Mysql Comprehensive Assessment

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:

```
create DATABASE library ;

USE library;
```

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

```
CREATE TABLE Branch(

  Branch_no INT NOT NULL auto_increment,

  Manager_Id INT NOT NULL,

  Branch_address  VARCHAR(100) NOT NULL,

  Contact_no VARCHAR(20) NOT NULL,

  primary key(Branch_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

```
CREATE TABLE Employee(

  Emp_Id INT NOT NULL auto_increment,

  Emp_name  VARCHAR(50) NOT NULL,

  Position VARCHAR(50),

  Salary DECIMAL(10,2),

  Branch_no INT,

  primary key(Emp_Id),

  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

```
CREATE TABLE Books(

  ISBN INT NOT NULL auto_increment,

  Book_title  VARCHAR(50) NOT NULL,

  Category  VARCHAR(50) NOT NULL,

  Rental_Price  DECIMAL(10,2) NOT NULL,

  Status VARCHAR(10) NOT NULL COMMENT 'Give yes if book available and no if book not available',

  Author varchar(50) NOT NULL,

  Publisher varchar(50) NOT NULL,

  primary key(ISBN)

);
```

4. Customer

• Customer_Id Set as PRIMARY KEY
• Customer_name

• Customer_address
• Reg_date

```
create table Customer(
   Customer_Id int not null auto_increment,
   Customer_name varchar(50) not null,
   Customer_address varchar(100) not null,
   Reg_date date not null,
   primary key(Customer_Id)
);
```

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 not null auto_increment,

   Issued_cust int not null,

   Issue_date date not null,

   Isbn_book int not null,

   primary key(Issue_Id),

   foreign key(Issued_cust) references Customer(Customer_Id),

   foreign key(Isbn_book) references Books(ISBN)

);
```

6. ReturnStatus

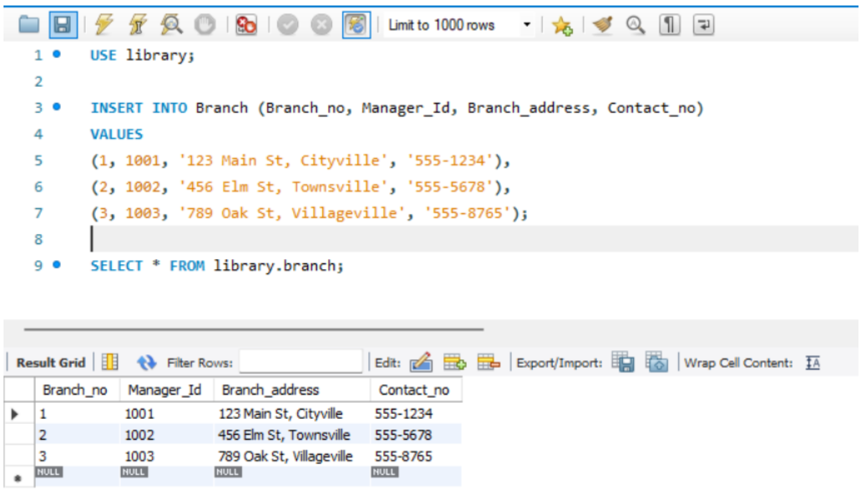
• Return_Id

- 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 not null auto_increment,

  Return_cust int,

  Return_book_name varchar(50),
  Return_date date not null,
  Isbn_book2 int not null,
  primary key(Return_Id),
  foreign key(Isbn_book2) references Books(ISBN),
  foreign key(Return_cust) references Customer(Customer_Id)
);
```

Insert values into the table

```
INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
VALUES
(1, 1001, '123 Main St, Cityville', '555-1234'),
(2, 1002, '456 Elm St, Townsville', '555-5678'),
(3, 1003, '789 Oak St, Villageville', '555-8765');
```



```
USE library;

INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
VALUES
(1, 1001, '123 Main St, Cityville', '555-1234'),
(2, 1002, '456 Elm St, Townsville', '555-5678'),
(3, 1003, '789 Oak St, Villageville', '555-8765');

SELECT * FROM library.branch;
```

Branch_no	Manager_Id	Branch_address	Contact_no
1	1001	123 Main St, Cityville	555-1234
2	1002	456 Elm St, Townsville	555-5678
3	1003	789 Oak St, Villageville	555-8765
NULL	NULL	NULL	NULL

```
INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no)
VALUES
(1, 'Alice Smith', 'Manager', 60000, 1),
(2, 'Bob Johnson', 'Assistant', 40000, 1),
(3, 'Carol White', 'Manager', 65000, 2),
(4, 'David Brown', 'Clerk', 35000, 2),
(5, 'Eve Davis', 'Assistant', 42000, 3),
```

(6, 'Frank Miller', 'Clerk', 37000, 3),

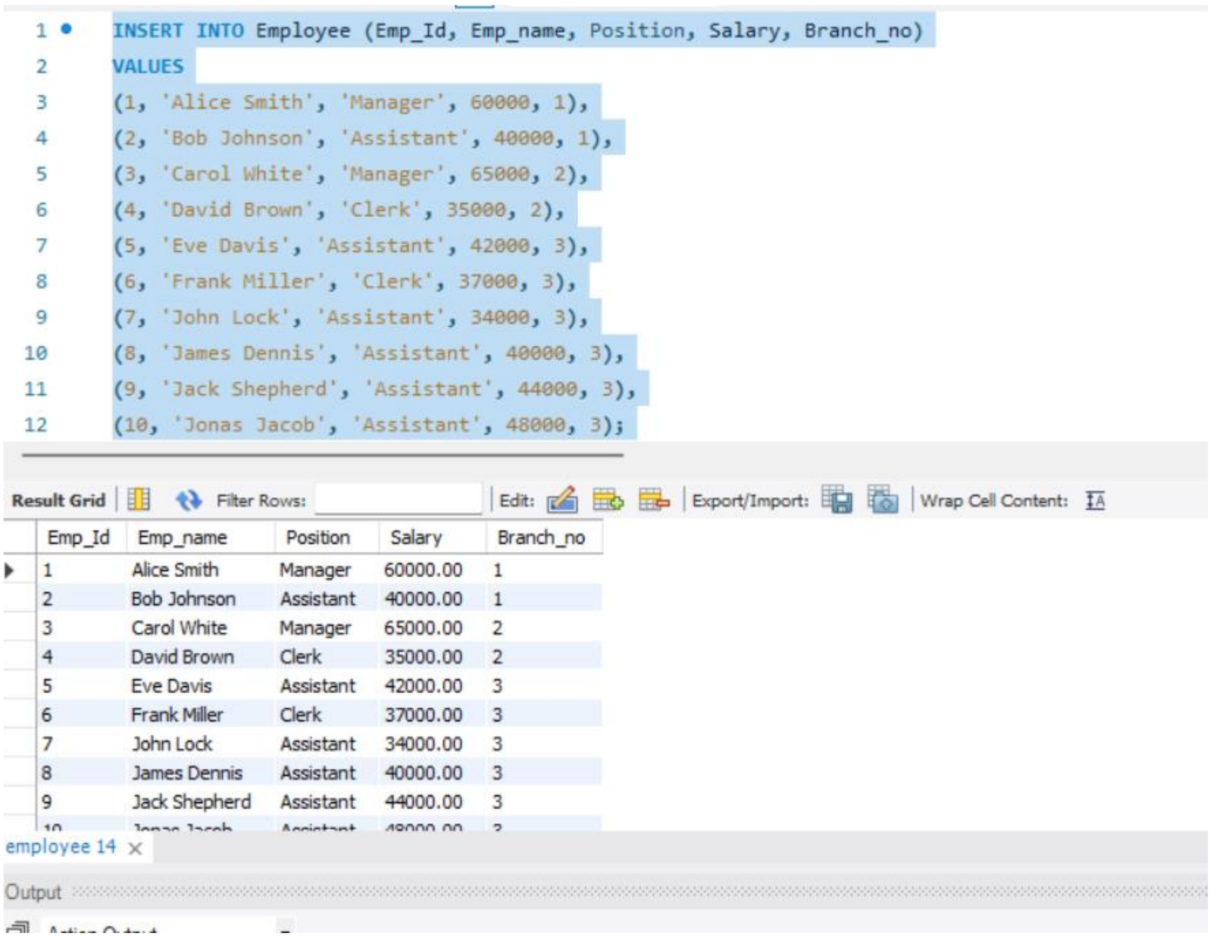
(7, 'John Lock', 'Assistant', 34000, 3),

(8, 'James Dennis', 'Assistant', 40000, 3),

(9, 'Jack Shepherd', 'Assistant', 44000, 3),

(10, 'Jonas Jacob', 'Assistant', 48000, 3);

SELECT * FROM employee;



```
INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no)
VALUES
(1, 'Alice Smith', 'Manager', 60000, 1),
(2, 'Bob Johnson', 'Assistant', 40000, 1),
(3, 'Carol White', 'Manager', 65000, 2),
(4, 'David Brown', 'Clerk', 35000, 2),
(5, 'Eve Davis', 'Assistant', 42000, 3),
(6, 'Frank Miller', 'Clerk', 37000, 3),
(7, 'John Lock', 'Assistant', 34000, 3),
(8, 'James Dennis', 'Assistant', 40000, 3),
(9, 'Jack Shepherd', 'Assistant', 44000, 3),
(10, 'Jonas Jacob', 'Assistant', 48000, 3);
```

Emp_Id	Emp_name	Position	Salary	Branch_no
1	Alice Smith	Manager	60000.00	1
2	Bob Johnson	Assistant	40000.00	1
3	Carol White	Manager	65000.00	2
4	David Brown	Clerk	35000.00	2
5	Eve Davis	Assistant	42000.00	3
6	Frank Miller	Clerk	37000.00	3
7	John Lock	Assistant	34000.00	3
8	James Dennis	Assistant	40000.00	3
9	Jack Shepherd	Assistant	44000.00	3
10	Jonas Jacob	Assistant	48000.00	3

INSERT INTO Books (Book_title, Category, Rental_Price, Status, Author, Publisher)

VALUES

('The Great Gatsby', 'Fiction', 30.00, 'yes', 'F. Scott Fitzgerald', 'Scribner'),

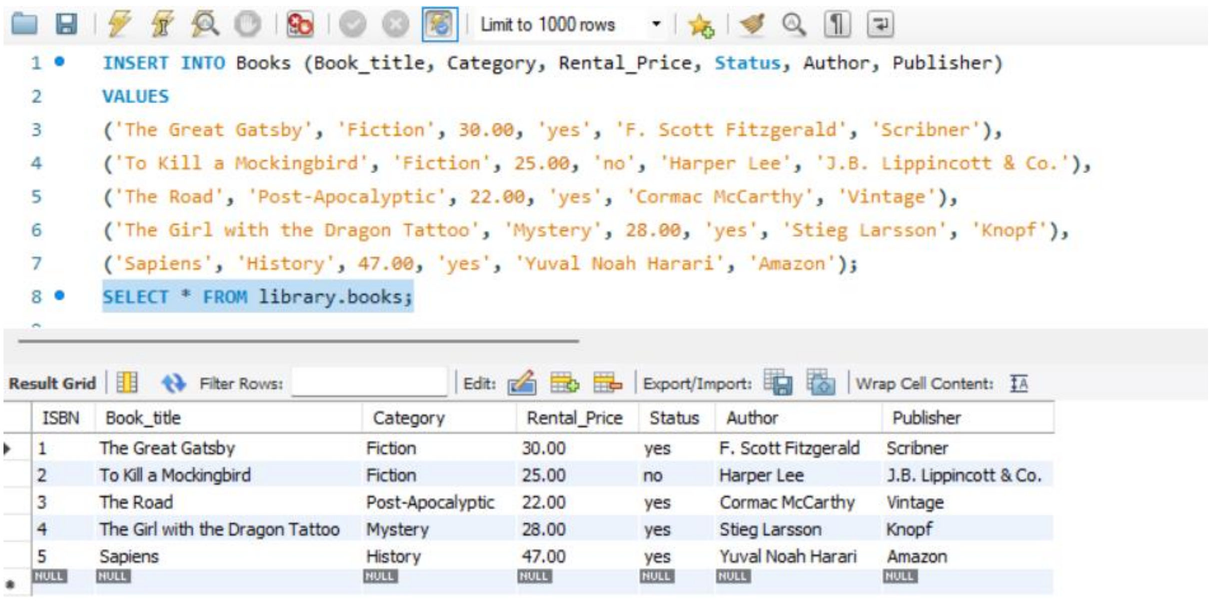
('To Kill a Mockingbird', 'Fiction', 25.00, 'no', 'Harper Lee', 'J.B. Lippincott & Co.'),

('The Road', 'Post-Apocalyptic', 22.00, 'yes', 'Cormac McCarthy', 'Vintage'),

('The Girl with the Dragon Tattoo', 'Mystery', 28.00, 'yes', 'Stieg Larsson', 'Knopf'),

('Sapiens', 'History', 47.00, 'yes', 'Yuval Noah Harari', 'Amazon');

SELECT * FROM library.books;



```
INSERT INTO Books (Book_title, Category, Rental_Price, Status, Author, Publisher)
VALUES
('The Great Gatsby', 'Fiction', 30.00, 'yes', 'F. Scott Fitzgerald', 'Scribner'),
('To Kill a Mockingbird', 'Fiction', 25.00, 'no', 'Harper Lee', 'J.B. Lippincott & Co.'),
('The Road', 'Post-Apocalyptic', 22.00, 'yes', 'Cormac McCarthy', 'Vintage'),
('The Girl with the Dragon Tattoo', 'Mystery', 28.00, 'yes', 'Stieg Larsson', 'Knopf'),
('Sapiens', 'History', 47.00, 'yes', 'Yuval Noah Harari', 'Amazon');
SELECT * FROM library.books;
```

ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
1	The Great Gatsby	Fiction	30.00	yes	F. Scott Fitzgerald	Scribner
2	To Kill a Mockingbird	Fiction	25.00	no	Harper Lee	J.B. Lippincott & Co.
3	The Road	Post-Apocalyptic	22.00	yes	Cormac McCarthy	Vintage
4	The Girl with the Dragon Tattoo	Mystery	28.00	yes	Stieg Larsson	Knopf
5	Sapiens	History	47.00	yes	Yuval Noah Harari	Amazon
NULL	NULL	NULL	NULL	NULL	NULL	NULL

INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date)

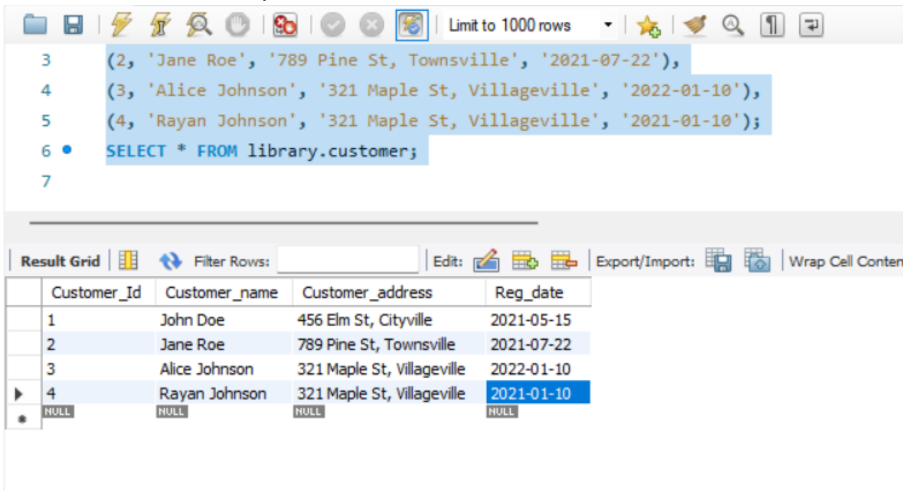
VALUES (1, 'John Doe', '456 Elm St, Cityville', '2021-05-15'),

(2, 'Jane Roe', '789 Pine St, Townsville', '2021-07-22'),

(3, 'Alice Johnson', '321 Maple St, Villageville', '2022-01-10'),

(4, 'Rayan Johnson', '321 Maple St, Villageville', '2021-01-10');

SELECT * FROM library.customer;



```
(2, 'Jane Roe', '789 Pine St, Townsville', '2021-07-22'),
(3, 'Alice Johnson', '321 Maple St, Villageville', '2022-01-10'),
(4, 'Rayan Johnson', '321 Maple St, Villageville', '2021-01-10');
SELECT * FROM library.customer;
```

Customer_Id	Customer_name	Customer_address	Reg_date
1	John Doe	456 Elm St, Cityville	2021-05-15
2	Jane Roe	789 Pine St, Townsville	2021-07-22
3	Alice Johnson	321 Maple St, Villageville	2022-01-10
4	Rayan Johnson	321 Maple St, Villageville	2021-01-10
NULL	NULL	NULL	NULL

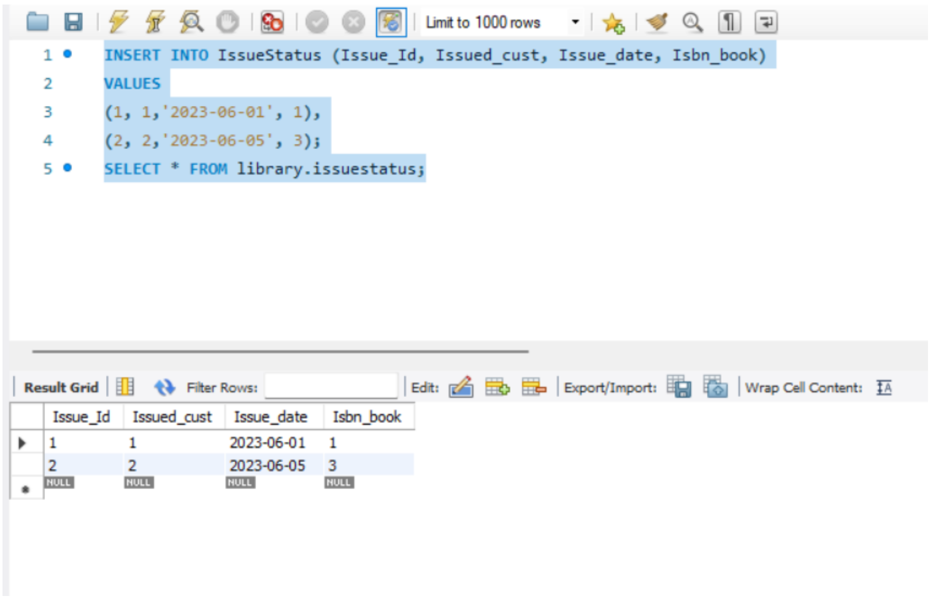
INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issue_date, Isbn_book)

VALUES

(1, 1,'2023-06-01', 1),

(2, 2,'2023-06-05', 3);

SELECT * FROM library.issuestatus;



```
INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issue_date, Isbn_book)
VALUES
(1, 1,'2023-06-01', 1),
(2, 2,'2023-06-05', 3);
SELECT * FROM library.issuestatus;
```

Issue_Id	Issued_cust	Issue_date	Isbn_book
1	1	2023-06-01	1
2	2	2023-06-05	3
NULL	NULL	NULL	NULL

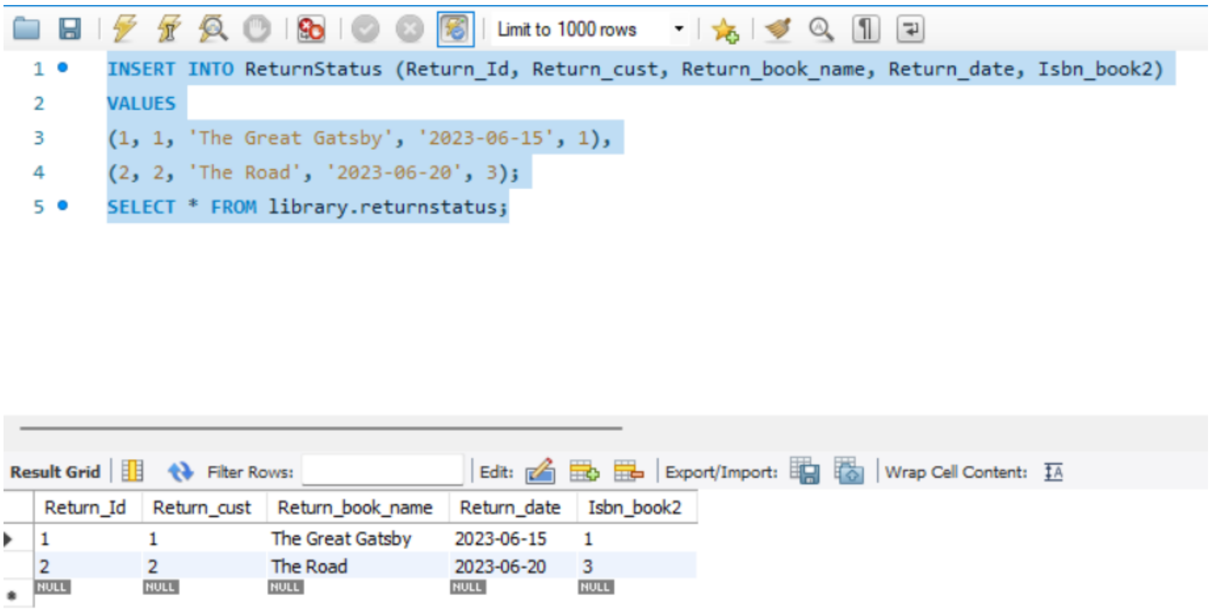
INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)

VALUES

(1, 1, 'The Great Gatsby', '2023-06-15', 1),

(2, 2, 'The Road', '2023-06-20', 3);

SELECT * FROM library.returnstatus;

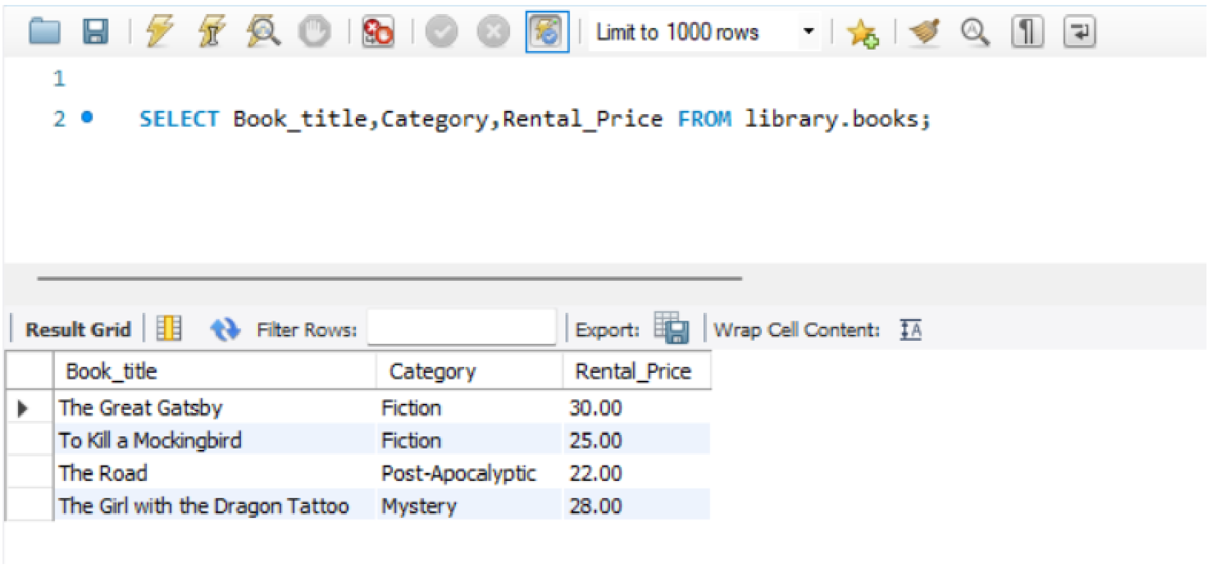


```
INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)
VALUES
(1, 1, 'The Great Gatsby', '2023-06-15', 1),
(2, 2, 'The Road', '2023-06-20', 3);
SELECT * FROM library.returnstatus;
```

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
1	1	The Great Gatsby	2023-06-15	1
2	2	The Road	2023-06-20	3
NULL	NULL	NULL	NULL	NULL

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

   SELECT Book_title,Category,Rental_Price FROM library.books;

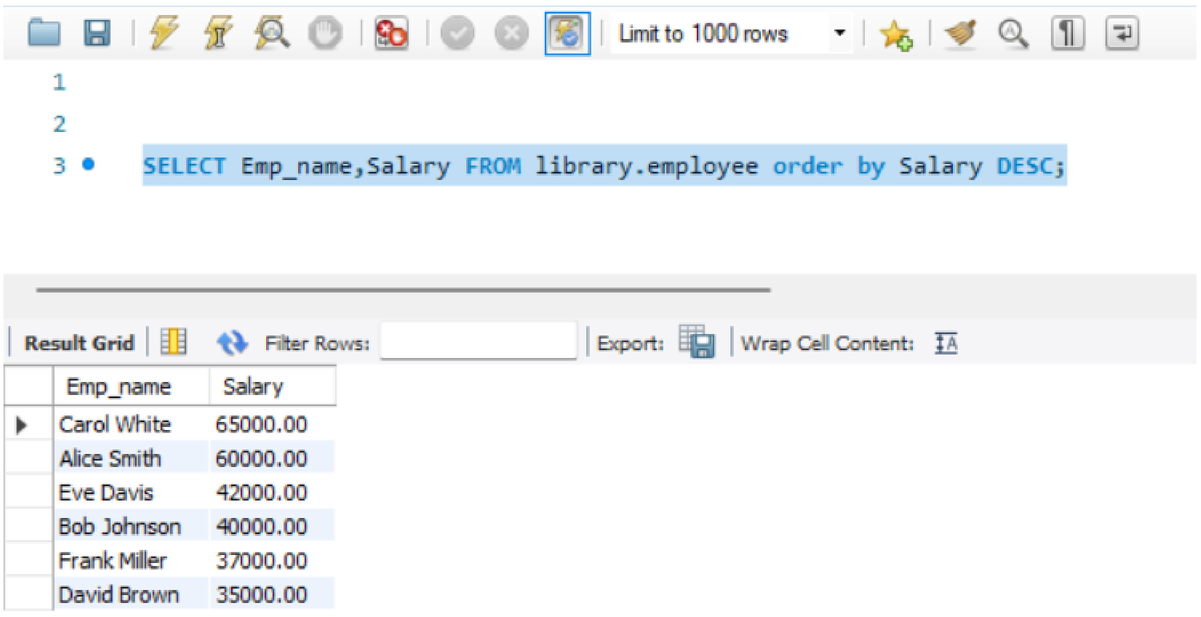


```
SELECT Book_title,Category,Rental_Price FROM library.books;
```

Book_title	Category	Rental_Price
The Great Gatsby	Fiction	30.00
To Kill a Mockingbird	Fiction	25.00
The Road	Post-Apocalyptic	22.00
The Girl with the Dragon Tattoo	Mystery	28.00

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

SELECT Emp_name,Salary FROM library.employee order by Salary DESC;

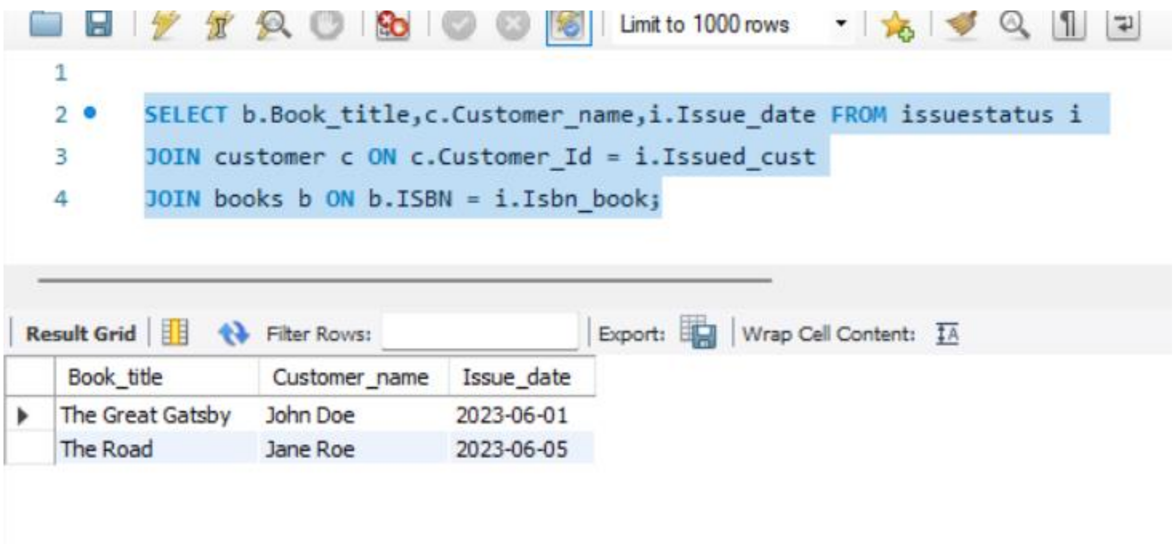


```
SELECT Emp_name,Salary FROM library.employee order by Salary DESC;
```

Emp_name	Salary
Carol White	65000.00
Alice Smith	60000.00
Eve Davis	42000.00
Bob Johnson	40000.00
Frank Miller	37000.00
David Brown	35000.00

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

```
 SELECT b.Book_title,c.Customer_name,i.Issue_date FROM issuestatus i
JOIN customer c ON c.Customer_Id = i.Issued_cust
JOIN books b ON b.ISBN = i.Isbn_book;
```

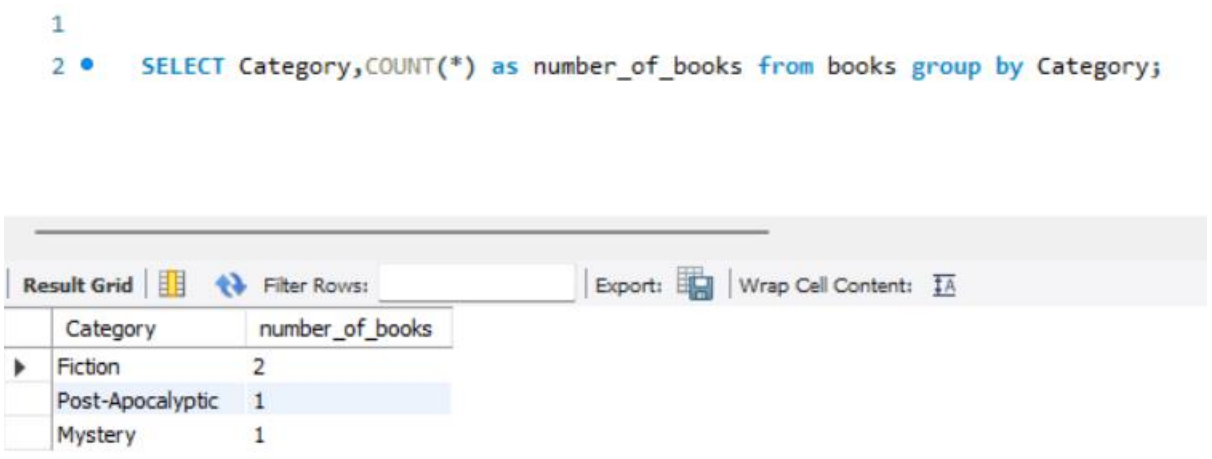


```
SELECT b.Book_title,c.Customer_name,i.Issue_date FROM issuestatus i
JOIN customer c ON c.Customer_Id = i.Issued_cust
JOIN books b ON b.ISBN = i.Isbn_book;
```

Book_title	Customer_name	Issue_date
The Great Gatsby	John Doe	2023-06-01
The Road	Jane Roe	2023-06-05

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

SELECT Category,COUNT(*) as number_of_books from books group by Category;

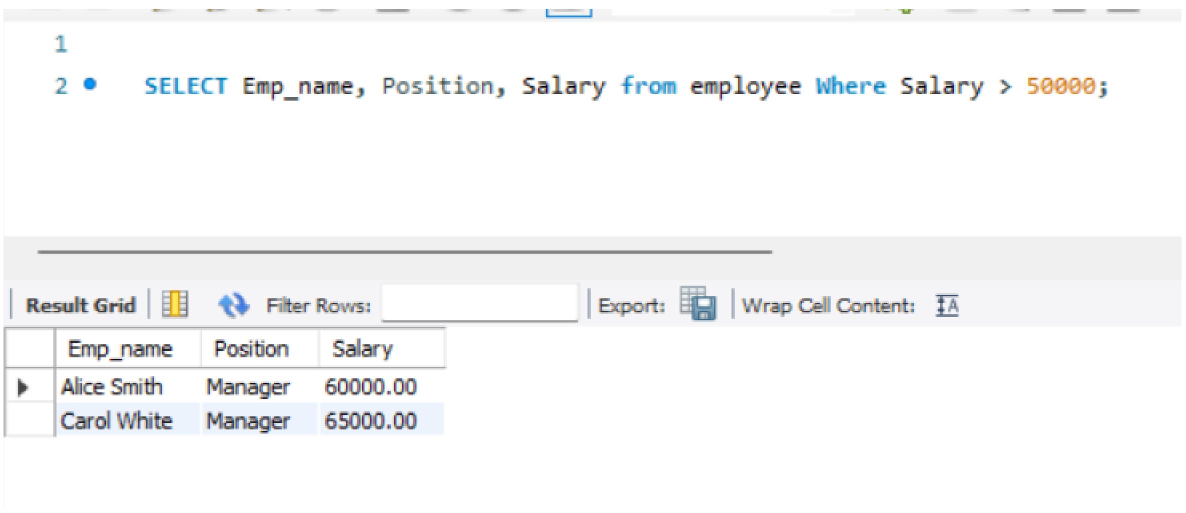


```
SELECT Category,COUNT(*) as number_of_books from books group by Category;
```

Category	number_of_books
Fiction	2
Post-Apocalyptic	1
Mystery	1

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

SELECT Emp_name, Position, Salary from employee Where Salary > 50000;



```
SELECT Emp_name, Position, Salary from employee Where Salary > 50000;
```

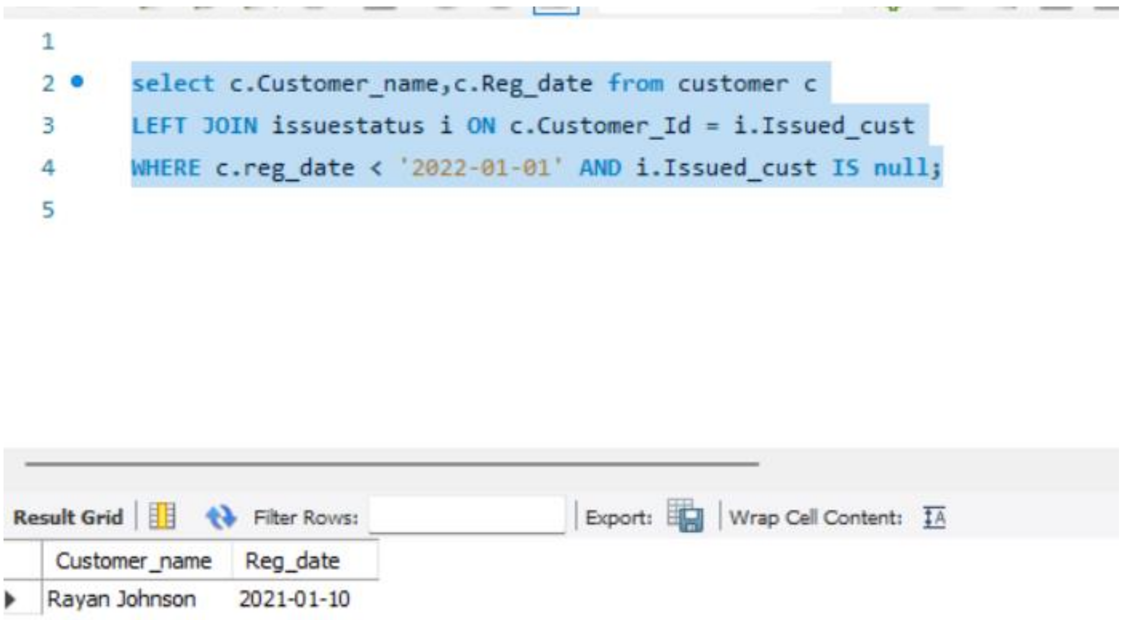
Emp_name	Position	Salary
Alice Smith	Manager	60000.00
Carol White	Manager	65000.00

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

select c.Customer_name,c.Reg_date from customer c

LEFT JOIN issuestatus i ON c.Customer_Id = i.Issued_cust

WHERE c.reg_date < '2022-01-01' AND i.Issued_cust IS null;



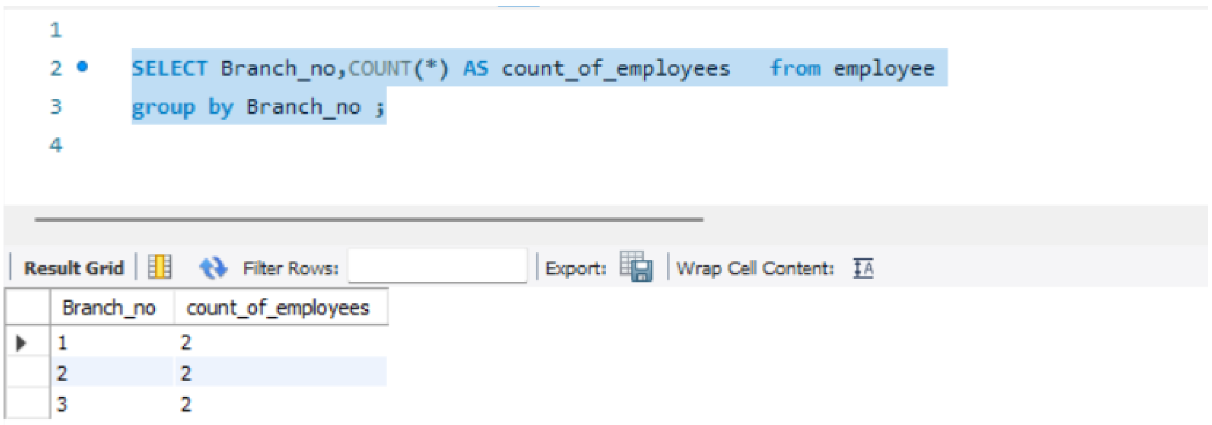
```
select c.Customer_name,c.Reg_date from customer c
LEFT JOIN issuestatus i ON c.Customer_Id = i.Issued_cust
WHERE c.reg_date < '2022-01-01' AND i.Issued_cust IS null;
```

Customer_name	Reg_date
Rayan Johnson	2021-01-10

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

SELECT Branch_no,COUNT(*) AS count_of_employees   from employee

group by Branch_no ;



```
SELECT Branch_no,COUNT(*) AS count_of_employees    from employee
group by Branch_no ;
```

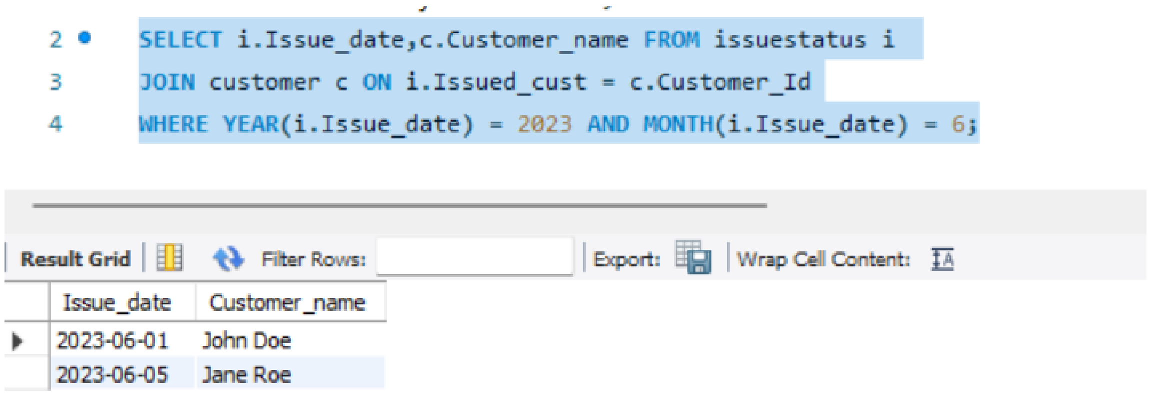
Branch_no	count_of_employees
1	2
2	2
3	2

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

SELECT i.Issue_date,c.Customer_name FROM issuestatus i

JOIN customer c ON i.Issued_cust = c.Customer_Id

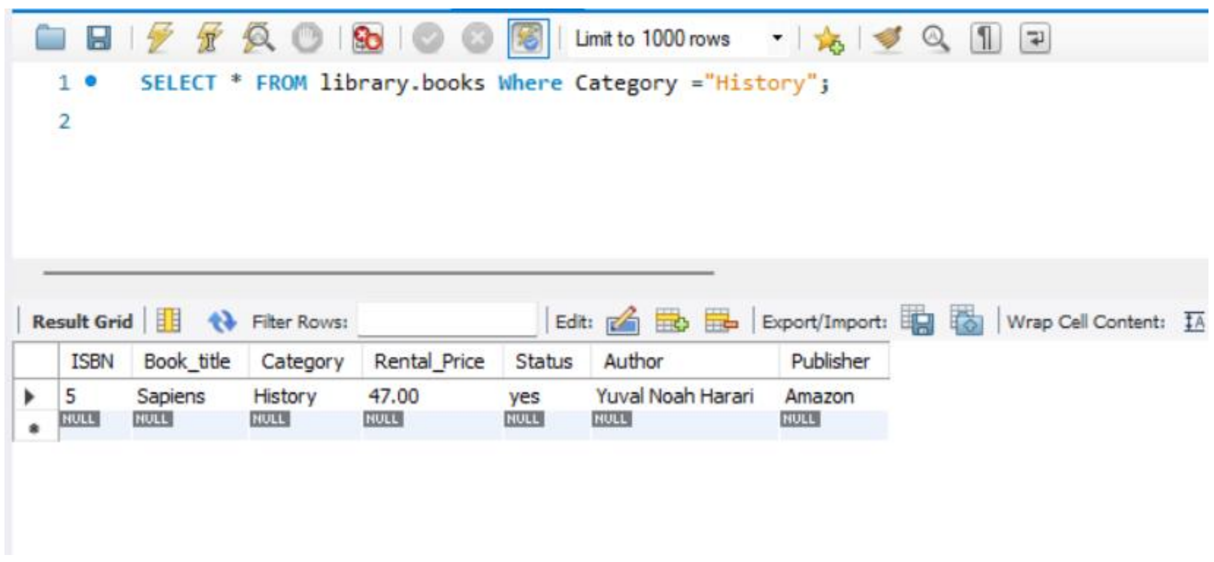
WHERE YEAR(i.Issue_date) = 2023 AND MONTH(i.Issue_date) = 6;



```sql
SELECT i.Issue_date,c.Customer_name FROM issuestatus i
JOIN customer c ON i.Issued_cust = c.Customer_Id
WHERE YEAR(i.Issue_date) = 2023 AND MONTH(i.Issue_date) = 6;
```

Issue_date	Customer_name
2023-06-01	John Doe
2023-06-05	Jane Roe

9. Retrieve book_title from book table containing history.



```sql
SELECT * FROM library.books Where Category ="History";
```

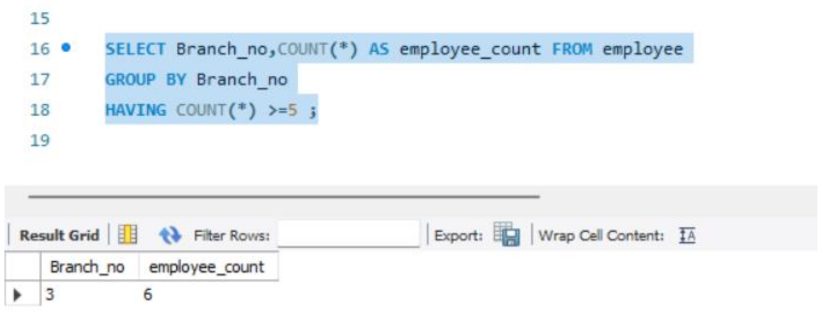
ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
5	Sapiens	History	47.00	yes	Yuval Noah Harari	Amazon
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

SELECT Branch_no,COUNT(*) AS employee_count FROM employee

GROUP BY Branch_no

HAVING COUNT(*) >=5 ;



```sql
SELECT Branch_no,COUNT(*) AS employee_count FROM employee
GROUP BY Branch_no
HAVING COUNT(*) >=5 ;
```

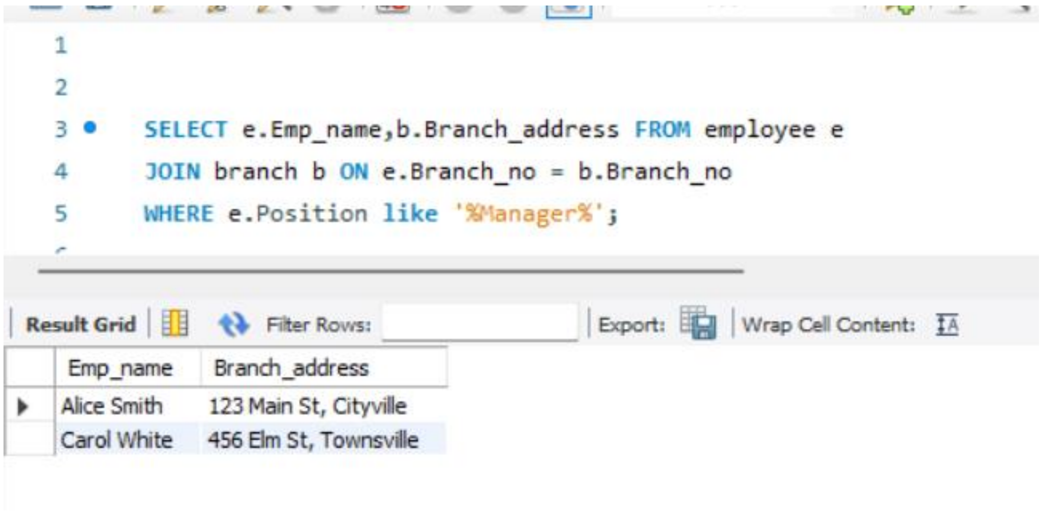
Branch_no	employee_count
3	6

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

SELECT e.Emp_name,b.Branch_address FROM employee e

JOIN branch b ON e.Branch_no = b.Branch_no

WHERE e.Position like '%Manager%';



```
SELECT e.Emp_name,b.Branch_address FROM employee e
JOIN branch b ON e.Branch_no = b.Branch_no
WHERE e.Position like '%Manager%';
```

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

Emp_name	Branch_address
Alice Smith	123 Main St, Cityville
Carol White	456 Elm St, Townsville

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

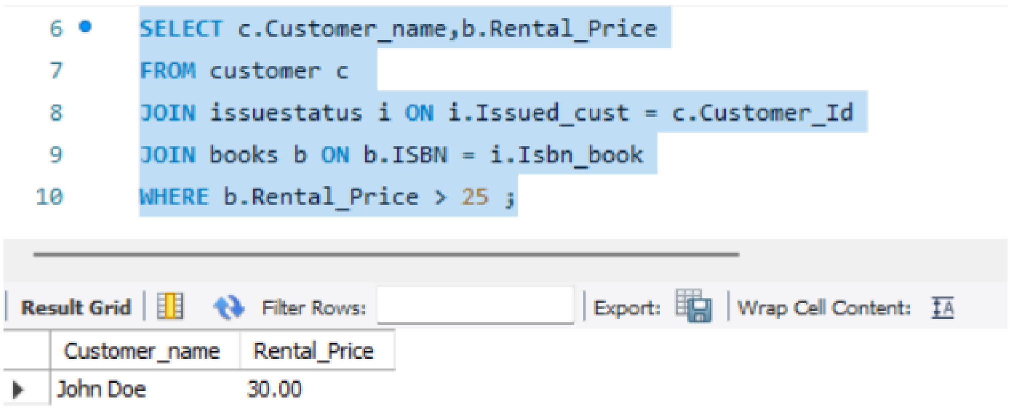
SELECT c.Customer_name,b.Rental_Price

FROM customer c

JOIN issuestatus i ON i.Issued_cust = c.Customer_Id

JOIN books b ON b.ISBN = i.Isbn_book

WHERE b.Rental_Price > 25 ;



```
SELECT c.Customer_name,b.Rental_Price
FROM customer c
JOIN issuestatus i ON i.Issued_cust = c.Customer_Id
JOIN books b ON b.ISBN = i.Isbn_book
WHERE b.Rental_Price > 25 ;
```

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

Customer_name	Rental_Price
John Doe	30.00