Library Management System

1 . CREATE DATABASE library;

```
20  23:10:33  CREATE DATABASE library
```

2.

a)

```
CREATE TABLE Branch (
   Branch_no INT PRIMARY KEY,
   Manager_Id INT,
   Branch_address VARCHAR(100),
   Contact_no VARCHAR(15)
);
INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
VALUES
   (1, 101, '123 Main St', '555-1234'),
   (2, 102, '456 Elm St', '555-5678'),
   (3, 103, '789 Oak St', '555-9876'),
   (4, 104, '321 Pine St', '555-4321'),
   (5, 105, '654 Cedar St', '555-8765');
```

b)

```
CREATE TABLE Employee (
   Emp_Id INT PRIMARY KEY,
   Emp_name VARCHAR(50),
   Position VARCHAR(50),
   Salary DECIMAL(10, 2),
   Branch_no INT,
   FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
);

INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no)
VALUES
   (1, 'John Smith', 'Manager', RAND() * 10000 + 50000, 1),
   (2, 'Jane Doe', 'Assistant', RAND() * 5000 + 30000, 2),
   (3, 'Michael Johnson', 'Librarian', RAND() * 4000 + 25000, 1),
   (4, 'Emily Brown', 'Clerk', RAND() * 3000 + 20000, 3),
   (5, 'Daniel Martinez', 'Assistant', RAND() * 5000 + 30000, 2);
```

c)

```
CREATE TABLE Books (
   ISBN VARCHAR(20) PRIMARY KEY,
   Book_title VARCHAR(100),
   Category VARCHAR(50),
   Rental_Price DECIMAL(10, 2),
   Status ENUM('yes', 'no'),
   Author VARCHAR(50),
   Publisher VARCHAR(50)
);
INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)
VALUES
   ('978-3-16-148410-0', 'The Great Gatsby', 'Fiction', RAND() * 10 + 10, 'yes', 'F. Scott Fitzgerald', 'Scribner'),
   ('978-0-306-40615-7', '1984', 'Dystopian', RAND() * 8 + 8, 'yes', 'George Orwell', 'Secker & Warburg'),
   ('978-0-8129-7263-5', 'To Kill a Mockingbird', 'Classic', RAND() * 9 + 9, 'yes', 'Harper Lee', 'HarperCollins'),
   ('978-0-451-52360-4', 'The Catcher in the Rye', 'Coming-of-Age', RAND() * 7 + 7, 'yes', 'J.D. Salinger', 'Little, Brown'),
   ('978-1-101-93575-9', 'Pride and Prejudice', 'Romance', RAND() * 11 + 11, 'yes', 'Jane Austen', 'T. Egerton, Whitehall');
```

d)

```
CREATE TABLE Customer (

   Customer_Id INT PRIMARY KEY,

   Customer_name VARCHAR(50),

   Customer_address VARCHAR(100),

   Reg_date DATE

);

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

VALUES

   (1, 'Alice Johnson', '789 Maple Ave', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 10) YEAR)),

   (2, 'Bob Smith', '456 Birch St', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 10) YEAR)),

   (3, 'Carol Williams', '321 Cedar Ln', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 10) YEAR)),

   (4, 'David Brown', '987 Oak Dr', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 10) YEAR)),

   (5, 'Emma Garcia', '654 Pine Rd', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 10) YEAR));
```

e)

```
CREATE TABLE IssueStatus (

   Issue_Id INT PRIMARY KEY,

   Issued_cust INT,

   Issued_book_name VARCHAR(100),

   Issue_date DATE,

   Isbn_book VARCHAR(20),

   FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),

   FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)

);

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

VALUES
```

```
    (1, 1, 'The Great Gatsby', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 30) DAY), '978-3-16-148410-0'),

    (2, 2, '1984', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 30) DAY), '978-0-306-40615-7'),

    (3, 3, 'To Kill a Mockingbird', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 30) DAY), '978-0-8129-7263-5'),

    (4, 4, 'The Catcher in the Rye', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 30) DAY), '978-0-451-52360-4'),

    (5, 5, 'Pride and Prejudice', DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 30) DAY), '978-1-101-93575-9');
```

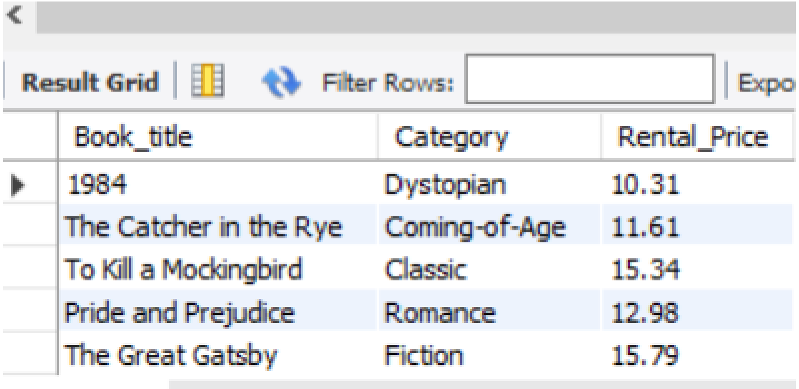
f)

```
CREATE TABLE ReturnStatus (

    Return_Id INT PRIMARY KEY,

    Return_cust INT,

    Return_book_name VARCHAR(100),

    Return_date DATE,

    Isbn_book2 VARCHAR(20),

    FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),

    FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)

);

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

SELECT

    i.Issue_Id,

    i.Issued_cust,

    i.Issued_book_name,

    DATE_ADD(i.Issue_date, INTERVAL FLOOR(RAND() * 15) DAY),

    i.Isbn_book

FROM IssueStatus i;
```

Queries

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

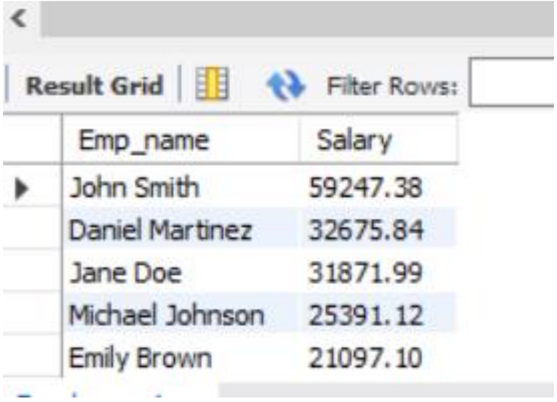
```
SELECT Book_title, Category, Rental_Price
FROM Books
WHERE Status = 'yes';
```



Book_title	Category	Rental_Price
1984	Dystopian	10.31
The Catcher in the Rye	Coming-of-Age	11.61
To Kill a Mockingbird	Classic	15.34
Pride and Prejudice	Romance	12.98
The Great Gatsby	Fiction	15.79

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

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



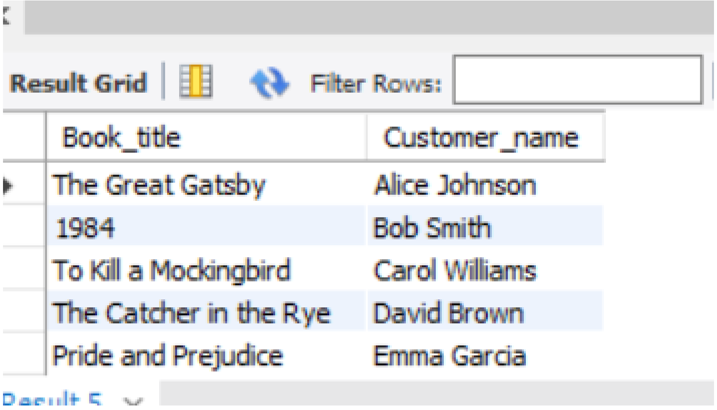
Emp_name	Salary
John Smith	59247.38
Daniel Martinez	32675.84
Jane Doe	31871.99
Michael Johnson	25391.12
Emily Brown	21097.10

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

```
SELECT b.Book_title, c.Customer_name

FROM IssueStatus i
```

JOIN Books b ON i.Isbn_book = b.ISBN

JOIN Customers c ON i.Issued_cust = c.Customer_Id;



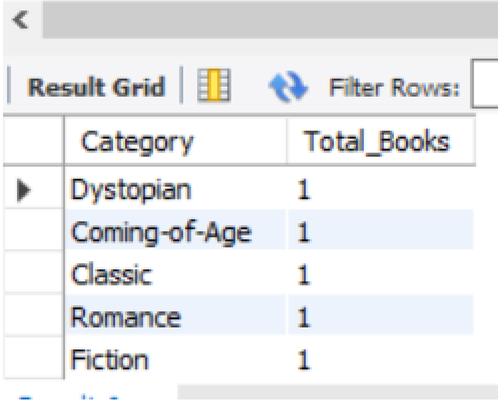
Book_title	Customer_name
The Great Gatsby	Alice Johnson
1984	Bob Smith
To Kill a Mockingbird	Carol Williams
The Catcher in the Rye	David Brown
Pride and Prejudice	Emma Garcia

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

SELECT Category, COUNT(*) AS Total_Books

FROM Books

GROUP BY Category;



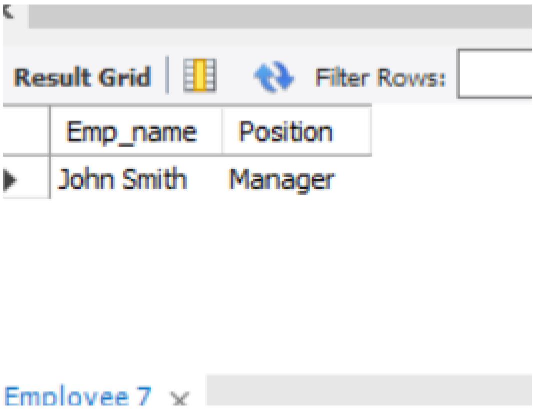
Category	Total_Books
Dystopian	1
Coming-of-Age	1
Classic	1
Romance	1
Fiction	1

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

SELECT Emp_name, Position

FROM Employee

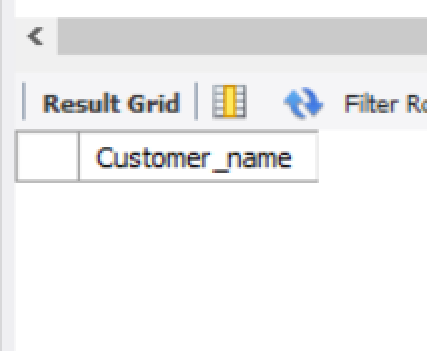
WHERE Salary > 50000;



Emp_name	Position
John Smith	Manager

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

```
SELECT Customer_name

FROM Customers c

WHERE Reg_date < '2022-01-01'

AND NOT EXISTS (

   SELECT 1

   FROM IssueStatus i

   WHERE i.Issued_cust = c.Customer_Id

);
```

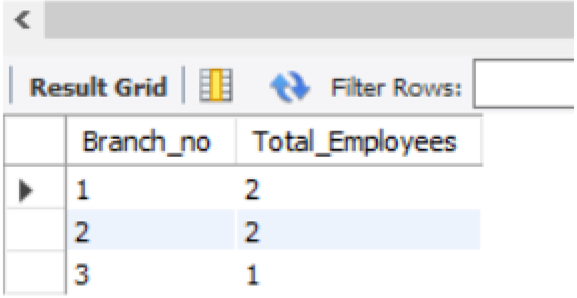


Customer_name

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

```
SELECT e.Branch_no, COUNT(*) AS Total_Employees

FROM Employee e
```

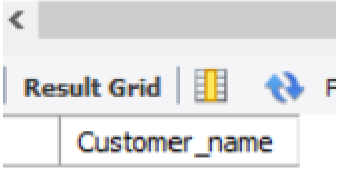
```
GROUP BY e.Branch_no;
```



Branch_no	Total_Employees
1	2
2	2
3	1

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

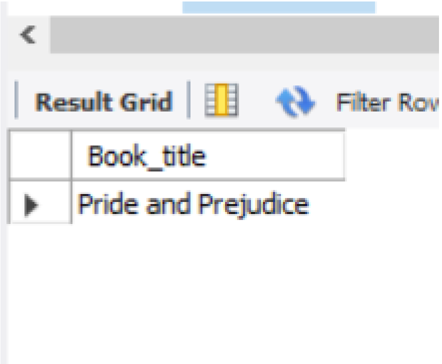
```
SELECT DISTINCT c.Customer_name

FROM Customers c

JOIN IssueStatus i ON c.Customer_Id = i.Issued_cust

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



Customer_name

9. Retrieve `Book_title` from the `Books` table containing "history".

```
SELECT Book_title

FROM Books

WHERE Category LIKE '%history%';
```



Book_title
Pride and Prejudice

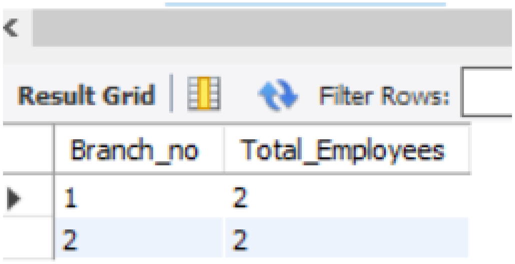
10. Retrieve the branch numbers along with the count of employees for branches having more than 1 employees.

SELECT e.Branch_no, COUNT(*) AS Total_Employees

FROM Employee e

GROUP BY e.Branch_no

HAVING COUNT(*) > 1;



Branch_no	Total_Employees
1	2
2	2

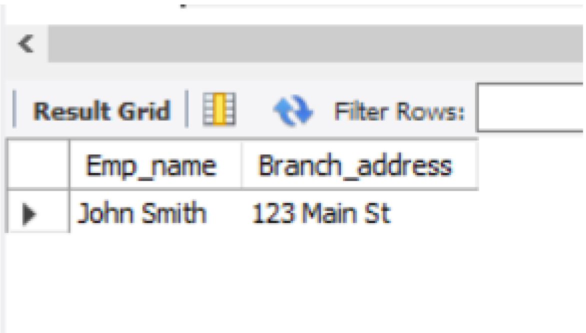
11. Retrieve the names of employees who manage branches and their respective branch addresses.

SELECT e.Emp_name, b.Branch_address

FROM Employee e

JOIN Branch b ON e.Branch_no = b.Branch_no

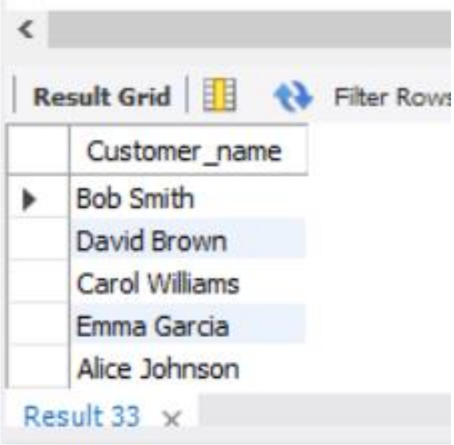
WHERE e.Position = 'Manager';



Emp_name	Branch_address
John Smith	123 Main St

12. Display the names of customers who have issued books with a rental price higher than Rs. 2.

```
SELECT DISTINCT c.Customer_name

FROM Customers c

JOIN IssueStatus i ON c.Customer_Id = i.Issued_cust

JOIN Books b ON i.Isbn_book = b.ISBN

WHERE b.Rental_Price > 25;
```



Customer_name
Bob Smith
David Brown
Carol Williams
Emma Garcia
Alice Johnson