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

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

1.

```
1  CREATE DATABASE library;
2  USE library;
3  CREATE TABLE Branch (
4  Branch_no INT PRIMARY KEY,
5  Manager_Id INT,
6  Branch_address VARCHAR(255),
7  Contact_no VARCHAR(15)
8  );
9  desc branch;
10
11
```

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

Field	Type	Null	Key	Default	Extra
Branch_no	int	NO	PRI	NULL	
Manager_Id	int	YES		NULL	
Branch_address	varchar(255)	YES		NULL	
Contact_no	varchar(15)	YES		NULL	

2.

```
10 CREATE TABLE Employee (
11 Emp_Id INT PRIMARY KEY,
12 Emp_name VARCHAR(255),
13 Position VARCHAR(255),
14 Salary DECIMAL(10, 2),
15 Branch_no INT,
16 FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
17 );
18 desc employee;
19
20
21
```

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

Field	Type	Null	Key	Default	Extra
Emp_Id	int	NO	PRI	NULL	
Emp_name	varchar(255)	YES		NULL	
Position	varchar(255)	YES		NULL	
Salary	decimal(10,2)	YES		NULL	
Branch_no	int	YES	MUL	NULL	

.

3.

```
CREATE TABLE Books (
ISBN VARCHAR(13) PRIMARY KEY,
Book_title VARCHAR(255),
Category VARCHAR(100),
Rental_Price DECIMAL(10, 2),
Status VARCHAR(3) CHECK (Status IN ('yes', 'no')),
Author VARCHAR(255),
Publisher VARCHAR(255)
);
desc books;
```

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

Field	Type	Null	Key	Default	Extra
ISBN	varchar(13)	NO	PRI	NULL	
Book_title	varchar(255)	YES		NULL	
Category	varchar(100)	YES		NULL	
Rental_Price	decimal(10,2)	YES		NULL	
Status	varchar(3)	YES		NULL	
Author	varchar(255)	YES		NULL	
Publisher	varchar(255)	YES		NULL	

4.

```
CREATE TABLE Customer (
Customer_Id INT PRIMARY KEY,
Customer_name VARCHAR(255),
Customer_address VARCHAR(255),
Reg_date DATE
);
desc customer;
```

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

Field	Type	Null	Key	Default	Extra
Customer_Id	int	NO	PRI	NULL	
Customer_name	varchar(255)	YES		NULL	
Customer_address	varchar(255)	YES		NULL	
Reg_date	date	YES		NULL	

5.

```
36  CREATE TABLE IssueStatus (
37  Issue_Id INT PRIMARY KEY,
38  Issued_cust INT,
39  Issued_book_name VARCHAR(255),
40  Issue_date DATE,
41  Isbn_book VARCHAR(13),
42  FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
43  FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
44  );
45  desc IssueStatus;
46
```

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

Field	Type	Null	Key	Default	Extra
Issue_Id	int	NO	PRI	NULL	
Issued_cust	int	YES	MUL	NULL	
Issued_book_name	varchar(255)	YES		NULL	
Issue_date	date	YES		NULL	
Isbn_book	varchar(13)	YES	MUL	NULL	

6.

```
46  CREATE TABLE ReturnStatus (
47  Return_Id INT PRIMARY KEY,
48  Return_cust INT,
49  Return_book_name VARCHAR(255),
50  Return_date DATE,
51  Isbn_book2 VARCHAR(13),
52  FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
53  FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
54  );
55  desc ReturnStatus;
56
57
```

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

Field	Type	Null	Key	Default	Extra
Return_Id	int	NO	PRI	NULL	
Return_cust	int	YES	MUL	NULL	
Return_book_name	varchar(255)	YES		NULL	
Return_date	date	YES		NULL	
Isbn_book2	varchar(13)	YES	MUL	NULL	

Show Tables

```
56 • show tables;
```

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

Tables_in_library
books
branch
customer
employee
issuestatus
returnstatus

Insert Data

```
17 • INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES
18   (1, 101, '123 Main St, Springfield', '555-1234'),
19   (2, 102, '456 Oak St, Shelbyville', '555-5678'),
20   (3, 103, '789 Pine St, Capital City', '555-9101'),
21   (4, 104, '321 Cedar St, Springfield', '555-1112'),
22   (5, 105, '654 Maple St, Shelbyville', '555-1314');
23 • select*from  branch;
```

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

Branch_no	Manager_Id	Branch_address	Contact_no
1	101	123 Main St, Springfield	555-1234
2	102	456 Oak St, Shelbyville	555-5678
3	103	789 Pine St, Capital City	555-9101
4	104	321 Cedar St, Springfield	555-1112
5	105	654 Maple St, Shelbyville	555-1314
NULL	NULL	NULL	NULL

```
31   (202, 'Eve Black', 'Assistant', 42000, 2),
32   (203, 'Frank Green', 'Assistant', 48000, 3),
33   (204, 'George Brown', 'Assistant', 46000, 4),
34   (205, 'Helen Smith', 'Assistant', 47000, 5),
35   (301, 'Grace Blue', 'Clerk', 35000, 1),
36   (302, 'Hannah Red', 'Clerk', 36000, 2),
37   (303, 'Ivy Yellow', 'Clerk', 37000, 3),
38   (304, 'Jack Black', 'Clerk', 38000, 4),
39   (305, 'Kelly White', 'Clerk', 39000, 5);
40 • select*from employee;
```

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

Emp_Id	Emp_name	Position	Salary	Branch_no
101	Alice Johnson	Manager	75000.00	1
102	Bob Smith	Manager	78000.00	2
103	Charlie Brown	Manager	76000.00	3
104	Diana Prince	Manager	79000.00	4
105	Edward Johnson	Manager	80000.00	5
201	David White	Assistant	45000.00	1
202	Eve Black	Assistant	42000.00	2
203	Frank Green	Assistant	48000.00	3
204	George Brown	Assistant	46000.00	4
205	Helen Smith	Assistant	47000.00	5
301	Grace Blue	Clerk	35000.00	1

```sql
INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES
('9783161484100', 'The Great Gatsby', 'Fiction', 30.00, 'yes', 'F. Scott Fitzgerald', 'Scribner'),
('9781402894626', 'To Kill a Mockingbird', 'Fiction', 25.00, 'no', 'Harper Lee', 'J.B. Lippincott & Co.'),
('9780743273565', 'The Da Vinci Code', 'Thriller', 28.00, 'yes', 'Dan Brown', 'Doubleday'),
('9780394523204', 'The History of Ancient Rome', 'History', 35.00, 'no', 'Livy', 'Penguin Books'),
('9780743273572', 'Digital Fortress', 'Thriller', 22.00, 'yes', 'Dan Brown', 'St. Martin\'s Press'),
('9780670813028', 'Sapiens: A Brief History of Humankind', 'History', 40.00, 'yes', 'Yuval Noah Harari', 'Harper'),
('9780743273589', 'Angels and Demons', 'Thriller', 24.00, 'yes', 'Dan Brown', 'Pocket Books');
select*from books;
```

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

ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
9780394523204	The History of Ancient Rome	History	35.00	no	Livy	Penguin Books
9780670813028	Sapiens: A Brief History of Humankind	History	40.00	yes	Yuval Noah Harari	Harper
9780743273565	The Da Vinci Code	Thriller	28.00	yes	Dan Brown	Doubleday
9780743273572	Digital Fortress	Thriller	22.00	yes	Dan Brown	St. Martin's Press
9780743273589	Angels and Demons	Thriller	24.00	yes	Dan Brown	Pocket Books
9781402894626	To Kill a Mockingbird	Fiction	25.00	no	Harper Lee	J.B. Lippincott & Co.
9783161484100	The Great Gatsby	Fiction	30.00	yes	F. Scott Fitzgerald	Scribner
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) VALUES
(1, 'John Doe', '321 Maple St, Springfield', '2021-12-15'),
(2, 'Jane Smith', '654 Elm St, Shelbyville', '2020-05-10'),
(3, 'Emily Davis', '987 Birch St, Capital City', '2022-02-20'),
(4, 'Michael Brown', '147 Oak St, Springfield', '2023-06-05'),
(5, 'Sarah Wilson', '258 Pine St, Shelbyville', '2021-01-25'),
(6, 'Robert Johnson', '369 Cedar St, Springfield', '2021-03-30'),
(7, 'Laura Lee', '741 Cherry St, Capital City', '2020-11-11');
select*from Customer;
```

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

Customer_Id	Customer_name	Customer_address	Reg_date
1	John Doe	321 Maple St, Springfield	2021-12-15
2	Jane Smith	654 Elm St, Shelbyville	2020-05-10
3	Emily Davis	987 Birch St, Capital City	2022-02-20
4	Michael Brown	147 Oak St, Springfield	2023-06-05
5	Sarah Wilson	258 Pine St, Shelbyville	2021-01-25
6	Robert Johnson	369 Cedar St, Springfield	2021-03-30
7	Laura Lee	741 Cherry St, Capital City	2020-11-11
NULL	NULL	NULL	NULL

```sql
INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book) VALUES
(1, 1, 'The Great Gatsby', '2023-05-12', '9783161484100'),
(2, 3, 'The Da Vinci Code', '2023-06-15', '9780743273565'),
(3, 4, 'Digital Fortress', '2023-06-20', '9780743273572'),
(4, 5, 'Sapiens: A Brief History of Humankind', '2023-07-01', '9780670813028'),
(5, 6, 'Angels and Demons', '2023-06-25', '9780743273589');
select* from IssueStatus ;
```

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

Issue_Id	Issued_cust	Issued_book_name	Issue_date	Isbn_book
1	1	The Great Gatsby	2023-05-12	9783161484100
2	3	The Da Vinci Code	2023-06-15	9780743273565
3	4	Digital Fortress	2023-06-20	9780743273572
4	5	Sapiens: A Brief History of Humankind	2023-07-01	9780670813028
5	6	Angels and Demons	2023-06-25	9780743273589
NULL	NULL	NULL	NULL	NULL

```sql
INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2) VALUES
(1, 1, 'To Kill a Mockingbird', '2023-06-10', '9781402894626'),
(2, 2, 'The Great Gatsby', '2023-07-01', '9783161484100'),
(3, 3, 'The Da Vinci Code', '2023-07-10', '9780743273565');
select*from ReturnStatus;
```

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

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
1	1	To Kill a Mockingbird	2023-06-10	9781402894626
2	2	The Great Gatsby	2023-07-01	9783161484100
3	3	The Da Vinci Code	2023-07-10	9780743273565
NULL	NULL	NULL	NULL	NULL

Queries

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

```sql
SELECT Book_title, Category, Rental_Price
FROM Books
WHERE Status = 'yes';
```

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

Book_title	Category	Rental_Price
Sapiens: A Brief History of Humankind	History	40.00
The Da Vinci Code	Thriller	28.00
Digital Fortress	Thriller	22.00
Angels and Demons	Thriller	24.00
The Great Gatsby	Fiction	30.00

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

```sql
SELECT Emp_name, Salary
FROM Employee
ORDER BY Salary DESC;
```

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

Emp_name	Salary
Edward Johnson	80000.00
Diana Prince	79000.00
Bob Smith	78000.00
Charlie Brown	76000.00
Alice Johnson	75000.00
Frank Green	48000.00
Helen Smith	47000.00
George Brown	46000.00
David White	45000.00
Eve Black	42000.00
Kelly White	39000.00

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

.

```
118 •   SELECT Books.Book_title, Customer.Customer_name
119     FROM IssueStatus
120     JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
121     JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
```

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

Book_title	Customer_name
The Great Gatsby	John Doe
The Da Vinci Code	Emily Davis
Digital Fortress	Michael Brown
Sapiens: A Brief History of Humankind	Sarah Wilson
Angels and Demons	Robert Johnson

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

```
122
123 •   SELECT Category, COUNT(*) AS Total_Books
124     FROM Books
125     GROUP BY Category;
```

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

Category	Total_Books
History	2
Thriller	3
Fiction	2

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

```
127 •  SELECT Emp_name, Position
128    FROM Employee
129    WHERE Salary > 50000;
```

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

Emp_name	Position
Alice Johnson	Manager
Bob Smith	Manager
Charlie Brown	Manager
Diana Prince	Manager
Edward Johnson	Manager

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

```
130
131 •  SELECT Customer.Customer_name
132    FROM Customer
133    LEFT JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
134    WHERE Customer.Reg_date < '2022-01-01' AND IssueStatus.Issue_Id IS NULL;
```

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

Customer_name
Jane Smith
Laura Lee

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

```
135
136 •  SELECT Branch_no, COUNT(*) AS Total_Employees
137    FROM Employee
138    GROUP BY Branch_no;
```

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

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

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

```
139
140 •   SELECT DISTINCT Customer.Customer_name
141     FROM IssueStatus
142     JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id
143     WHERE Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
```

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

Customer_name
Emily Davis
Michael Brown
Robert Johnson

9. Retrieve book_title from book table containing history.

```
144
145 •   SELECT Book_title
146     FROM Books
147     WHERE Book_title LIKE '%history%';
```

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

Book_title
The History of Ancient Rome
Sapiens: A Brief History of Humankind

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

```
148
149 •   SELECT Branch_no, COUNT(*) AS Total_Employees
150     FROM Employee
151     GROUP BY Branch_no
152     HAVING COUNT(*) > 5;
153
```

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

```
1  SELECT Employee.Emp_name, Branch.Branch_address
2  FROM Branch
3  JOIN Employee ON Branch.Manager_Id = Employee.Emp_Id;
```

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

Emp_name	Branch_address
Alice Johnson	123 Main St, Springfield
Bob Smith	456 Oak St, Shelbyville
Charlie Brown	789 Pine St, Capital City
Diana Prince	321 Cedar St, Springfield
Edward Johnson	654 Maple St, Shelbyville

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

```
153
154  SELECT DISTINCT Customer.Customer_name
155  FROM IssueStatus
156  JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
157  JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id
158  WHERE Books.Rental_Price > 25;
```

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

Customer_name
Sarah Wilson
Emily Davis
John Doe