbers and the total count of employees in each branch.
147 •   SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no;
148     -- 8.Display the names of customers who have issued books in the month of June 2023.
```

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

Branch_no	Total_Employees
1	2
2	2
3	2
4	2
5	2

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

```
148     -- 8.Display the names of customers who have issued books in the month of June 2023.
149 •   SELECT DISTINCT c.Customer_name FROM IssueStatus i JOIN Customer c ON i.Issued_cust = c.Customer_Id
150     WHERE i.Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
151     -- 9.Retrieve book_title from the book table containing history.
```

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

Customer_name
John Doe
Carol White
Eva Black
Hank Purple

9. Retrieve book_title from book table containing history.

module2- Library Management...

Limit to 1000 rows

```
151     -- 9.Retrieve book_title from the book table containing history.
152 •   SELECT DISTINCT b.Book_title FROM Books b JOIN IssueStatus i ON b.ISBN = i.Isbn_book
153     UNION
154     SELECT DISTINCT b.Book_title
155     FROM Books b
156     JOIN ReturnStatus r ON b.ISBN = r.Isbn_book2;
157     -- 10.Retrieve the branch numbers along with the count of employees for branches having more than 5 employees.
```

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

Book_title
Catch-22
The Great Gatsby
1984
To Kill a Mockingbird
The Catcher in the Rye
Moby Dick
Pride and Prejudice
War and Peace
The Hobbit
Brave New World

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

```
157     -- 10.Retrieve the branch numbers along with the count of employees for branches having more than 5 employees.
158 •   SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no
159     HAVING COUNT(*) > 5;
```

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

Branch_no	Total_Employees

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

```
160     -- 11.Retrieve the names of employees who manage branches and their respective branch addresses.
161 •   SELECT e.Emp_name, b.Branch_address FROM Employee e JOIN Branch b ON e.Branch_no = b.Branch_no
162     WHERE e.Position = 'Manager';
```

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

Emp_name	Branch_address
Alice Johnson	123 Elm St, Springfield
Charlie Brown	456 Oak St, Shelbyville
Eva Adams	789 Pine St, Capital City
Grace Lee	321 Maple St, Springfield
Ivy Davis	654 Cedar St, Shelbyville

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

```
164 •   SELECT DISTINCT c.Customer_name FROM IssueStatus i JOIN Books b ON i.Isbn_book = b.ISBN
165     JOIN Customer c ON i.Issued_cust = c.Customer_Id
166     WHERE b.Rental_Price > 25;
```

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

Customer_name
Hank Purple
Eva Black
Frank Blue