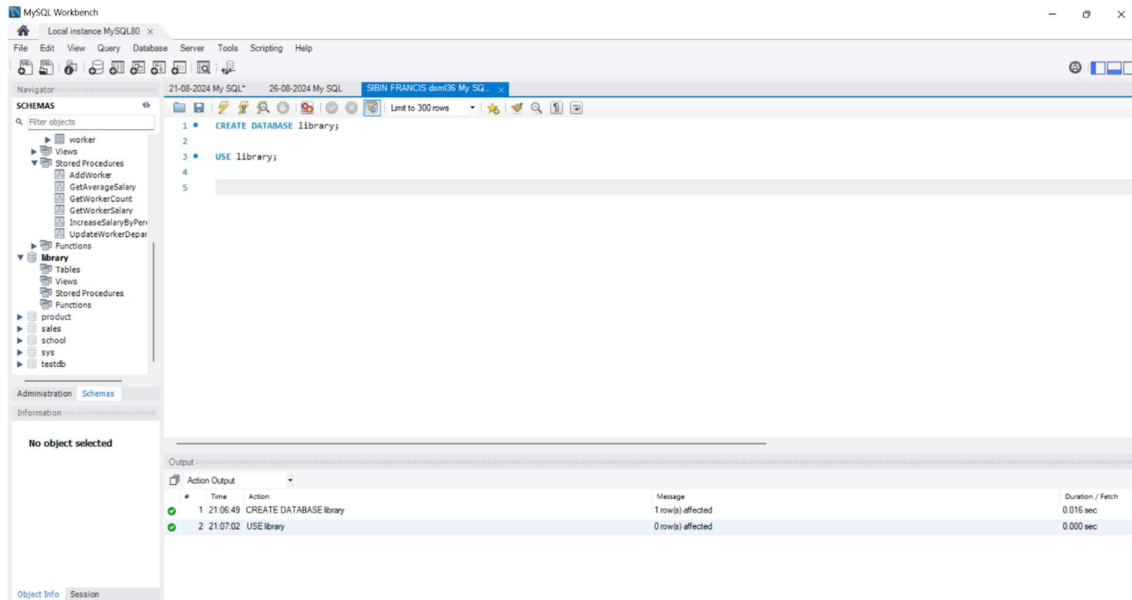
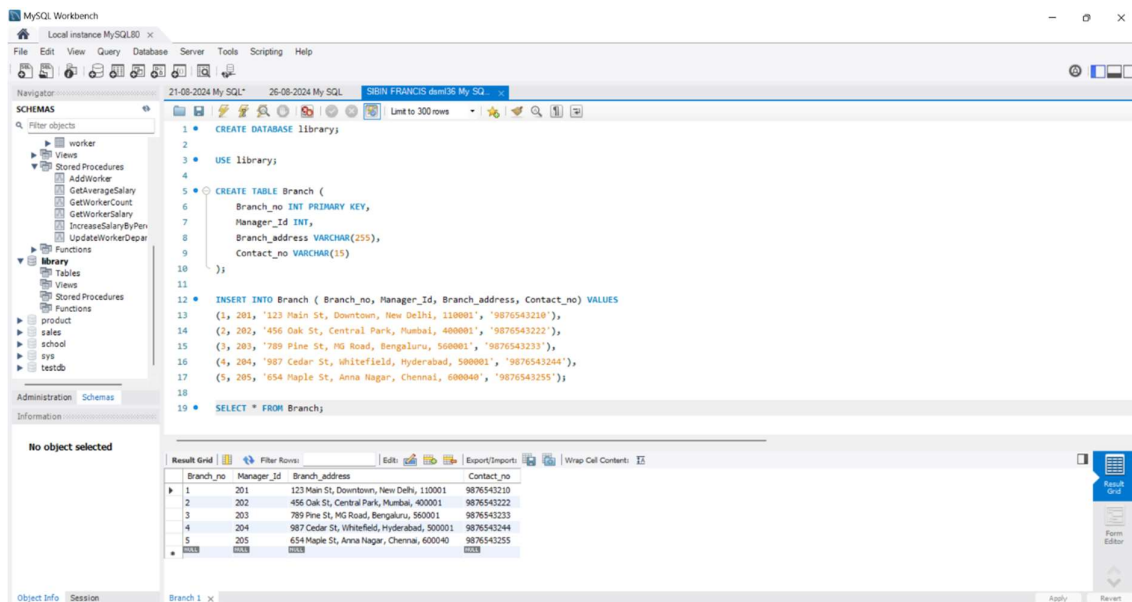


Library Management System

➤ CREADTED DATABASE **library**



➤ CREATED TABLE **Branch** AND INSERTED VALUES



➤ CREATED TABLE **Employee** AND INSERTED VALUES

The screenshot shows the MySQL Workbench interface with the 'SIBIN FRANCIS dem36 My S2' database selected. The 'SCHEMAS' pane on the left shows the 'library' schema. The main editor displays the following SQL code:

```
21 CREATE TABLE Employee (  
22   Emp_Id INT PRIMARY KEY,  
23   Emp_name VARCHAR(100),  
24   Position VARCHAR(100),  
25   Salary DECIMAL(10, 2),  
26   Branch_no INT,  
27   FOREIGN KEY (Branch_no) REFERENCES Branch (Branch_no)  
28 );  
29  
30 INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES  
31 (201, 'Alice Smith', 'Manager', 80000.00, 1),  
32 (202, 'Bob Johnson', 'Assistant Manager', 60000.00, 2),  
33 (203, 'Charlie Davis', 'Manager', 75000.00, 3),  
34 (204, 'Eve Thompson', 'Clerk', 30000.00, 1),  
35 (205, 'Frank White', 'Clerk', 55000.00, 2),  
36 (206, 'Grace Lee', 'Clerk', 45000.00, 3),  
37 (207, 'Henry Walker', 'Clerk', 52000.00, 4),  
38 (208, 'Ivy Adams', 'Manager', 90000.00, 4),  
39 (209, 'Jack Evans', 'Assistant Manager', 70000.00, 5);  
40
```

The 'Result Grid' at the bottom shows the inserted data:

Emp_Id	Emp_name	Position	Salary	Branch_no
201	Alice Smith	Manager	80000.00	1
202	Bob Johnson	Assistant Manager	60000.00	2
203	Charlie Davis	Manager	75000.00	3
204	Eve Thompson	Clerk	30000.00	1
205	Frank White	Clerk	55000.00	2
206	Grace Lee	Clerk	45000.00	3
207	Henry Walker	Clerk	52000.00	4
208	Ivy Adams	Manager	90000.00	4
209	Jack Evans	Assistant Manager	70000.00	5

➤ CREATED TABLE **Books** AND INSERTED VALUES

The screenshot shows the MySQL Workbench interface with the 'SIBIN FRANCIS dem36 My S2' database selected. The 'SCHEMAS' pane on the left shows the 'library' schema. The main editor displays the following SQL code:

```
43 CREATE TABLE Books (  
44   ISBN VARCHAR(13) PRIMARY KEY,  
45   Book_title VARCHAR(255),  
46   Category VARCHAR(100),  
47   Rental_Price DECIMAL(10, 2),  
48   Status VARCHAR(1),  
49   Author VARCHAR(100),  
50   Publisher VARCHAR(100)  
51 );  
52  
53 INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES  
54 ('1234567890', 'The History of Time', 'History', 40.00, 'Yes', 'Stephen Hawking', 'Penguin'),  
55 ('1234567891', 'Science Revolution', 'Science', 50.00, 'Yes', 'Isaac Newton', 'HarperCollins'),  
56 ('1234567892', 'The Art of War', 'History', 30.00, 'No', 'Sun Tzu', 'Random House'),  
57 ('1234567893', 'Quantum Physics', 'Science', 60.00, 'Yes', 'Niels Bohr', 'Springer'),  
58 ('1234567894', 'World War II', 'History', 45.00, 'No', 'John Keegan', 'Oxford Press'),  
59 ('1234567895', 'The History of Ancient Rome', 'History', 35.00, 'Yes', 'Mary Beard', 'Penguin'),  
60 ('1234567896', 'Astrophysics Explained', 'Science', 70.00, 'No', 'Neil deGrasse Tyson', 'Random House');  
61  
62 SELECT * FROM Books ;  
63
```

The 'Result Grid' at the bottom shows the inserted data:

ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
1234567890	The History of Time	History	40.00	Yes	Stephen Hawking	Penguin
1234567891	Science Revolution	Science	50.00	Yes	Isaac Newton	HarperCollins
1234567892	The Art of War	History	30.00	No	Sun Tzu	Random House
1234567893	Quantum Physics	Science	60.00	Yes	Niels Bohr	Springer
1234567894	World War II	History	45.00	No	John Keegan	Oxford Press
1234567895	The History of Ancient Rome	History	35.00	Yes	Mary Beard	Penguin
1234567896	Astrophysics Explained	Science	70.00	No	Neil deGrasse Tyson	Random House

➤ CREATED TABLE **Customer** AND INSERTED VALUES

The screenshot shows the MySQL Workbench interface with the 'Customer' table created and populated. The SQL editor contains the following code:

```
64 CREATE TABLE Customer (  
65     Customer_id INT PRIMARY KEY,  
66     Customer_name VARCHAR(100),  
67     Customer_address VARCHAR(255),  
68     Reg_date DATE  
69 );  
70  
71 INSERT INTO Customer ( Customer_id, Customer_name, Customer_address, Reg_date) VALUES  
72 (301, 'Emily Brown', '12 Willow St, City A', '2021-12-15'),  
73 (302, 'Michael Green', '98 Elm St, City B', '2022-01-20'),  
74 (303, 'Sophia Miller', '56 Maple St, City C', '2020-10-05'),  
75 (304, 'James Harris', '77 Cedar St, City A', '2021-07-22'),  
76 (305, 'Olivia Lewis', '22 Birch St, City B', '2022-06-18');  
77  
78 SELECT * FROM Customer;
```

The Result Grid shows the following data:

Customer_id	Customer_name	Customer_address	Reg_date
301	Emily Brown	12 Willow St, City A	2021-12-15
302	Michael Green	98 Elm St, City B	2022-01-20
303	Sophia Miller	56 Maple St, City C	2020-10-05
304	James Harris	77 Cedar St, City A	2021-07-22
305	Olivia Lewis	22 Birch St, City B	2022-06-18

➤ CREATED TABLE **IssueStatus** AND INSERTED VALUES

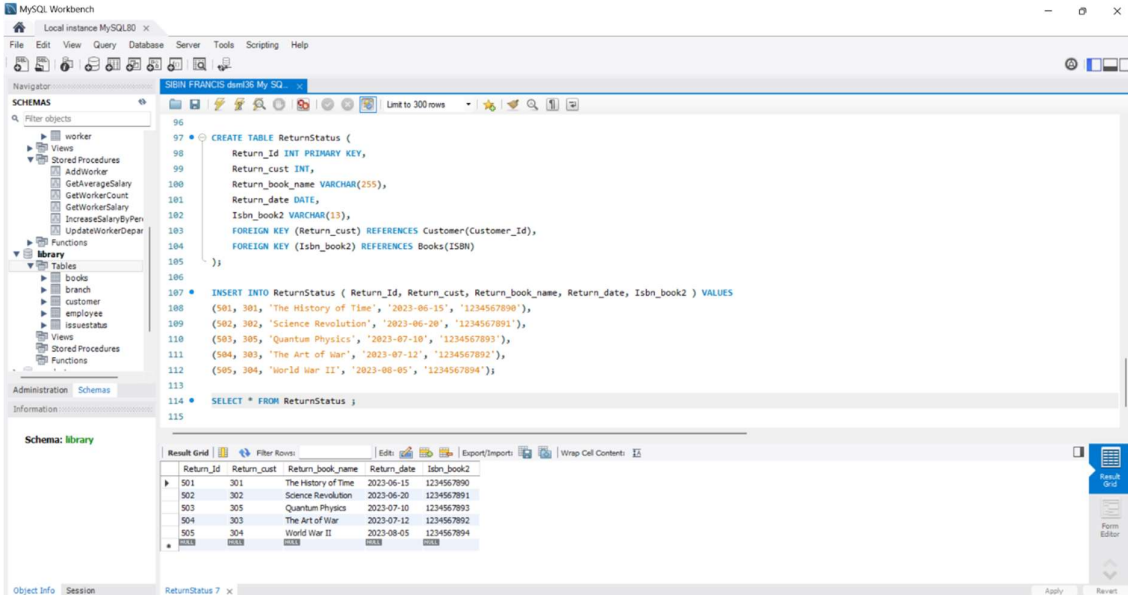
The screenshot shows the MySQL Workbench interface with the 'IssueStatus' table created and populated. The SQL editor contains the following code:

```
79  
80 CREATE TABLE IssueStatus (  
81     Issue_id INT PRIMARY KEY,  
82     Issued_cust INT,  
83     Issued_book_name VARCHAR(255),  
84     Issue_date DATE,  
85     Isbn_book VARCHAR(13),  
86     FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_id),  
87     FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)  
88 );  
89  
90 INSERT INTO IssueStatus (Issue_id, Issued_cust, Issued_book_name, Issue_date, Isbn_book) VALUES  
91 (401, 301, 'The History of Time', '2023-06-10', '1234567890'),  
92 (402, 302, 'Science Revolution', '2023-06-15', '1234567891'),  
93 (403, 305, 'Quantum Physics', '2023-07-05', '1234567893');  
94  
95 SELECT * FROM IssueStatus;
```

The Result Grid shows the following data:

Issue_id	Issued_cust	Issued_book_name	Issue_date	Isbn_book
401	301	The History of Time	2023-06-10	1234567890
402	302	Science Revolution	2023-06-15	1234567891
403	305	Quantum Physics	2023-07-05	1234567893

➤ CREATED TABLE **ReturnStatus** AND INSERTED VALUES



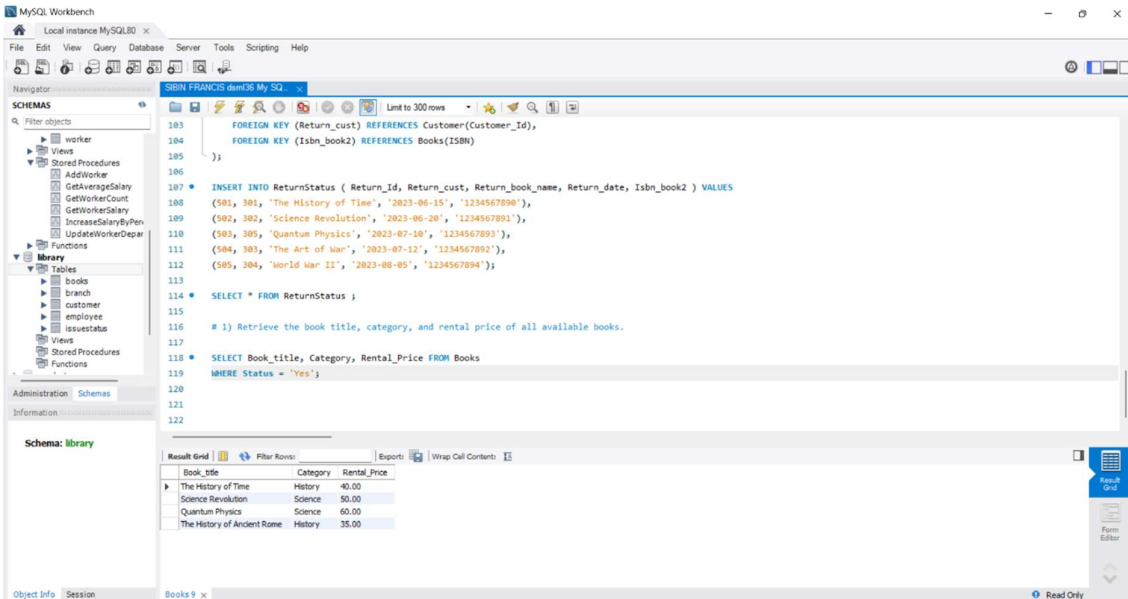
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the 'library' database. The main editor window shows the following SQL code:

```
96
97 CREATE TABLE ReturnStatus (
98     Return_Id INT PRIMARY KEY,
99     Return_cust INT,
100     Return_book_name VARCHAR(255),
101     Return_date DATE,
102     Isbn_book2 VARCHAR(13),
103     FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
104     FOREIGN KEY (Isbn_book2) REFERENCES Books(Isbn)
105 );
106
107 INSERT INTO ReturnStatus ( Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2 ) VALUES
108 (501, 301, 'The History of Time', '2023-06-15', '1234567890'),
109 (502, 302, 'Science Revolution', '2023-06-20', '1234567891'),
110 (503, 305, 'Quantum Physics', '2023-07-10', '1234567893'),
111 (504, 303, 'The Art of War', '2023-07-12', '1234567892'),
112 (505, 304, 'World War II', '2023-08-05', '1234567894');
113
114 SELECT * FROM ReturnStatus ;
115
```

The 'Result Grid' at the bottom displays the data inserted into the 'ReturnStatus' table:

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
501	301	The History of Time	2023-06-15	1234567890
502	302	Science Revolution	2023-06-20	1234567891
503	305	Quantum Physics	2023-07-10	1234567893
504	303	The Art of War	2023-07-12	1234567892
505	304	World War II	2023-08-05	1234567894

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



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the 'library' database. The main editor window shows the following SQL code:

```
103 FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
104 FOREIGN KEY (Isbn_book2) REFERENCES Books(Isbn)
105 );
106
107 INSERT INTO ReturnStatus ( Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2 ) VALUES
108 (501, 301, 'The History of Time', '2023-06-15', '1234567890'),
109 (502, 302, 'Science Revolution', '2023-06-20', '1234567891'),
110 (503, 305, 'Quantum Physics', '2023-07-10', '1234567893'),
111 (504, 303, 'The Art of War', '2023-07-12', '1234567892'),
112 (505, 304, 'World War II', '2023-08-05', '1234567894');
113
114 SELECT * FROM ReturnStatus ;
115
116 # 1) Retrieve the book title, category, and rental price of all available books.
117
118 SELECT Book_title, Category, Rental_Price FROM Books
119 WHERE Status = 'Yes';
120
121
122
```

The 'Result Grid' at the bottom displays the data retrieved from the 'Books' table:

Book_title	Category	Rental_Price
The History of Time	History	40.00
Science Revolution	Science	50.00
Quantum Physics	Science	60.00
The History of Ancient Rome	History	35.00

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

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
107 • INSERT INTO ReturnStatus ( Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2 ) VALUES
108 (501, 301, 'The History of Time', '2023-06-15', '1234567890'),
109 (502, 302, 'Science Revolution', '2023-06-20', '1234567891'),
110 (503, 305, 'Quantum Physics', '2023-07-10', '1234567893'),
111 (504, 303, 'The Art of War', '2023-07-12', '1234567892'),
112 (505, 304, 'World War II', '2023-08-05', '1234567894');
113
114 • SELECT * FROM ReturnStatus ;
115
116 # 1) Retrieve the book title, category, and rental price of all available books.
117
118 • SELECT Book_title, Category, Rental_Price FROM Books
119 WHERE Status = 'Yes';
120
121 # 2) List the employee names and their respective salaries in descending order of salary.
122
123 • SELECT Emp_Id, Emp_name, Salary FROM Employee
124 ORDER BY Salary DESC;
```

The result grid displays the following data:

Emp_Id	Emp_name	Salary
208	Ivy Adams	90000.00
201	Alice Smith	80000.00
203	Charlie Davis	75000.00 75000.00
209	Jack Evans	70000.00
202	Bob Johnson	60000.00
205	Frank White	55000.00
207	Henry Walker	52000.00
206	Grace Lee	45000.00
204	Eve Thompson	30000.00

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

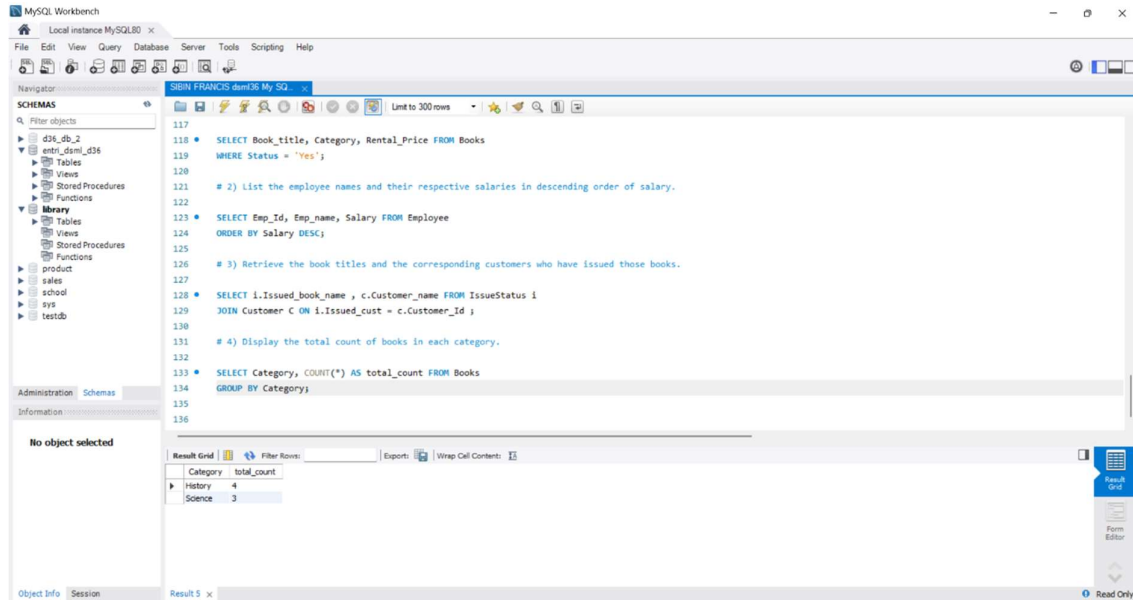
The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
112 (505, 304, 'World War II', '2023-08-05', '1234567894');
113
114 • SELECT * FROM ReturnStatus ;
115
116 # 1) Retrieve the book title, category, and rental price of all available books.
117
118 • SELECT Book_title, Category, Rental_Price FROM Books
119 WHERE Status = 'Yes';
120
121 # 2) List the employee names and their respective salaries in descending order of salary.
122
123 • SELECT Emp_Id, Emp_name, Salary FROM Employee
124 ORDER BY Salary DESC;
125
126 # 3) Retrieve the book titles and the corresponding customers who have issued those books.
127
128 • SELECT l.Issued_book_name , c.Customer_name FROM IssueStatus l
129 JOIN Customer c ON l.Issued_cust = c.Customer_Id ;
130
131
```

The result grid displays the following data:

Issued_book_name	Customer_name
The History of Time	Emily Brown
Science Revolution	Michael Green
Quantum Physics	Olivia Lewis

➤ Display the total count of books in each category.



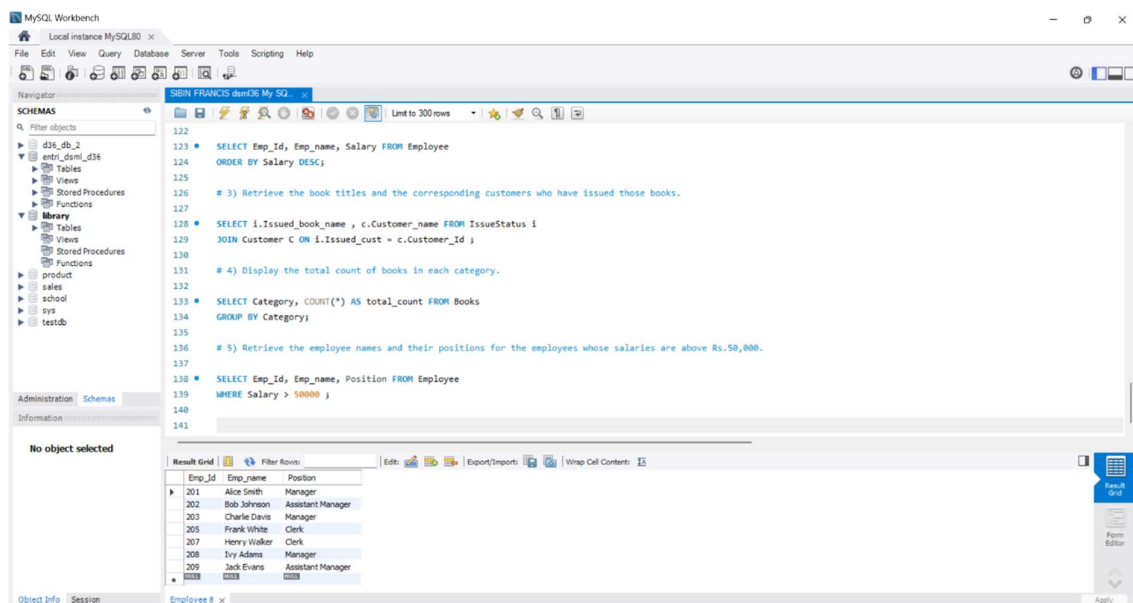
The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
117 *
118 * SELECT Book_title, Category, Rental_Price FROM Books
119 * WHERE Status = "Yes";
120 *
121 * # 2) List the employee names and their respective salaries in descending order of salary.
122 *
123 * SELECT Emp_Id, Emp_name, Salary FROM Employee
124 * ORDER BY Salary DESC;
125 *
126 * # 3) Retrieve the book titles and the corresponding customers who have issued those books.
127 *
128 * SELECT i.Issued_book_name , c.Customer_name FROM IssueStatus i
129 * JOIN Customer C ON i.Issued_cust = c.Customer_Id ;
130 *
131 * # 4) Display the total count of books in each category.
132 *
133 * SELECT Category, COUNT(*) AS total_count FROM Books
134 * GROUP BY Category;
135 *
136 *
```

The result grid shows the following data:

Category	total_count
History	4
Science	3

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



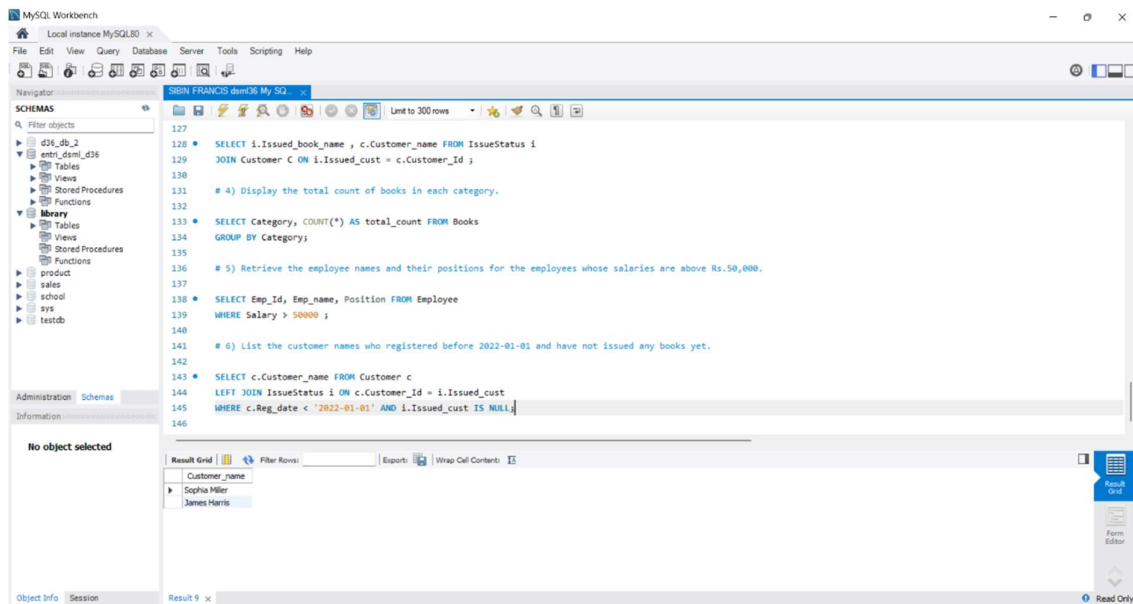
The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
122 *
123 * SELECT Emp_Id, Emp_name, Salary FROM Employee
124 * ORDER BY Salary DESC;
125 *
126 * # 3) Retrieve the book titles and the corresponding customers who have issued those books.
127 *
128 * SELECT i.Issued_book_name , c.Customer_name FROM IssueStatus i
129 * JOIN Customer C ON i.Issued_cust = c.Customer_Id ;
130 *
131 * # 4) Display the total count of books in each category.
132 *
133 * SELECT Category, COUNT(*) AS total_count FROM Books
134 * GROUP BY Category;
135 *
136 * # 5) Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
137 *
138 * SELECT Emp_Id, Emp_name, Position FROM Employee
139 * WHERE Salary > 50000 ;
140 *
141 *
```

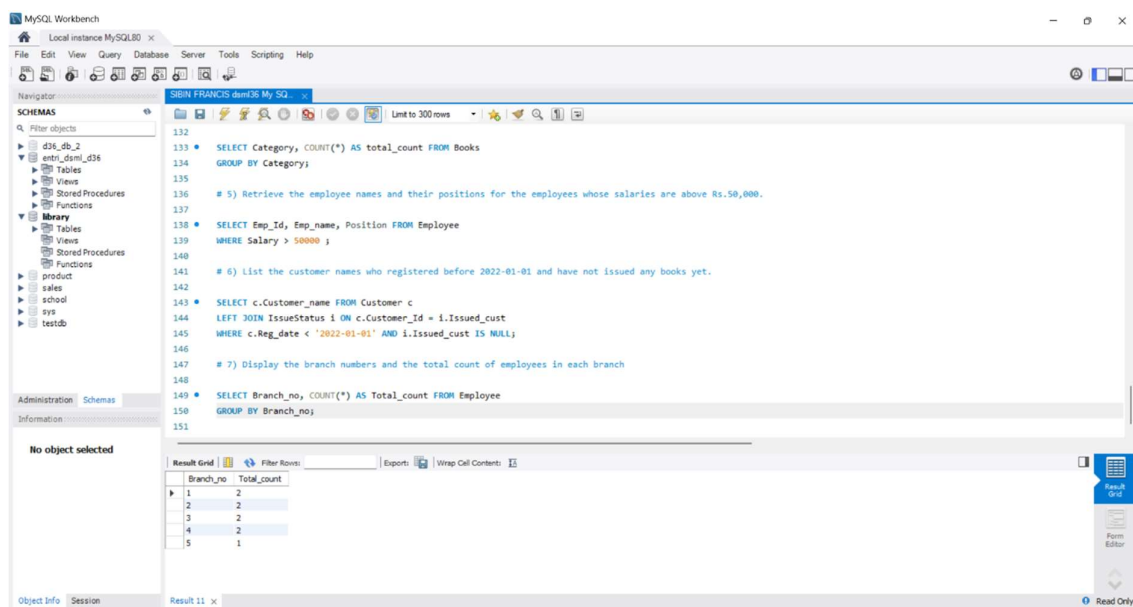
The result grid shows the following data:

Emp_Id	Emp_name	Position
201	Alice Smith	Manager
202	Bob Johnson	Assistant Manager
203	Charlie Davis	Manager
205	Frank White	Clerk
207	Henry Walker	Clerk
208	Ivy Adams	Manager
209	Jack Evans	Assistant Manager

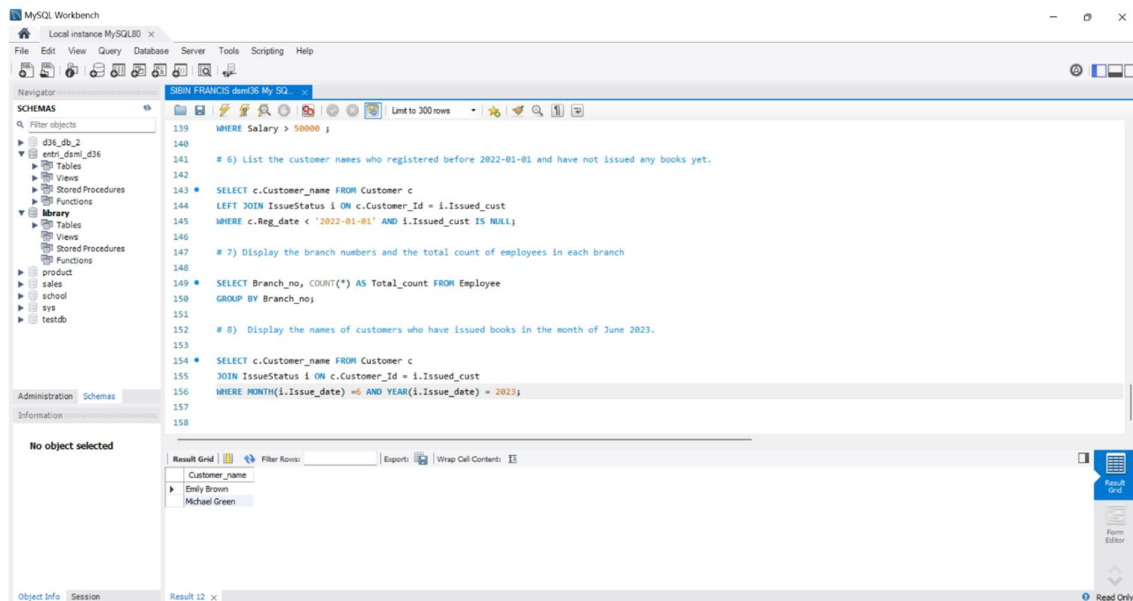
- List the customer names who registered before 2022-01-01 and have not issued any books yet.



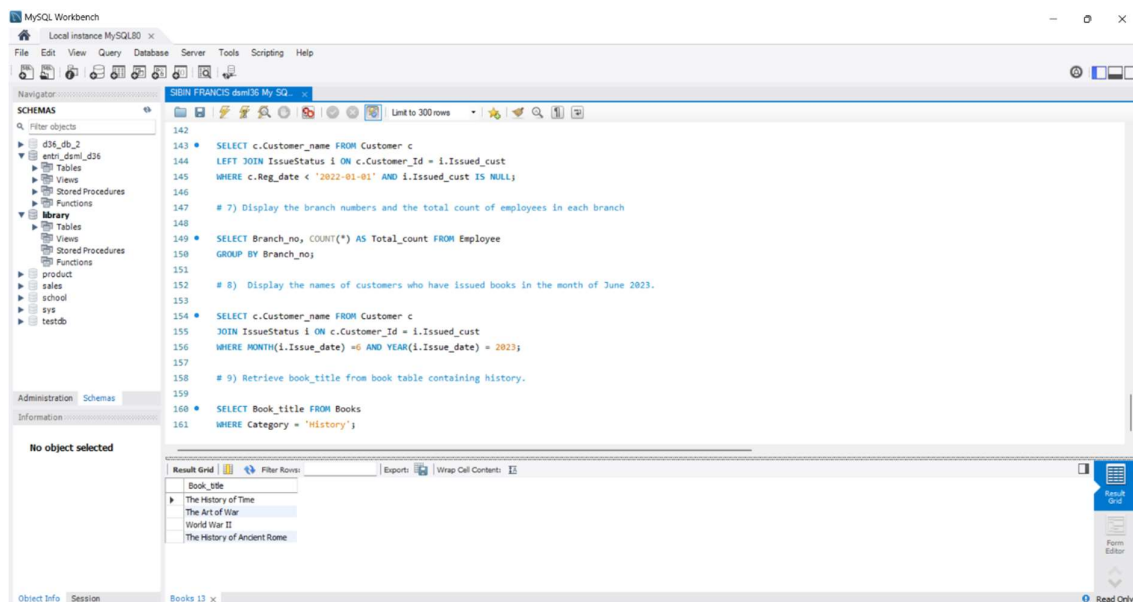
- Display the branch numbers and the total count of employees in each branch



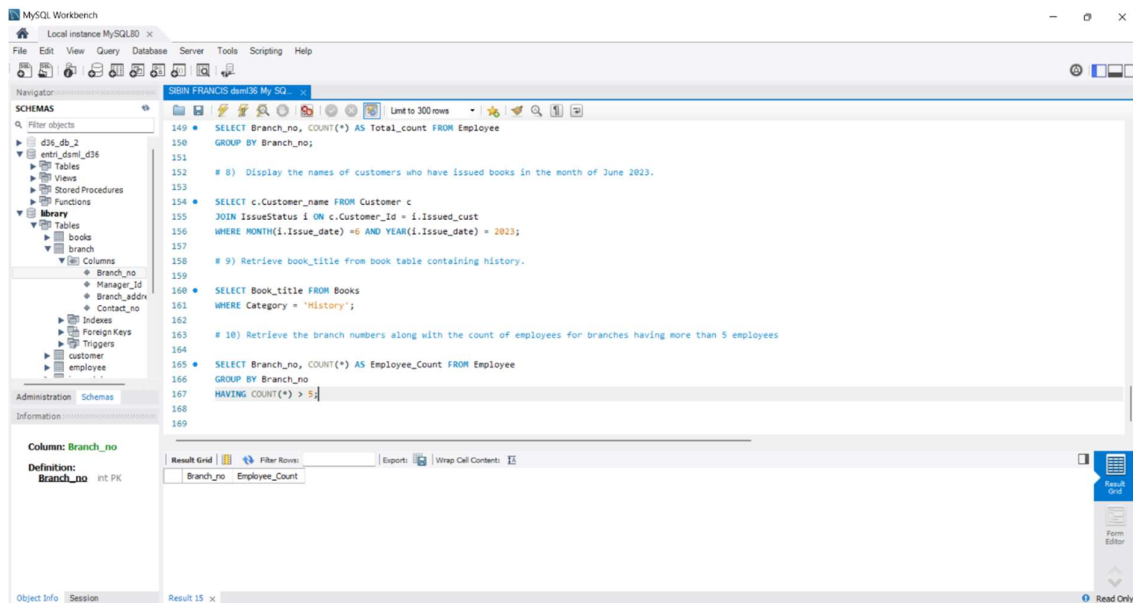
- Display the names of customers who have issued books in the month of June 2023.



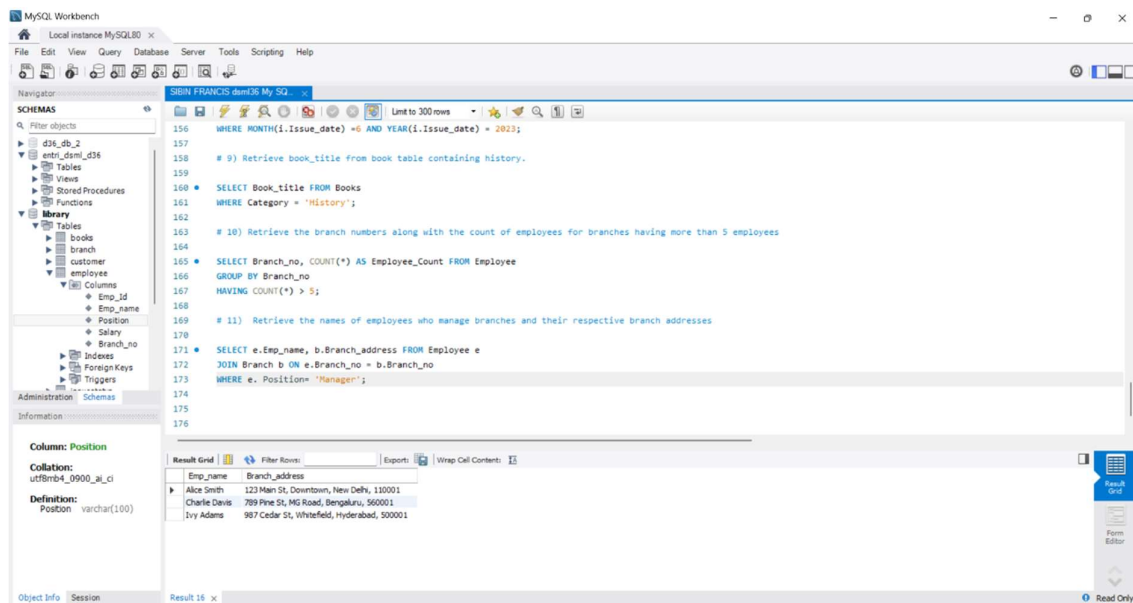
- Retrieve book_title from book table containing history.



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



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



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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with a search filter 'Filter objects'. The 'customer' schema is selected, showing tables like 'books', 'branch', 'customer', and 'employee'. The 'employee' table is expanded, showing columns: Emp_Id, Emp_name, Position, Salary, and Branch_no. The 'Position' column is highlighted. The main editor window shows a SQL script with two queries. The first query is a comment: '# 11) Retrieve the names of employees who manage branches and their respective branch addresses'. The second query is: '# 12) Display the names of customers who have issued books with a rental price higher than Rs. 25.' The query is executed, and the results are displayed in the 'Result Grid' at the bottom. The results show the 'Customer_name' for three customers: Emily Brown, Michael Green, and Olivia Lewis.

```
167  HAVING COUNT(*) > 5;
168
169  # 11) Retrieve the names of employees who manage branches and their respective branch addresses
170
171  SELECT e.Emp_name, b.Branch_address FROM Employee e
172  JOIN Branch b ON e.Branch_no = b.Branch_no
173  WHERE e.Position = 'Manager';
174
175  # 12) Display the names of customers who have issued books with a rental price higher than Rs. 25.
176
177  SELECT c.Customer_name FROM Customer c
178  JOIN IssueStatus i ON c.Customer_Id = i.Issued_cust
179  JOIN Books b ON i.Isbn_book = b.ISBN
180  WHERE b.Rental_Price > 25;
181
182
183
184
185
186
187
```

Column: Position
Collation: utf8mb4_0900_ai_ci
Definition: Position varchar(100)

Customer_name
Emily Brown
Michael Green
Olivia Lewis