

# Topic : Library Management System

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

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

## 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

## 5. IssueStatus

- Issue\_Id - Set as PRIMARY KEY
- Issued\_cust\_id – 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

## 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

```
34 • INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES
35     (1, 101, '123 Elm Street, Springfield', 5550101),
36     (2, 102, '456 Oak Avenue, Shelbyville', 5550202),
37     (3, 103, '789 Pine Road, Capital City', 5550303),
38     (4, 104, '101 Maple Lane, Ogdenville', 5550404),
39     (5, 105, '202 Birch Boulevard, North Haverbrook', 5550505),
40     (6, 106, '303 Cedar Street, Brockway', 5550606),
41     (7, 107, '404 Walnut Drive, East Ogdenville', 5550707),
42     (8, 108, '505 Aspen Road, South Brockway', 5550808);
43
44 • INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES
45     (201, 'Alice Johnson', 'Manager', 50000, 1),
46     (202, 'Bob Smith', 'Assistant Manager', 40000, 2),
47     (203, 'Charlie Davis', 'Clerk', 30000, 3),
48     (204, 'Diana Evans', 'Librarian', 45000, 4),
49     (205, 'Edward White', 'Assistant Librarian', 35000, 5),
50     (206, 'Fiona Green', 'Clerk', 28000, 6),
51     (207, 'George Brown', 'Librarian', 47000, 7),
52     (208, 'Hannah Adams', 'Manager', 52000, 8),
53     (209, 'Alice Jude', 'Clerk', 25000, 7),
54     (210, 'Osama Kiran', 'Manager', 50000, 7),
55     (211, 'Kennath Martin', 'Assistant', 30000, 7),
```

```

46  (202, 'Bob Smith', 'Assistant Manager', 40000, 2),
47  (203, 'Charlie Davis', 'Clerk', 30000, 3),
48  (204, 'Diana Evans', 'Librarian', 45000, 4),
49  (205, 'Edward White', 'Assistant Librarian', 35000, 5),
50  (206, 'Fiona Green', 'Clerk', 28000, 6),
51  (207, 'George Brown', 'Librarian', 47000, 7),
52  (208, 'Hannah Adams', 'Manager', 52000, 8),
53  (209, 'Alice Jude', 'Clerk', 25000, 7),
54  (210, 'Osama Kiran', 'Manager', 500000, 7),
55  (211, 'Kennath Martin', 'Assistant', 30000, 7),
56  (212, 'Ahmmed Abdul', 'Librarian', 40000, 7),
57  (213, 'Leona Lucid', 'Clerk', 27000, 7);
58
59  • INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES
60  (978-3-16-148410-0, 'The Great Gatsby', 'Fiction', 30.00, 'yes', 'F. Scott Fitzgerald', 'Scribner'),
61  (978-0-14-118280-3, '1984', 'Dystopian', 12.50, 'no', 'George Orwell', 'Penguin'),
62  (978-0-06-112008-4, 'To Kill a Mockingbird', 'Fiction', 40.00, 'yes', 'Harper Lee', 'Harper Perennial'),
63  (978-0-452-28423-4, 'Pride and Prejudice', 'Romance', 10.00, 'no', 'Jane Austen', 'Penguin'),
64  (978-0-7432-7356-5, 'The Catcher in the Rye', 'Fiction', 35.00, 'yes', 'J.D. Salinger', 'Little, Brown and
65  (978-1-5011-4687-8, 'The Road', 'Post-apocalyptic', 13.00, 'no', 'Cormac McCarthy', 'Knopf'),
66  (978-0-374-53012-5, 'The history villa', 'Historical Fiction', 12.00, 'yes', 'Toni Morrison', 'Knopf'),
67  (978-0-14-017739-8, 'Of Mice and Men', 'Tragedy', 9.00, 'yes', 'John Steinbeck', 'Penguin');

```



```

61 (978-0-14-118280-3, '1984', 'Dystopian', 12.50, 'no', 'George Orwell', 'Penguin'),
62 (978-0-06-112008-4, 'To Kill a Mockingbird', 'Fiction', 40.00, 'yes', 'Harper Lee', 'Harper Perennial'),
63 (978-0-452-28423-4, 'Pride and Prejudice', 'Romance', 10.00, 'no', 'Jane Austen', 'Penguin'),
64 (978-0-7432-7356-5, 'The Catcher in the Rye', 'Fiction', 35.00, 'yes', 'J.D. Salinger', 'Little, Brown and'),
65 (978-1-5011-4687-8, 'The Road', 'Post-apocalyptic', 13.00, 'no', 'Cormac McCarthy', 'Knopf'),
66 (978-0-374-53012-5, 'The history villa', 'Historical Fiction', 12.00, 'yes', 'Toni Morrison', 'Knopf'),
67 (978-0-14-017739-8, 'Of Mice and Men', 'Tragedy', 9.00, 'yes', 'John Steinbeck', 'Penguin');
68
69 • INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) VALUES
70 (301, 'John Doe', '111 Apple Street, Springfield', '2023-01-10'),
71 (302, 'Jane Smith', '222 Orange Avenue, Shelbyville', '2023-02-15'),
72 (303, 'Robert Brown', '333 Banana Road, Capital City', '2023-03-20'),
73 (304, 'Lucy Green', '444 Grape Lane, Ogdenville', '2023-04-25'),
74 (305, 'Michael Black', '555 Lemon Drive, North Haverbrook', '2023-05-30'),
75 (306, 'Emma White', '666 Peach Boulevard, Brockway', '2023-06-05'),
76 (307, 'Thomas Adams', '777 Plum Street, East Ogdenville', '2023-07-15'),
77 (308, 'Olivia Clark', '888 Cherry Road, South Brockway', '2023-08-20'),
78 (309, 'Evan Pavlov', '999 foresthunt Road,North','2021-09-29'),
79 (310, 'Urie BroneFenbernner', '1000 Kalki Road,South','2021-12-01'),
80 (311, 'Kim Lal', 'abcd Line,South','2023-06-05'),
81 (312, 'Charles Maslow', 'Olivia Street,South','2023-06-10');
82

```

82

83 • **INSERT INTO** IssueStatus (Issue\_Id, Issued\_cust, Issued\_book\_name, Issue\_date, Isbn\_book) **VALUES**

84 (401, 301, 'The Great Gatsby', '2023-09-01', 978-3-16-148410-0),

85 (402, 302, '1984', '2023-09-05', 978-0-14-118280-3),

86 (403, 303, 'To Kill a Mockingbird', '2023-09-10', 978-0-06-112008-4),

87 (404, 304, 'Pride and Prejudice', '2023-09-15', 978-0-452-28423-4),

88 (405, 305, 'The Catcher in the Rye', '2023-09-20', 978-0-7432-7356-5),

89 (406, 306, 'The Road', '2023-09-25', 978-1-5011-4687-8),

90 (407, 307, 'The history villa', '2023-09-30', 978-0-374-53012-5),

91 (408, 308, 'Of Mice and Men', '2023-10-05', 978-0-14-017739-8);

92

93 • **INSERT INTO** ReturnStatus (Return\_Id, Return\_cust, Return\_book\_name, Return\_date, Isbn\_book2) **VALUES**

94 (501, 301, 'The Great Gatsby', '2023-10-10', 978-3-16-148410-0),

95 (502, 302, '1984', '2023-10-15', 978-0-14-118280-3),

96 (503, 303, 'To Kill a Mockingbird', '2023-10-20', 978-0-06-112008-4),

97 (504, 304, 'Pride and Prejudice', '2023-10-25', 978-0-452-28423-4),

98 (505, 305, 'The Catcher in the Rye', '2023-10-30', 978-0-7432-7356-5),

99 (506, 306, 'The Road', '2023-11-05', 978-1-5011-4687-8),

100 (507, 307, 'The history villa', '2023-11-10', 978-0-374-53012-5),

101 (508, 308, 'Of Mice and Men', '2023-11-15', 978-0-14-017739-8);

102


103




# Branch Table:



171


172 • `SELECT * FROM BRANCH;`

Result Grid

 Filter Rows:

Edit:   

Export/Import:  

Wrap Cell Content: 

	BRANCH_NO	MANAGER_ID	BRANCH_ADDRESS	CONTACT_NO
▶	1	101	123 Elm Street, Springfield	5550101
	2	102	456 Oak Avenue, Shelbyville	5550202
	3	103	789 Pine Road, Capital City	5550303
	4	104	101 Maple Lane, Ogdenville	5550404
	5	105	202 Birch Boulevard, North Haverbrook	5550505
	6	106	303 Cedar Street, Brockway	5550606
	7	107	404 Walnut Drive, East Ogdenville	5550707
	8	108	505 Aspen Road, South Brockway	5550808
•	NULL	NULL	NULL	NULL



# Employee table

1/0

177 • `SELECT * FROM EMPLOYEE;`

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

	Emp_ID	Emp_NAME	POSITION	SALARY	BRANCH_NO
▶	201	Alice Johnson	Manager	50000	1
	202	Bob Smith	Assistant Manager	40000	2
	203	Charlie Davis	Clerk	30000	3
	204	Diana Evans	Librarian	45000	4
	205	Edward White	Assistant Librarian	35000	5
	206	Fiona Green	Clerk	28000	6
	207	George Brown	Librarian	47000	7
	208	Hannah Adams	Manager	52000	8
	209	Alice Jude	Clerk	25000	7
	210	Osama Kiran	Manager	500000	7
	211	Kennath Martin	Assistant	30000	7
	212	Ahmmmed Abdul	Librarian	40000	7
	213	Leona Lucid	Clerk	27000	7
*	NULL	NULL	NULL	NULL	NULL

## Books table

```
182 • SELECT * FROM BOOKS;
```

[illegible]

# Customer table:

187 • `SELECT * FROM CUSTOMER;`



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




	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	REG_DATE
▶	301	John Doe	111 Apple Street, Springfield	2023-01-10
	302	Jane Smith	222 Orange Avenue, Shelbyville	2023-02-15
	303	Robert Brown	333 Banana Road, Capital City	2023-03-20
	304	Lucy Green	444 Grape Lane, Ogdenville	2023-04-25
	305	Michael Black	555 Lemon Drive, North Haverbrook	2023-05-30
	306	Emma White	666 Peach Boulevard, Brockway	2023-06-05
	307	Thomas Adams	777 Plum Street, East Ogdenville	2023-07-15
	308	Olivia Clark	888 Cherry Road, South Brockway	2023-08-20
	309	Evan Pavlov	999 foresthunt Road,North	2021-09-29
	310	Urie BroneFenbernner	1000 Kalki Road,South	2021-12-01
	311	Kim Lal	abcd Line,South	2023-06-05
	312	Charles Maslow	Olivia Street,South	2023-06-10
•	NULL	NULL	NULL	NULL



# Issuestatus Table:


192 • `SELECT * FROM ISSUESTATUS;`

Result Grid



Filter Rows:

Edit:








Export/Import:



Wrap Cell Content:


	ISSUE_ID	ISSUED_CUST	ISSUED_BOOK_NAME	ISSUE_DATE	ISBN_BOOK
▶	401	301	The Great Gatsby	2023-09-01	-147451
	402	302	1984	2023-09-05	-117319
	403	303	To Kill a Mockingbird	2023-09-10	-111040
	404	304	Pride and Prejudice	2023-09-15	-27901
	405	305	The Catcher in the Rye	2023-09-20	-13815
	406	306	The Road	2023-09-25	-8729
	407	307	The history villa	2023-09-30	-52413
	408	308	Of Mice and Men	2023-10-05	-16783
✱	NULL	NULL	NULL	NULL	NULL

# Returnstatus Table

197 • `SELECT * FROM RETURNSTATUS;`

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
Export/Import:  					
Wrap Cell Content: 					
	RETURN_ID	RETURN_CUST	RETURN_BOOK_NAME	RETURN_DATE	ISBN_BOOK2
▶	501	301	The Great Gatsby	2023-10-10	-147451
	502	302	1984	2023-10-15	-117319
	503	303	To Kill a Mockingbird	2023-10-20	-111040
	504	304	Pride and Prejudice	2023-10-25	-27901
	505	305	The Catcher in the Rye	2023-10-30	-13815
	506	306	The Road	2023-11-05	-8729
	507	307	The history villa	2023-11-10	-52413
	508	308	Of Mice and Men	2023-11-15	-16783
•	NULL	NULL	NULL	NULL	NULL



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

Limit to 1000 rows

104

105

106 • `SELECT BOOK_TITLE,CATEGORY,RENTAL_PRICE FROM BOOKS;`

107

108

Result Grid

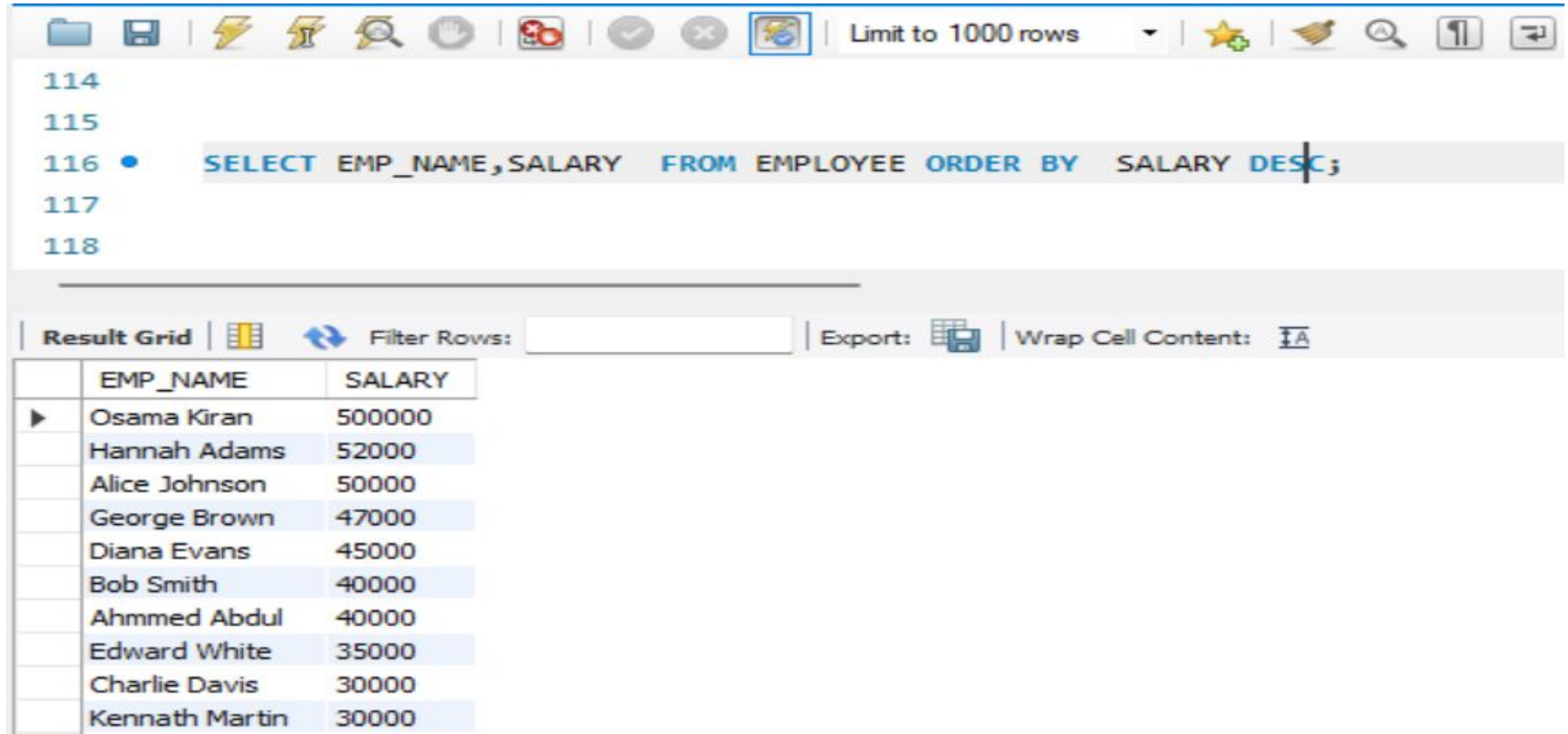
Filter Rows:

Export:

Wrap Cell Content:

	BOOK_TITLE	CATEGORY	RENTAL_PRICE
▶	The Great Gatsby	Fiction	30
	1984	Dystopian	13
	To Kill a Mockingbird	Fiction	40
	The history villa	Historical Fiction	12
	Pride and Prejudice	Romance	10
	Of Mice and Men	Tragedy	9
	The Catcher in the Rye	Fiction	35
	The Road	Post-apocalyptic	13

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



The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, search, and execution. Below the toolbar, a SQL query is entered in a text area:

```
114  
115  
116 • SELECT EMP_NAME, SALARY FROM EMPLOYEE ORDER BY SALARY DESC;  
117  
118
```

Below the query editor, there is a section for the results. It includes a "Result Grid" tab, a "Filter Rows:" input field, and an "Export:" button. The results are displayed in a table with two columns: "EMP\_NAME" and "SALARY". The table contains 10 rows of data, sorted in descending order of salary.

	EMP_NAME	SALARY
▶	Osama Kiran	500000
	Hannah Adams	52000
	Alice Johnson	50000
	George Brown	47000
	Diana Evans	45000
	Bob Smith	40000
	Ahmmed Abdul	40000
	Edward White	35000
	Charlie Davis	30000
	Kennath Martin	30000

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



```
120
121
122 • SELECT ISSUESTATUS.ISSUED_BOOK_NAME,CUSTOMER.CUSTOMER_NAME FROM ISSUESTATUS
123 JOIN CUSTOMER ON ISSUESTATUS.ISSUED_CUST = CUSTOMER.CUSTOMER_ID;
124
```

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

	ISSUED_BOOK_NAME	CUSTOMER_NAME
▶	The Great Gatsby	John Doe
	1984	Jane Smith
	To Kill a Mockingbird	Robert Brown
	Pride and Prejudice	Lucy Green
	The Catcher in the Rye	Michael Black
	The Road	Emma White
	The history villa	Thomas Adams
	Of Mice and Men	Olivia Clark

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

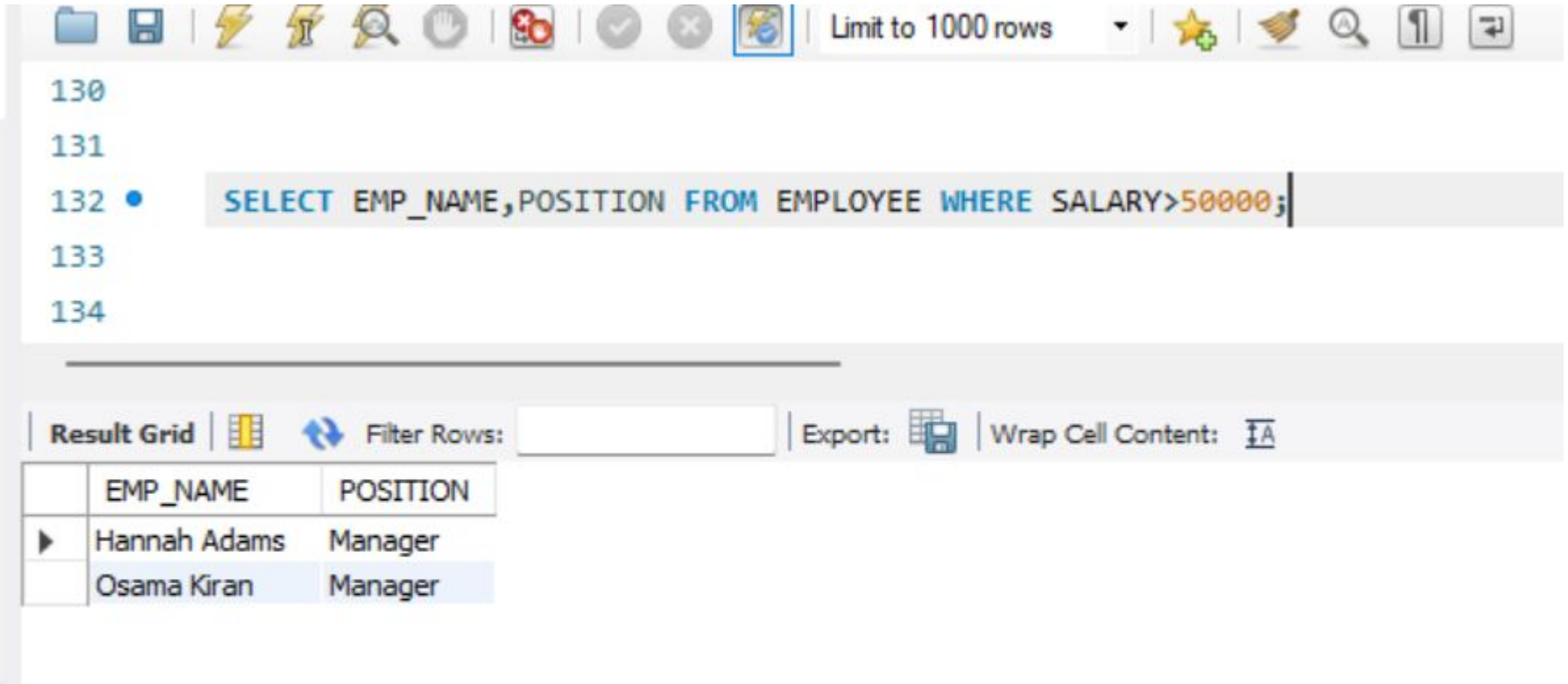
The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and viewing. Below the toolbar, a SQL query is entered in a text area:

```
124  
125  
126 • SELECT COUNT(BOOK_TITLE) AS TOTAL_BOOKS,CATEGORY FROM BOOKS GROUP BY CATEGORY;  
127  
128
```

Below the query editor, there is a section labeled "Result Grid" with a table icon. To the right of the "Result Grid" label is a "Filter Rows:" input field. Further right are "Export:" and "Wrap Cell Content:" options. The table displays the results of the query:

	TOTAL_BOOKS	CATEGORY
▶	3	Fiction
	1	Dystopian
	1	Historical Fiction
	1	Romance
	1	Tragedy
	1	Post-apocalyptic

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



The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and navigation. Below the toolbar, a text area contains a SQL query: `SELECT EMP_NAME, POSITION FROM EMPLOYEE WHERE SALARY > 50000;`. The query is highlighted in blue. Below the query editor, there is a horizontal separator line. Below the separator line, there is a toolbar for the result grid, including options for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar, a table displays the results of the query. The table has two columns: 'EMP\_NAME' and 'POSITION'. The first row shows 'Hannah Adams' as a 'Manager'. The second row shows 'Osama Kiran' as a 'Manager'.





130

131

132 • `SELECT EMP_NAME, POSITION FROM EMPLOYEE WHERE SALARY > 50000;`

133


134

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

	EMP_NAME	POSITION
▶	Hannah Adams	Manager
	Osama Kiran	Manager



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



135

136 • `SELECT CUSTOMER.CUSTOMER_NAME FROM CUSTOMER`

137 `LEFT JOIN ISSUESTATUS ON CUSTOMER.CUSTOMER_ID = ISSUESTATUS.ISSUED_CUST`

138 `WHERE CUSTOMER.REG_DATE < '2022-01-01' AND ISSUESTATUS.ISSUE_ID IS NULL;`

139

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

	CUSTOMER_NAME
▶	Evan Pavlov
	Urie BroneFenberner

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



140

141

142

143 • `SELECT BRANCH_NO ,COUNT(EMP_ID) AS TOTAL_EMPLY FROM EMPLOYEE GROUP BY BRANCH_NO ;`

144

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

	BRANCH_NO	TOTAL_EMPLY
▶	1	1
	2	1
	3	1
	4	1
	5	1
	6	1
	7	6
	8	1

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



```
144
145
146 • SELECT CUSTOMER_NAME ,REG_DATE FROM CUSTOMER WHERE MONTH(REG_DATE)=06 AND YEAR(REG_DATE)=2023;
147
148
```

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

	CUSTOMER_NAME	REG_DATE
▶	Emma White	2023-06-05
	Kim Lal	2023-06-05
	Charles Maslow	2023-06-10

9. Retrieve book\_title from book table containing history.



147

148

149

150

151 • `SELECT BOOK_TITLE FROM BOOKS WHERE BOOK_TITLE LIKE '%history%';`

Result Grid



Filter Rows:

Export:

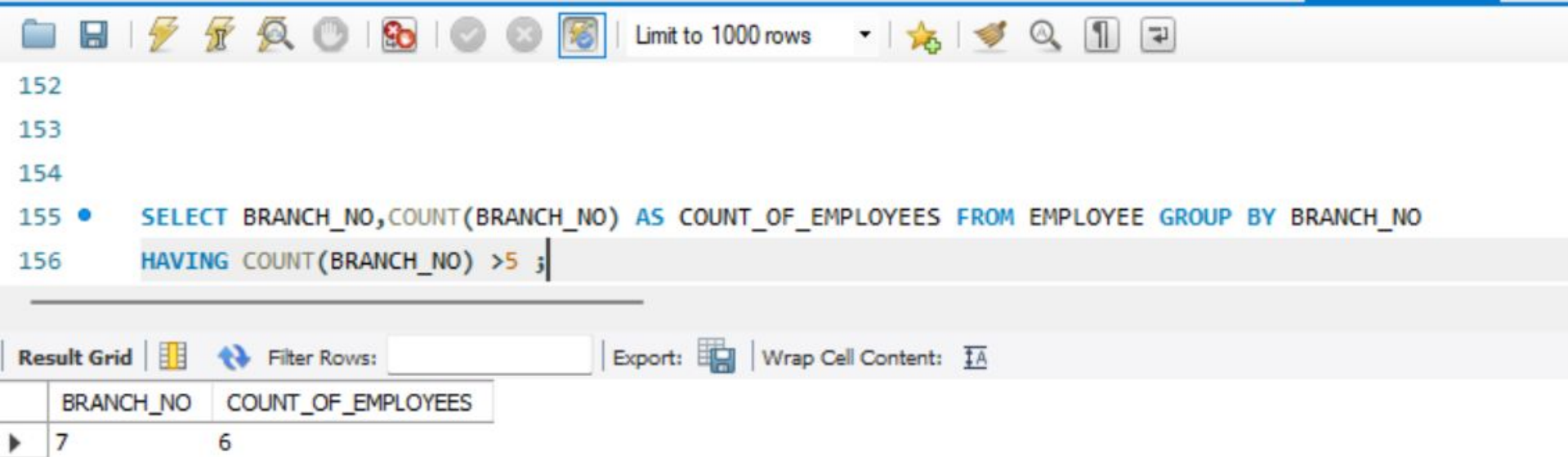


Wrap Cell Content:



	BOOK_TITLE
▶	The history villa

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



The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, search, and execution. Below the toolbar, the SQL query is entered in a text area. The query is as follows:

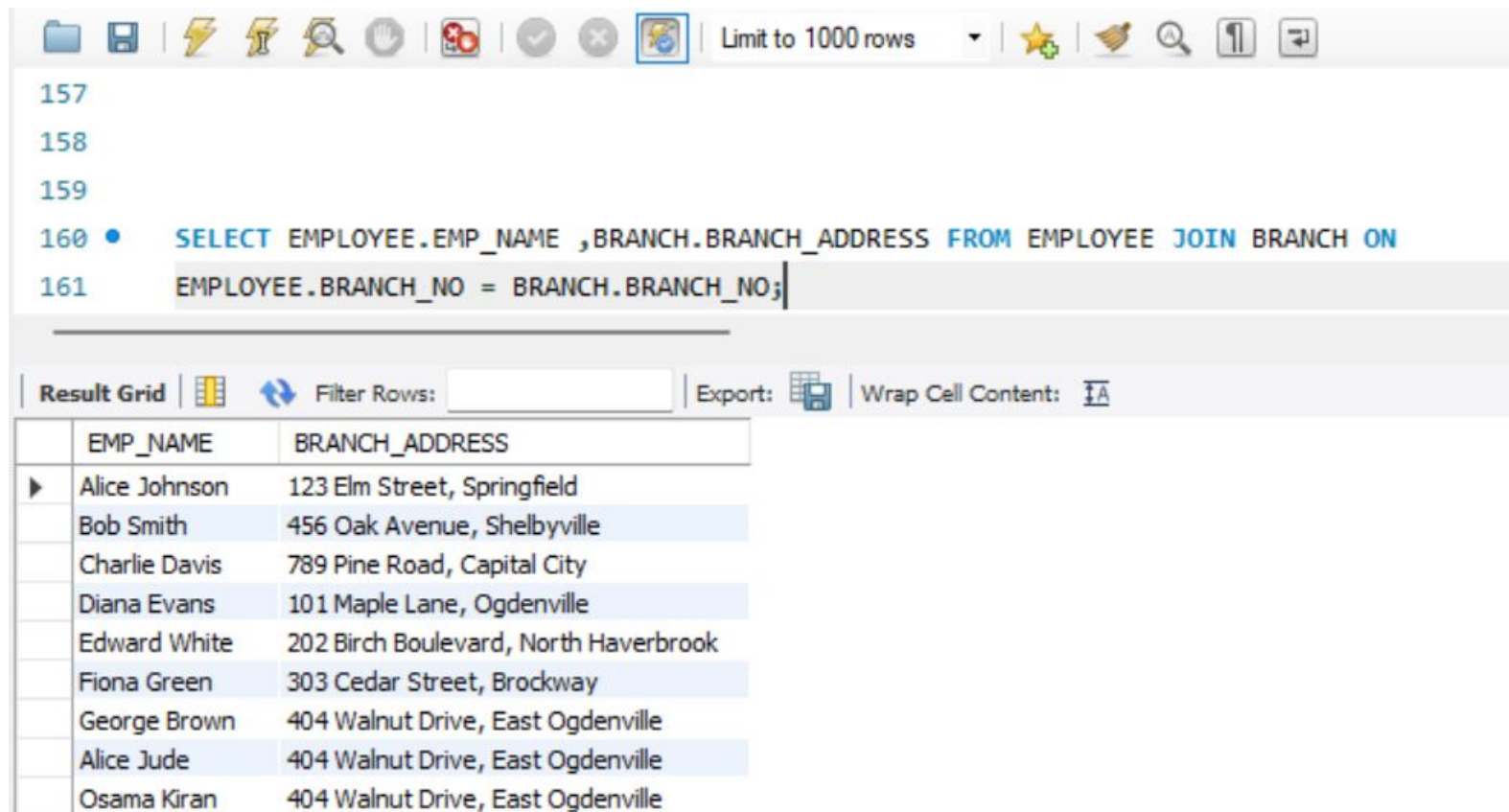
```
152  
153  
154  
155 • SELECT BRANCH_NO,COUNT(BRANCH_NO) AS COUNT_OF_EMPLOYEES FROM EMPLOYEE GROUP BY BRANCH_NO  
156 HAVING COUNT(BRANCH_NO) >5 ;
```

Below the query editor, there is a section for the results. It includes a "Result Grid" tab, a "Filter Rows" input field, and an "Export" button. The results are displayed in a table with two columns: "BRANCH\_NO" and "COUNT\_OF\_EMPLOYEES". The table shows one row with the value 7 for "BRANCH\_NO" and 6 for "COUNT\_OF\_EMPLOYEES".

BRANCH_NO	COUNT_OF_EMPLOYEES
7	6



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



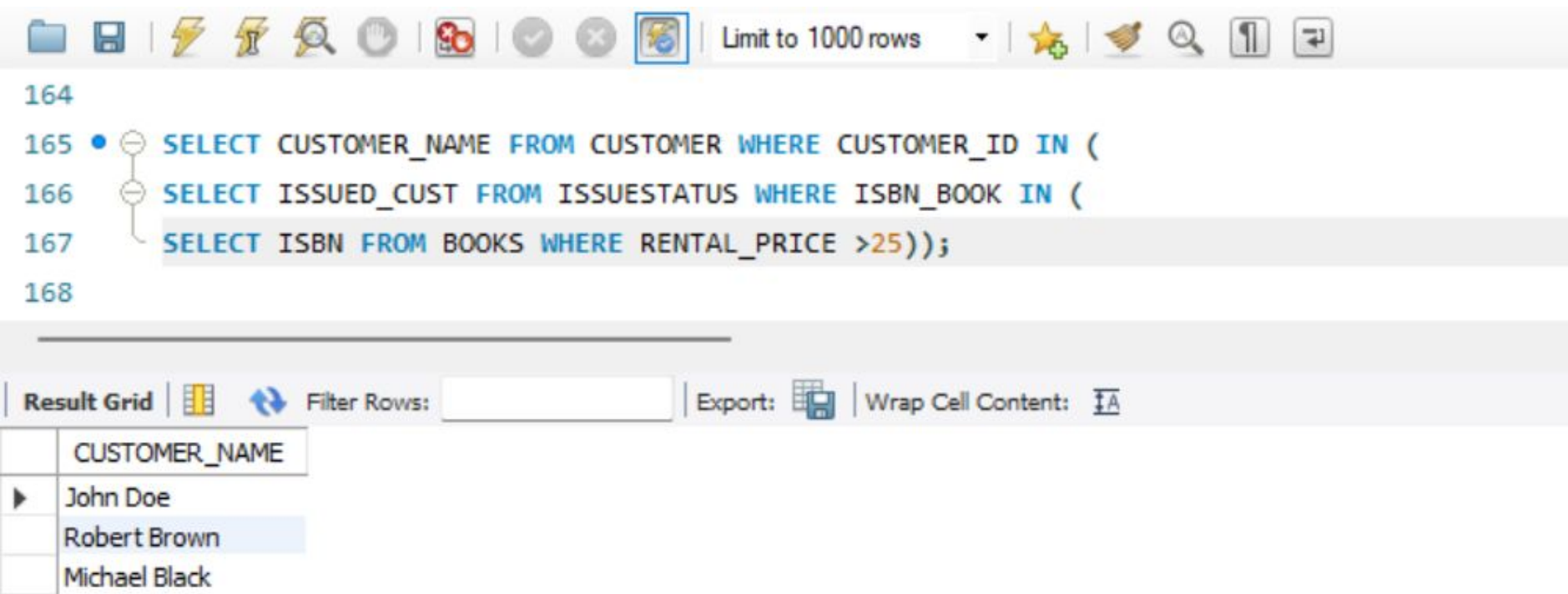
The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, search, and execution. Below the toolbar, the SQL query is displayed in a text area. The query is a SELECT statement that joins the EMPLOYEE and BRANCH tables on the basis of EMPLOYEE.BRANCH\_NO = BRANCH.BRANCH\_NO. The results of the query are shown in a table below the query editor. The table has two columns: EMP\_NAME and BRANCH\_ADDRESS. The results list nine employees and their respective branch addresses.

```
157
158
159
160 • SELECT EMPLOYEE.EMP_NAME ,BRANCH.BRANCH_ADDRESS FROM EMPLOYEE JOIN BRANCH ON
161 EMPLOYEE.BRANCH_NO = BRANCH.BRANCH_NO;
```

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

	EMP_NAME	BRANCH_ADDRESS
▶	Alice Johnson	123 Elm Street, Springfield
	Bob Smith	456 Oak Avenue, Shelbyville
	Charlie Davis	789 Pine Road, Capital City
	Diana Evans	101 Maple Lane, Ogdenville
	Edward White	202 Birch Boulevard, North Haverbrook
	Fiona Green	303 Cedar Street, Brockway
	George Brown	404 Walnut Drive, East Ogdenville
	Alice Jude	404 Walnut Drive, East Ogdenville
	Osama Kiran	404 Walnut Drive, East Ogdenville

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



The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and search. Below the toolbar, a SQL query is entered in a text area. The query is a nested SELECT statement designed to find customer names based on book rental prices. The query is as follows:

```
164  
165 • SELECT CUSTOMER_NAME FROM CUSTOMER WHERE CUSTOMER_ID IN (  
166     SELECT ISSUED_CUST FROM ISSUESTATUS WHERE ISBN_BOOK IN (  
167         SELECT ISBN FROM BOOKS WHERE RENTAL_PRICE >25));  
168
```

Below the query editor, there is a section for the results. It includes a 'Result Grid' tab, a 'Filter Rows' input field, and an 'Export' button. The 'Result Grid' is currently active and displays the following data:

CUSTOMER_NAME
John Doe
Robert Brown
Michael Black

