

Library Management System

1. Branch

```
CREATE TABLE Branch (  
    Branch_no INT PRIMARY KEY,  
    Manager_Id INT,  
    Branch_address VARCHAR(255),  
    Contact_no VARCHAR(15)  
);
```

2. Employee

```
CREATE TABLE Employee (  
    Emp_Id INT PRIMARY KEY,  
    Emp_name VARCHAR(255),  
    Position VARCHAR(255),  
    Salary DECIMAL(10, 2),  
    Branch_no INT,  
    FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)  
);
```

3. Books

```
CREATE TABLE Books (  
    ISBN VARCHAR(13) PRIMARY KEY,  
    Book_title VARCHAR(255),  
    Category VARCHAR(100),  
    Rental_Price DECIMAL(10, 2),  
    Status VARCHAR(3),  
    Author VARCHAR(255),  
    Publisher VARCHAR(255)  
);
```

4. Customer

```
CREATE TABLE Customer (  
    Customer_Id INT PRIMARY KEY,  
    Customer_name VARCHAR(255),
```

```
Customer_address VARCHAR(255),  
Reg_date DATE  
);
```

5. IssueStatus

```
CREATE TABLE IssueStatus (  
    Issue_Id INT PRIMARY KEY,  
    Issued_cust INT,  
    Issued_book_name VARCHAR(255),  
    Issue_date DATE,  
    Isbn_book VARCHAR(13),  
    FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),  
    FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)  
);
```

6. ReturnStatus

```
CREATE TABLE ReturnStatus (  
    Return_Id INT PRIMARY KEY,  
    Return_cust INT,  
    Return_book_name VARCHAR(255),  
    Return_date DATE,  
    Isbn_book2 VARCHAR(13),  
    FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),  
    FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)  
);
```

INSERT QUERY

BRANCH

```
INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)  
VALUES  
(101, 1, '123 Main St, City A', '9876543210'),  
(102, 2, '456 Elm St, City B', '9876543222'),  
(103, 3, '789 Maple St, City C', '9876543233');
```

EMPLOYEE

```
INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no)
```

VALUES

```
(1, 'John Doe', 'Manager', 60000, 101),  
(2, 'Jane Smith', 'Manager', 65000, 102),  
(3, 'Bob Johnson', 'Manager', 55000, 103),  
(4, 'Alice Brown', 'Librarian', 40000, 101),  
(5, 'Charlie Davis', 'Assistant Librarian', 35000, 101),  
(6, 'Eve Wilson', 'Librarian', 45000, 102),  
(7, 'Frank Miller', 'Assistant Librarian', 37000, 102),  
(8, 'Grace Lee', 'Librarian', 42000, 103);
```

BOOK

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

```
('A1234', 'The Great Gatsby', 'Fiction', 20, 'yes', 'F. Scott Fitzgerald', 'Scribner'),  
( 'B2345', 'The Art of War', 'History', 30, 'no', 'Sun Tzu', 'Penguin'),  
( 'C3456', 'Clean Code', 'Technology', 25, 'yes', 'Robert C. Martin', 'Prentice Hall'),  
( 'D4567', 'The History of Rome', 'History', 35, 'yes', 'Livy', 'Oxford'),  
( 'E5678', 'Python Crash Course', 'Technology', 40, 'no', 'Eric Matthes', 'No Starch Press');
```

CUSTOMER

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

```
(1, 'Michael Scott', 'Scranton, PA', '2021-12-15'),  
(2, 'Pam Beesly', 'Scranton, PA', '2022-01-05'),  
(3, 'Jim Halpert', 'Scranton, PA', '2023-05-10'),  
(4, 'Dwight Schrute', 'Scranton, PA', '2021-11-22'),  
(5, 'Stanley Hudson', 'Scranton, PA', '2023-06-05');
```

ISSUESTATUS

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

```
(1, 1, 'The Great Gatsby', '2023-06-01', 'A1234'),  
(2, 3, 'The Art of War', '2023-06-15', 'B2345'),  
(3, 5, 'The History of Rome', '2023-06-10', 'D4567');
```

RETURNSTATUS

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

```
VALUES
```

```
(1, 1, 'The Great Gatsby', '2023-07-01', 'A1234'),  
(2, 3, 'The Art of War', '2023-07-20', 'B2345');
```

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

The screenshot shows the SQL Developer interface for a local instance 3306. The 'Schemas' pane on the left lists various databases, with 'test_db' selected. The 'Query' pane shows the following SQL query:

```
1 • SELECT Book_title, Category, Rental_Price
2 FROM Books
3 WHERE Status = 'yes';
4
```

The 'Result Grid' at the bottom displays the results of the query:

#	Book_title	Category	Rental_Price
1	The Great Gatsby	Fiction	20.00
2	Clean Code	Technology	25.00
3	The History of Rome	History	35.00

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

The screenshot shows the SQL Developer interface for a local instance 3306. The 'Schemas' pane on the left lists various databases, with 'test_db' selected. The 'Query' pane shows the following SQL query:

```
1 • SELECT Emp_name, Salary
2 FROM Employee
3 ORDER BY Salary DESC;
4
```

The 'Result Grid' at the bottom displays the results of the query:

#	Emp_name	Salary
1	Jane Smith	65000.00
2	John Doe	60000.00
3	Bob Johnson	55000.00
4	Eve Wilson	45000.00
5	Grace Lee	42000.00
6	Alice Brown	40000.00
7	Frank Miller	37000.00
8	Charlie Davis	35000.00

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

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' panel lists various databases, with 'test_db' selected. The main area displays a SQL query:

```
1 • SELECT B.Book_title, C.Customer_name
2 FROM IssueStatus I
3 JOIN Books B ON I.Isbn_book = B.ISBN
4 JOIN Customer C ON I.Issued_cust = C.Customer_Id;
```

Below the query, the 'Result Grid' shows the following data:

#	Book_title	Customer_name
1	The Great Gatsby	Michael Scott
2	The Art of War	Jim Halpert
3	The History of Rome	Stanley Hudson

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

The screenshot shows the same database management tool interface. The SQL query is:

```
1 • SELECT Category, COUNT(*)
2 FROM Books
3 GROUP BY Category;
```

The 'Result Grid' displays the following data:

#	Category	COUNT(*)
1	Fiction	1
2	History	2
3	Technology	2

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

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane is expanded to show the 'test_db' database. The 'Tables' folder is selected. The main pane displays a SQL query:

```
1 • SELECT Emp_name, Position
2 FROM Employee
3 WHERE Salary > 50000;
4
```

Below the query, the 'Result Grid' shows the results of the query:

#	Emp_name	Position
1	John Doe	Manager
2	Jane Smith	Manager
3	Bob Johnson	Manager

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

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane is expanded to show the 'test_db' database. The 'Tables' folder is selected. The main pane displays a SQL query:

```
1 • SELECT C.Customer_name
2 FROM Customer C
3 LEFT JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust
4 WHERE C.Reg_date < '2022-01-01'
5 AND I.Issue_Id IS NULL;
6
```

Below the query, the 'Result Grid' shows the results of the query:

#	Customer_name
1	Dwight Schrute

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

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane is expanded to show the 'test_db' database, with 'Tables' selected. The main query editor displays the following SQL code:

```
1 • SELECT Branch_no, COUNT(*)
2 FROM Employee
3 GROUP BY Branch_no;
```

The 'Result Grid' pane at the bottom shows the results of the query:

#	Branch_no	COUNT(*)
1	101	3
2	102	3
3	103	2

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

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane is expanded to show the 'test_db' database, with 'Tables' selected. The main query editor displays the following SQL code:

```
1 • SELECT C.Customer_name
2 FROM IssueStatus I
3 JOIN Customer C ON I.Issued_cust = C.Customer_Id
4 WHERE I.Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
```

The 'Result Grid' pane at the bottom shows the results of the query:

#	Customer_name
1	Michael Scott
2	Jim Halpert
3	Stanley Hudson

9. Retrieve book titles from the book table containing 'history':

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' tree lists various databases, with 'test_db' expanded to show 'Tables' and 'Views'. The main area displays a SQL query:

```
1 • SELECT Book_title
2 FROM Books
3 WHERE Book_title LIKE '%history%';
4
```

Below the query editor, the 'Result Grid' shows a single row of results:

#	Book_title
1	The History of Rome

The interface also includes a 'Filter Rows' section and an 'Export' button.

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

The screenshot shows the same database management tool interface. The SQL query editor now contains:

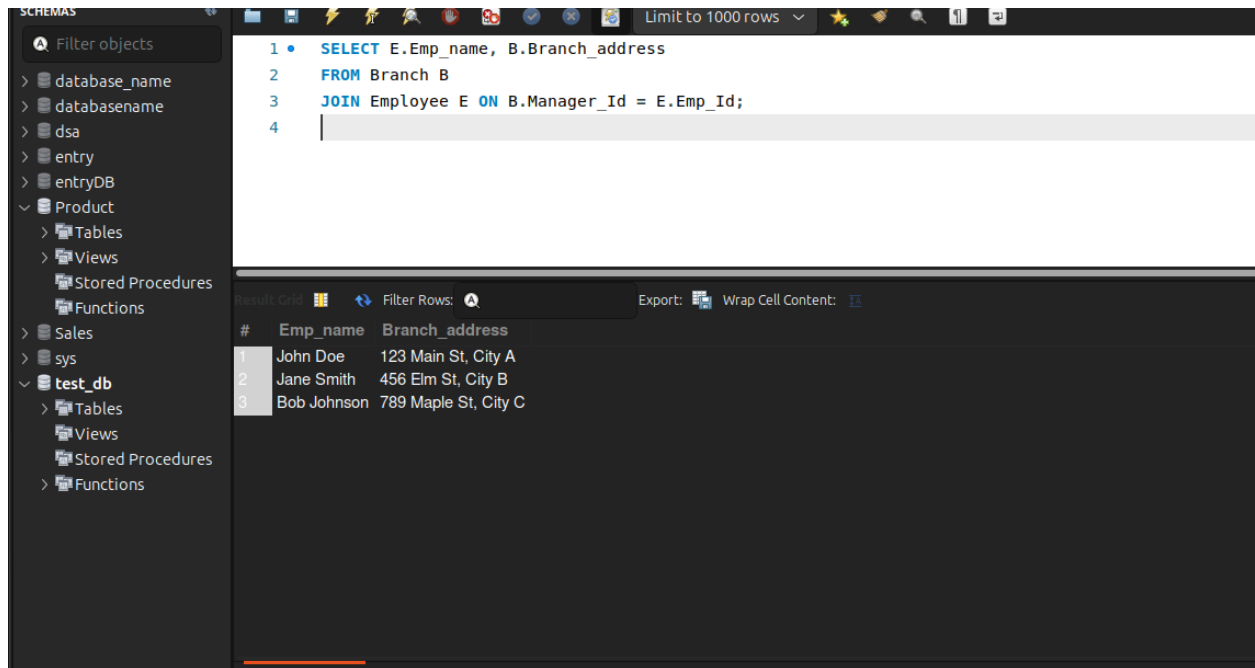
```
1 • SELECT Branch_no, COUNT(*)
2 FROM Employee
3 GROUP BY Branch_no
4 HAVING COUNT(*) > 5;
5
```

The 'Result Grid' is currently empty, showing only the column headers:

#	Branch_no	COUNT(*)
---	-----------	----------

The interface also includes a 'Filter Rows' section and an 'Export' button.

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



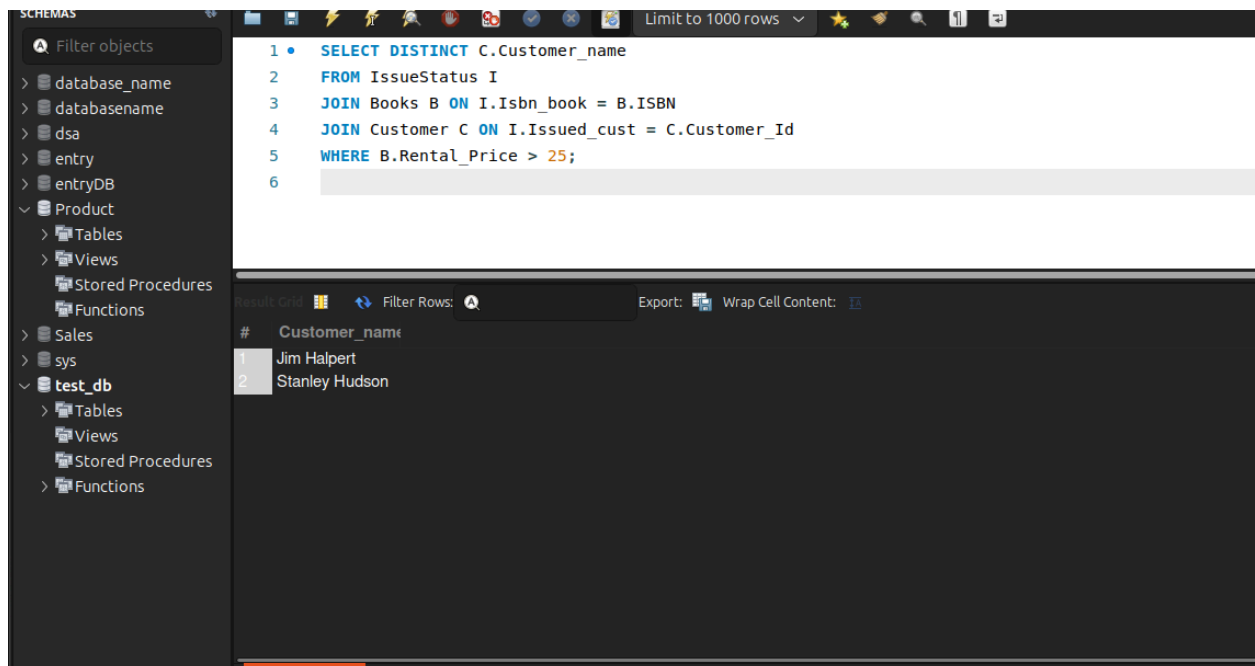
The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists various databases, with 'test_db' selected. The main area displays a SQL query:

```
1 • SELECT E.Emp_name, B.Branch_address
2 FROM Branch B
3 JOIN Employee E ON B.Manager_Id = E.Emp_Id;
4
```

Below the query, the 'Result Grid' shows the following data:

#	Emp_name	Branch_address
1	John Doe	123 Main St, City A
2	Jane Smith	456 Elm St, City B
3	Bob Johnson	789 Maple St, City C

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



The screenshot shows the same database management tool interface. The SQL query is:

```
1 • SELECT DISTINCT C.Customer_name
2 FROM IssueStatus I
3 JOIN Books B ON I.Isbn_book = B.ISBN
4 JOIN Customer C ON I.Issued_cust = C.Customer_Id
5 WHERE B.Rental_Price > 25;
6
```

The 'Result Grid' displays the following data:

#	Customer_name
1	Jim Halpert
2	Stanley Hudson

