# **Library Management System**

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 – 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.

Create a database named library and following TABLES in the database:

* The Library Management System is designed to efficiently manage and maintain the details of books, employees, branches, and customers. The system includes tables for various entities, each connected through foreign keys to ensure data integrity and facilitate complex queries. This system helps librarians manage books' availability, track issued and returned books and keep records of employees and branches.
* Creating a database Library.
  + Description: This database will contain all the details of the library in the tables created.
  + Query:

CREATE DATABASE Library;

GO

USE Library;

GO

1. Branch table-> Branch\_no - Set as PRIMARY KEY, Manager\_Id, Branch\_address, Contact\_no.
   1. Description: The Branch table holds information about different branches of the library. Each branch has a unique branch number and is managed by an employee.
   2. Query:

CREATE TABLE Branch (

Branch\_no INT PRIMARY KEY,

Manager\_Id INT,

Branch\_address VARCHAR(255),

Contact\_no VARCHAR(15)

);

INSERT INTO Branch (Branch\_no, Manager\_Id, Branch\_address, Contact\_no) VALUES

(1, 101, 'Mumbai', '02212345678'),

(2, 102, 'Delhi', '01112345678'),

(3, 103, 'Kolkata', '03312345678'),

(4, 104, 'Chennai', '04412345678'),

(5, 105, 'Bangalore', '08012345678'),

(6, 106, 'Hyderabad', '04012345678'),

(7, 107, 'Ahmedabad', '07912345678'),

(8, 108, 'Pune', '02012345678'),

(9, 109, 'Jaipur', '014112345678'),

(10, 110, 'Lucknow', '052212345678');

SELECT \* FROM Branch;

A screenshot of a computer program

Description automatically generated

1. Employee table-> Emp\_Id – Set as PRIMARY KEY, Emp\_name, Position, Salary,

Branch\_no - Set as FOREIGN KEY and it refer Branch\_no in Branch table.

* 1. Description: The Employee table contains details of all employees working in the library, including their positions and the branches they are associated with.
  2. Query:

CREATE TABLE Employee (

Emp\_id INT PRIMARY KEY,

Emp\_name VARCHAR(100),

Position VARCHAR(100),

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

(101, 'Ramesh Kumar', 'Manager', 60000, 1),

(102, 'Suresh Gupta', 'Manager', 62000, 2),

(103, 'Mahesh Jaiswal', 'Manager', 63000, 3),

(104, 'Rajesh Singh', 'Manager', 64000, 4),

(105, 'Naresh Singh', 'Manager', 65000, 5),

(106, 'Dinesh Gupta', 'Assistant Manager', 55000, 1),

(107, 'Mukesh Kumhar', 'Assistant Manager', 56000, 2),

(108, 'Lokesh Rahul', 'Assistant Manager', 57000, 3),

(109, 'Jignesh Reddy', 'Clerk', 30000, 1),

(110, 'Harish Chaudhary', 'Clerk', 31000, 2),

(111, 'Prakash Raj', 'Clerk', 32000, 3),

(112, 'Amit Trivedi', 'Clerk', 33000, 4),

(113, 'Sumit Shah', 'Clerk', 34000, 5),

(114, 'Pawan Kumar', 'BPO', 40000, 1),

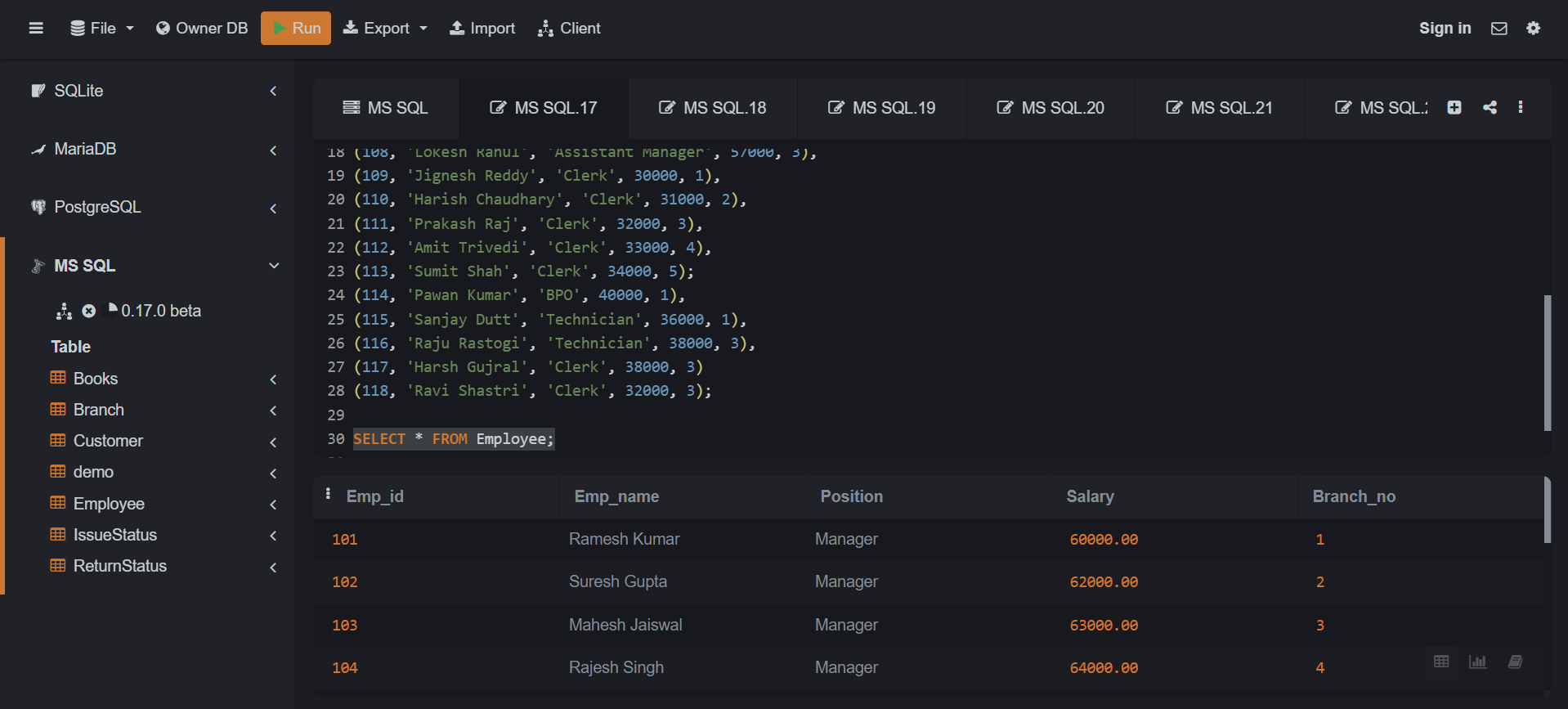
(115, 'Sanjay Dutt', 'Technician', 36000, 1),

(116, 'Raju Rastogi', 'Technician', 38000, 3),

(117, 'Harsh Gujral', 'Clerk', 38000, 3)

(118, 'Ravi Shastri', 'Clerk', 32000, 3);

SELECT \* from Employee;



1. Books table-> ISBN - Set as PRIMARY KEY, Book\_title, Category, Rental\_Price, Status, [Give yes if book available and no if book not available], Author, Publisher.
   1. Description: The Books table keeps track of all books in the library, including their status (available or not), rental price, and other bibliographic details.
   2. Query:

CREATE TABLE Books (

ISBN VARCHAR(20) PRIMARY KEY,

Book\_title VARCHAR(255),

Category VARCHAR(100),

Rental\_Price DECIMAL(10, 2),

Status VARCHAR(3) CHECK (Status IN ('yes', 'no')),

Author VARCHAR(100),

Publisher VARCHAR(100)

);

INSERT INTO Books (ISBN, Book\_title, Category, Rental\_Price, Status, Author, Publisher) VALUES

('978-3-16-148410-0', 'The Great Gatsby', 'Fiction', 20.00, 'yes', 'F. Scott Fitzgerald', 'Scribner'),

('978-0-14-044926-6', 'Crime and Punishment', 'Fiction', 25.00, 'no', 'Fyodor Dostoevsky', 'Penguin Classics'),

('978-1-56619-909-4', 'War and Peace', 'History', 30.00, 'yes', 'Leo Tolstoy', 'Modern Library'),

('978-0-307-74176-3', 'The Wright Brothers', 'Biography', 22.00, 'yes', 'David McCullough', 'Simon & Schuster'),

('978-1-250-15126-0', 'A Brief History of Time', 'Science', 28.00, 'no', 'Stephen Hawking', 'Bantam Books'),

('978-0-06-231609-7', 'Sapiens', 'Non-Fiction', 27.00, 'yes', 'Yuval Noah Harari', 'Harper'),

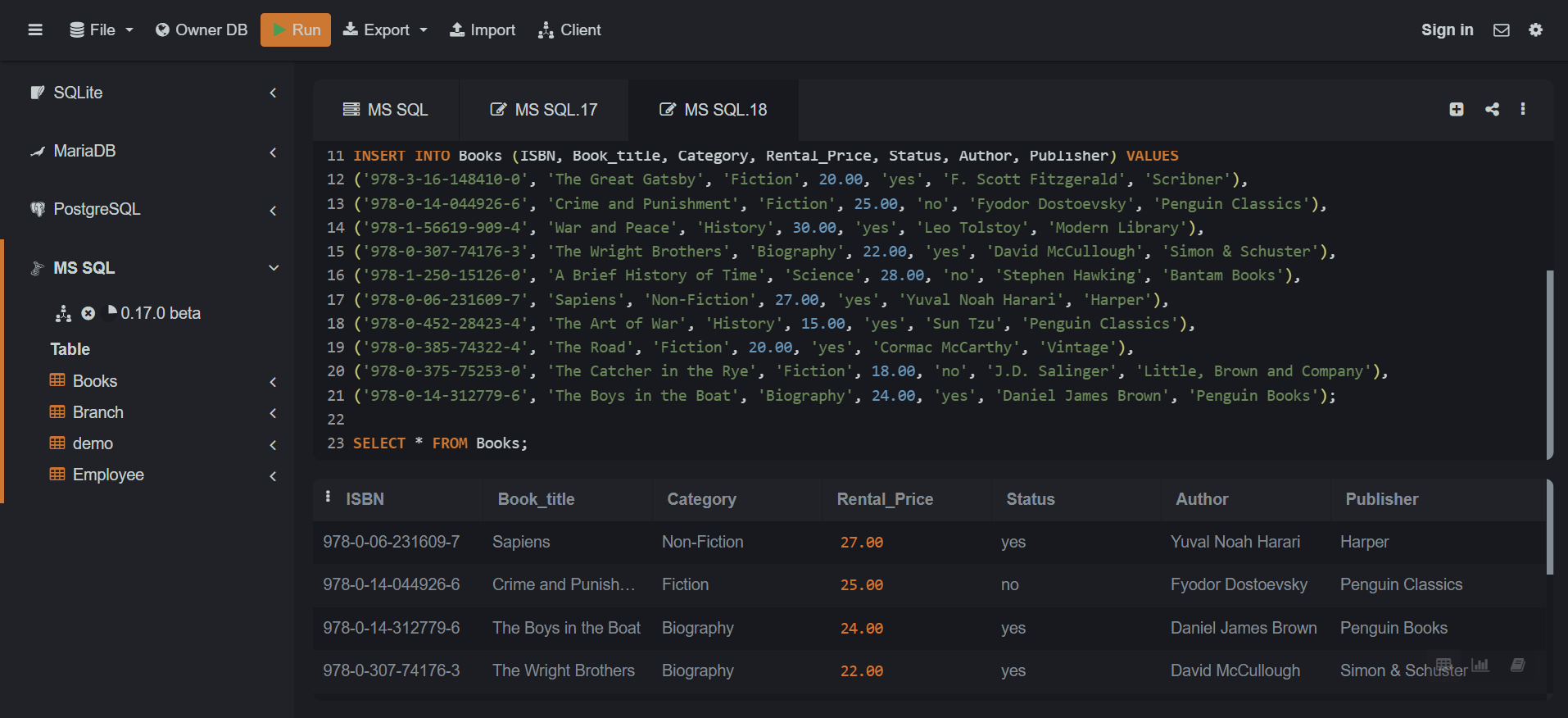
('978-0-452-28423-4', 'The Art of War', 'History', 15.00, 'yes', 'Sun Tzu', 'Penguin Classics'),

('978-0-385-74322-4', 'The Road', 'Fiction', 20.00, 'yes', 'Cormac McCarthy', 'Vintage'),

('978-0-375-75253-0', 'The Catcher in the Rye', 'Fiction', 18.00, 'no', 'J.D. Salinger', 'Little, Brown and Company'),

('978-0-14-312779-6', 'The Boys in the Boat', 'Biography', 24.00, 'yes', 'Daniel James Brown', 'Penguin Books');

SELECT \* FROM Books;



1. Customer table-> Customer\_Id - Set as PRIMARY KEY, Customer\_name, Customer\_address, Reg\_date.
   1. Description: The Customer table stores information about library customers, including their registration details and addresses.
   2. Query:

CREATE TABLE Customer (

Customer\_Id INT PRIMARY KEY,

Customer\_name VARCHAR(100),

Customer\_address VARCHAR(255),

Reg\_date DATE

);

INSERT INTO Customer (Customer\_Id, Customer\_name, Customer\_address, Reg\_date) VALUES

(301, 'Anil Sharma', 'Mumbai-Maharashtra', '2021-12-15'),

(302, 'Sunil Kumar', 'Delhi-Delhi', '2023-01-10'),

(303, 'Vinil Verma', 'Kolkata-West Bengal', '2020-05-20'),

(304, 'Rohit Gupta', 'Chennai-Tamil Nadu', '2021-11-25'),

(305, 'Manish Patel', 'Bangalore-Karnataka', '2022-08-14'),

(306, 'Rajeev Singh', 'Hyderabad-Telangana', '2023-02-10'),

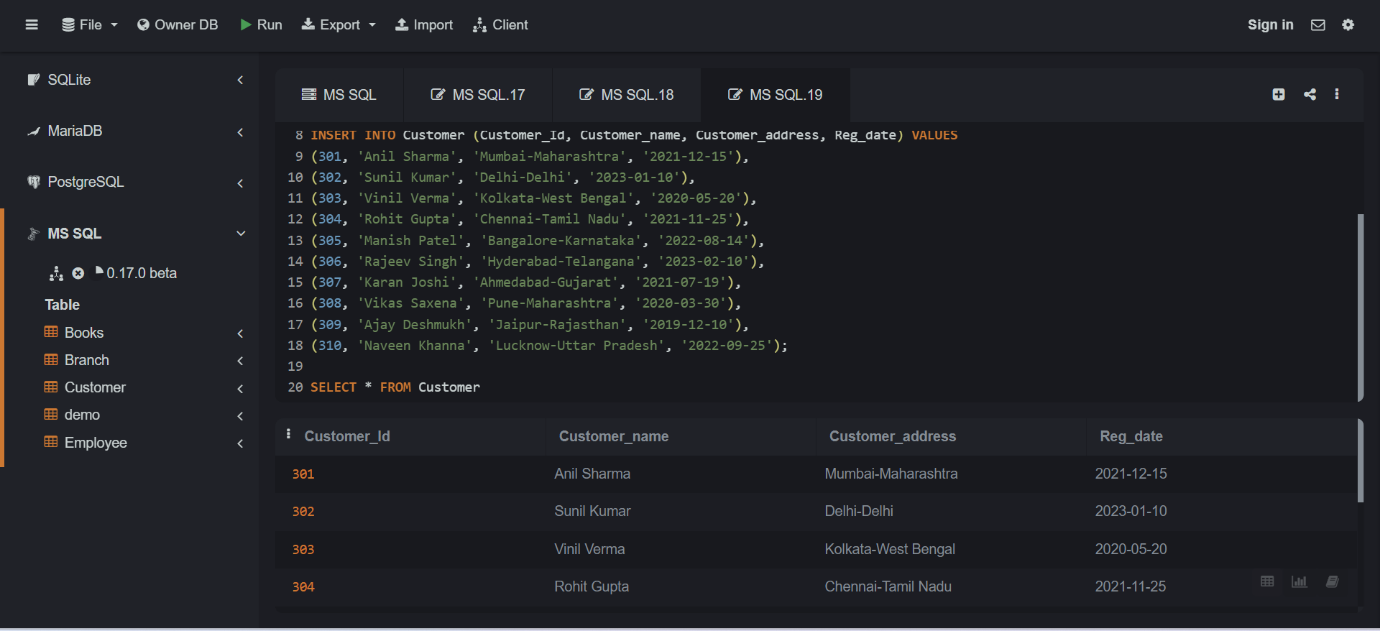
(307, 'Karan Joshi', 'Ahmedabad-Gujarat', '2021-07-19'),

(308, 'Vikas Saxena', 'Pune-Maharashtra', '2020-03-30'),

(309, 'Ajay Deshmukh', 'Jaipur-Rajasthan', '2019-12-10'),

(310, 'Naveen Khanna', 'Lucknow-Uttar Pradesh', '2022-09-25');

SELECT \* FROM Customer;



1. IssueStatus table-> 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.
   1. Description: The IssueStatus table records the details of books issued to customers, linking customers and books with issue dates.
   2. Query:

CREATE TABLE IssueStatus (

Issue\_Id INT PRIMARY KEY,

Issued\_cust INT,

Issued\_book\_name VARCHAR(255),

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

(401, 301, 'The Great Gatsby', '2023-06-10', '978-3-16-148410-0'),

(402, 302, 'War and Peace', '2023-06-15', '978-1-56619-909-4'),

(403, 303, 'The Wright Brothers', '2023-06-18', '978-0-307-74176-3'),

(404, 304, 'Sapiens', '2023-06-20', '978-0-06-231609-7'),

(405, 305, 'The Road', '2023-06-22', '978-0-385-74322-4'),

(406, 306, 'The Boys in the Boat', '2023-06-25', '978-0-14-312779-6');

SELECT \* from IssueStatus;

A screenshot of a computer program

Description automatically generated

1. ReturnStatus table-> 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.
   1. Description: customers and return dates.
   2. Query:

CREATE TABLE ReturnStatus (

Return\_Id INT PRIMARY KEY,

Return\_cust INT,

Return\_book\_name VARCHAR(255),

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) VALUES

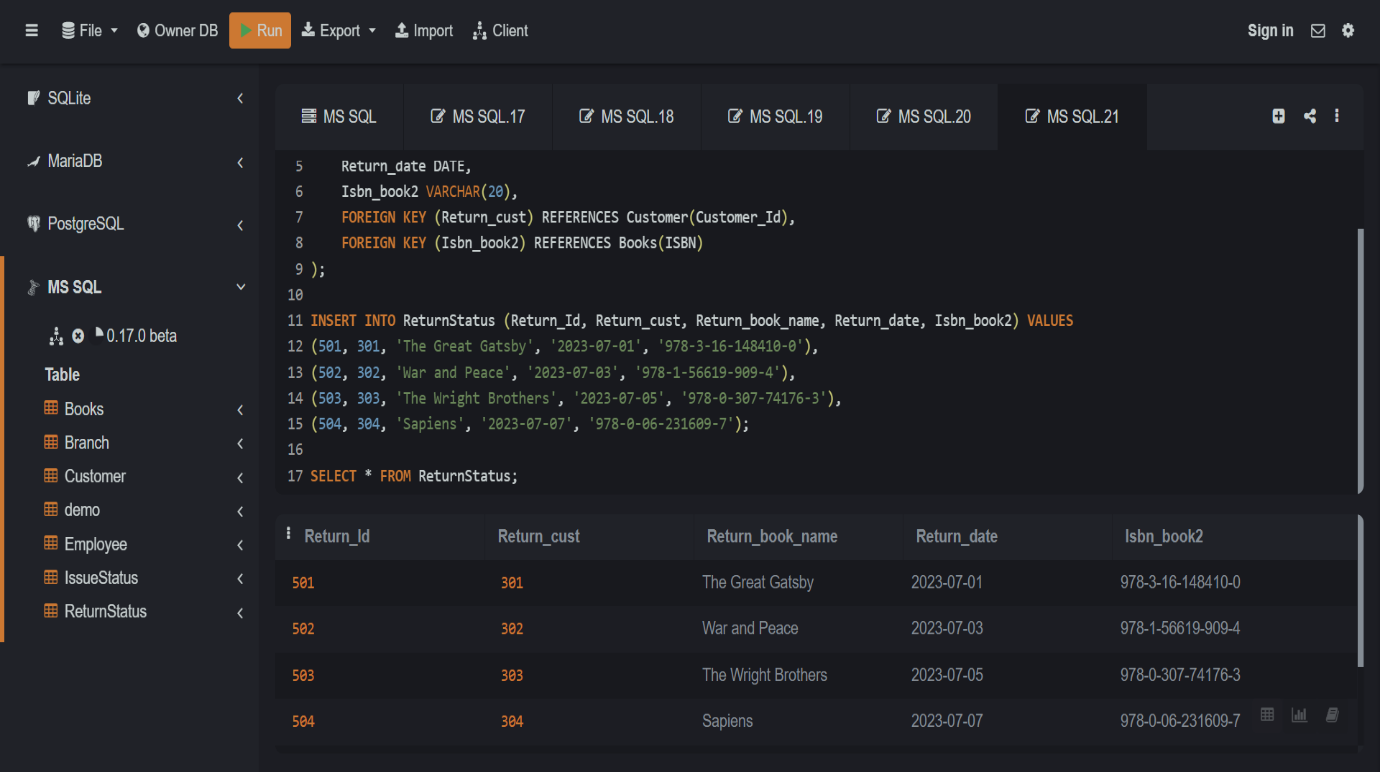
(501, 301, 'The Great Gatsby', '2023-07-01', '978-3-16-148410-0'),

(502, 302, 'War and Peace', '2023-07-03', '978-1-56619-909-4'),

(503, 303, 'The Wright Brothers', '2023-07-05', '978-0-307-74176-3'),

(504, 304, 'Sapiens', '2023-07-07', '978-0-06-231609-7');

SELECT \* from ReturnStatus;



**QUERY:**

1. Retrieve the book title, category, and rental price of all available books.
   1. Description: This query fetches the title, category, and rental price of books that are currently available for borrowing.
   2. Advantages: Helps librarians quickly identify books that are available for issue, aiding in customer service and inventory management.
   3. Query:

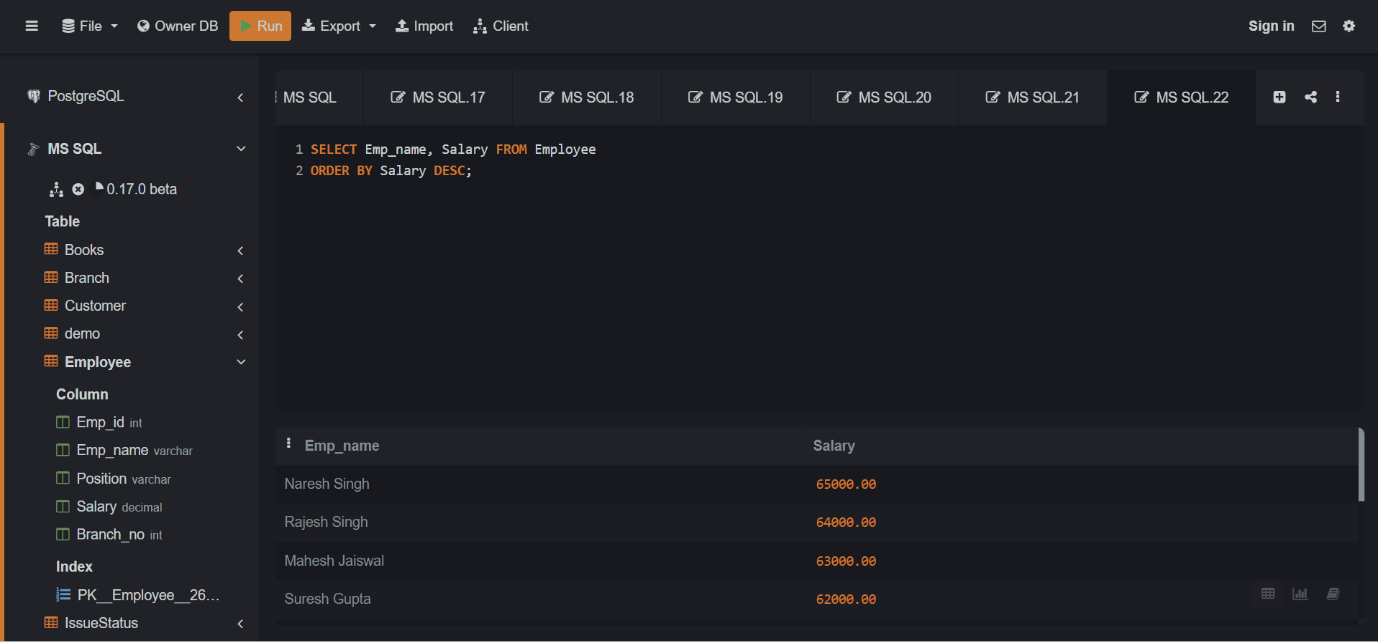
SELECT Book\_title, Category, Rental\_price FROM Books

A screenshot of a computer

Description automatically generatedwhere Status='yes';

1. List the employee names and their respective salaries in descending order of salary.
   1. Description: This query lists employees and their salaries, sorted from highest to lowest salary.
   2. Advantages: Useful for management to review payroll and make informed decisions regarding salary distributions and raises.
   3. Query:

SELECT Emp\_name, Salary FROM Employee

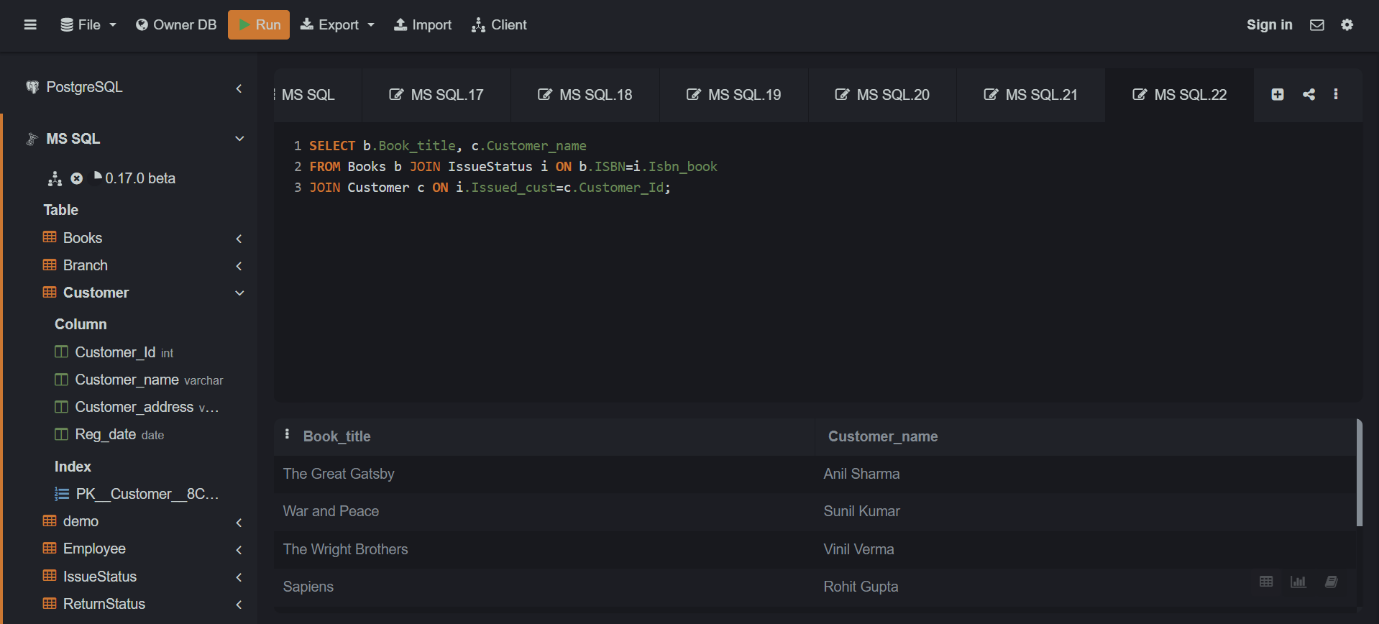
ORDER by Salary DESC;

1. Retrieve the book titles and the corresponding customers who have issued those books.
   1. Description: Fetches the titles of books along with the names of customers who have issued them.
   2. Advantages: Helps in tracking which customers have which books, useful for follow-ups and overdue notifications.
   3. Query:

SELECT b.Book\_title, c.Customer\_name

from Books b JOIN IssueStatus i on b.ISBN=i.Isbn\_book

JOIN Customer c on i.Issued\_cust=c.Customer\_Id;



1. Display the total count of books in each category.
   1. Description: Provides a count of books grouped by their categories.
   2. Advantages: Assists in understanding the distribution of books across different genres, helping in collection development and acquisition strategies.
   3. Query:

SELECT Category, count(\*) AS Total\_Books FROM Books

A screenshot of a computer

Description automatically generatedGROUP BY Category;

1. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
   1. Description: Lists the names and positions of employees earning more than Rs.50,000.
   2. Advantages: Useful for budget planning and identifying high-paid staff for potential leadership roles or salary reviews.
   3. Query:

SELECT Emp\_name, Position FROM Employee

A screenshot of a computer

Description automatically generatedWHERE Salary>50000;

1. List the customer names who registered before 2022-01-01 and have not issued any books yet.
   1. Description: Identifies customers who registered before a specific date and have not borrowed any books.
   2. Advantages: H elps in understanding customer engagement and can be used to encourage inactive members to utilize library services.
   3. Query:

SELECT Customer\_name FROM Customer

A screenshot of a computer program

Description automatically generatedWHERE Reg\_date<'2022-01-01' and Customer\_id NOT IN (SELECT Issued\_cust from IssueStatus);

1. Display the branch numbers and the total count of employees in each branch.
   1. Description: Provides the number of employees working in each branch.
   2. Advantages: Useful for branch managers to ensure adequate staffing and for planning resource allocation.
   3. Query:

SELECT Branch\_no, COUNT(Emp\_id) AS Total\_Employee FROM Employee

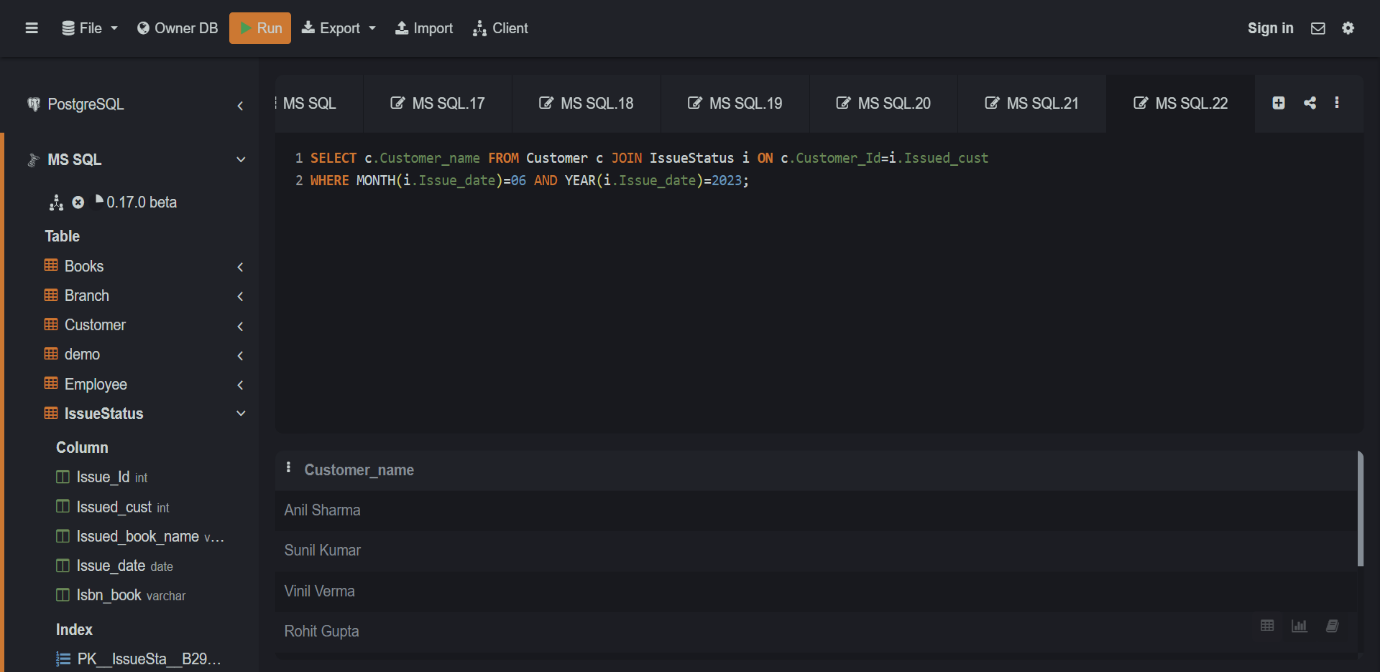
A screenshot of a computer

Description automatically generatedGROUP BY Branch\_no;

1. Display the names of customers who have issued books in the month of June 2023.
   1. Description: Lists customers who borrowed books during June 2023.
   2. Advantages: Helps in analyzing borrowing trends and identifying peak borrowing periods for better service management.
   3. Query:

SELECT c.Customer\_name FROM Customer c JOIN IssueStatus i ON c.Customer\_Id=i.Issued\_cust

WHERE MONTH(i.Issue\_date)=06 AND YEAR(i.Issue\_date)=2023;



1. Retrieve book\_title from book table containing history.
   1. Description: Fetches titles of books that have 'history' in their title.
   2. Advantages: Useful for users or staff searching for history-related books, aiding in quick information retrieval.
   3. Query:

SELECT Book\_title FROM Books

A screenshot of a computer

Description automatically generatedWHERE Category='History';

1. Retrieve the branch numbers along with the count of employees for branches having more than 5 employees.
   1. Description: Lists branches with more than five employees along with the number of employees.
   2. Advantages: Helps in identifying larger branches which may need more resources or could be restructured for efficiency.
   3. Query:

SELECT Branch\_no, COUNT(Emp\_id) AS Total\_Employee FROM Employee

GROUP BY Branch\_no

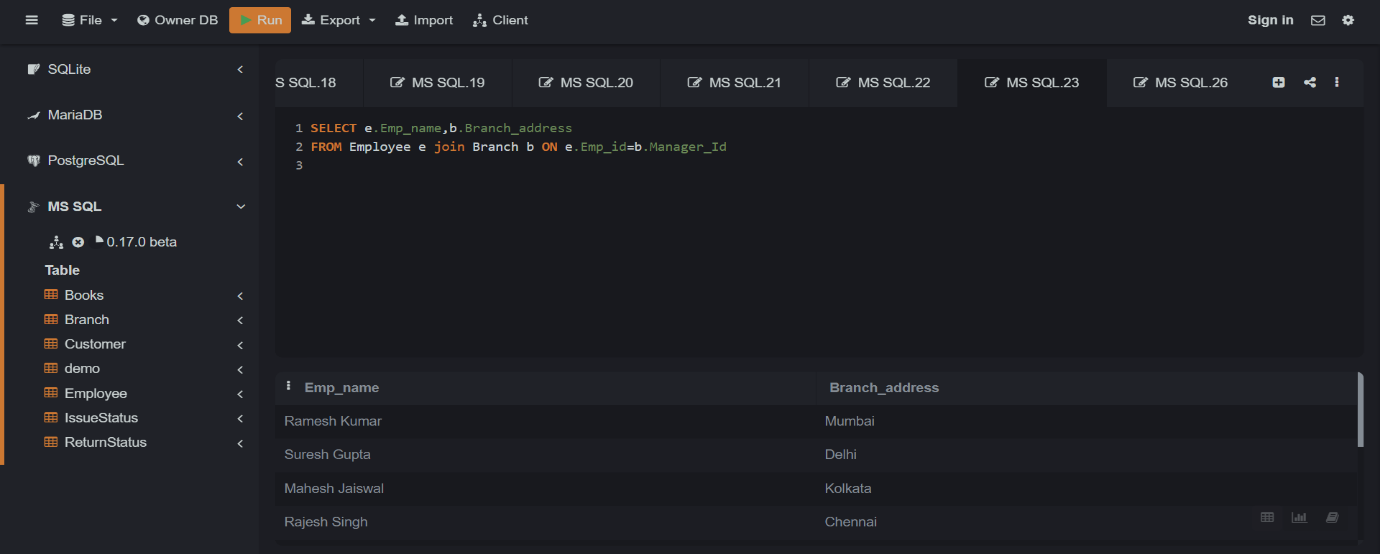
HAVING COUNT(Emp\_id)>5;

A screenshot of a computer

Description automatically generated

1. Retrieve the names of employees who manage branches and their respective branch addresses.
   1. Description: Fetches the names of employees managing branches and the addresses of those branches.
   2. Advantages: Useful for top management to quickly identify branch managers and their locations for coordination and decision-making.
   3. Query:

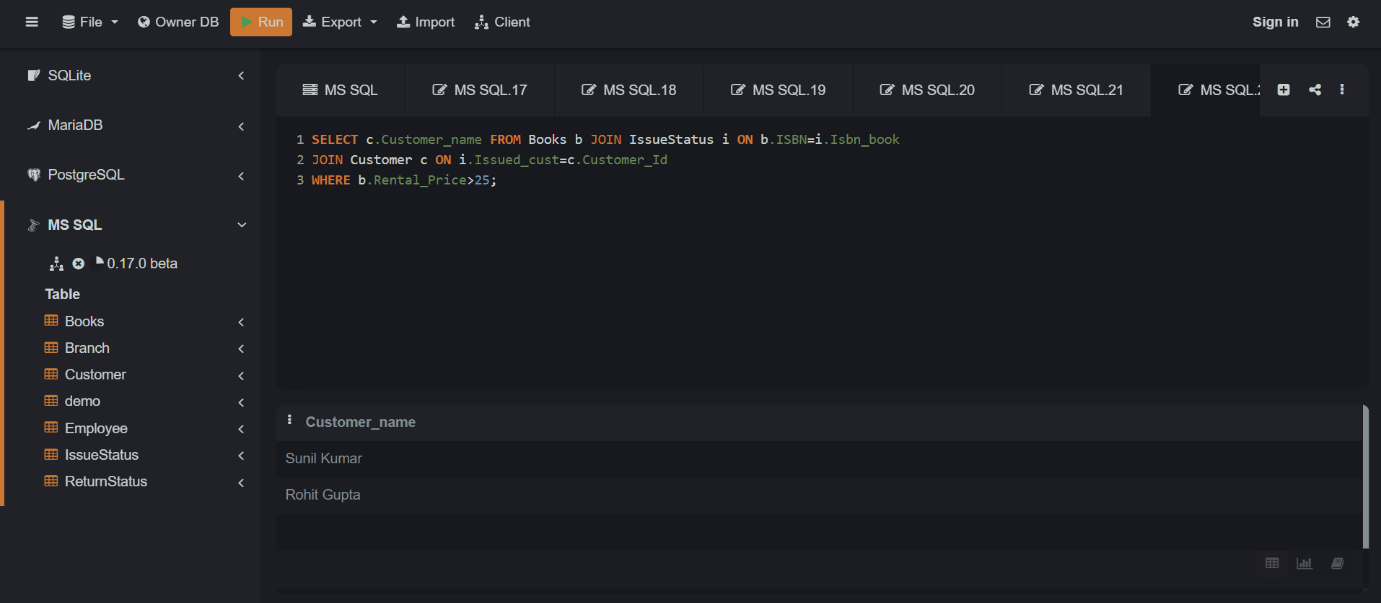
SELECT e.Emp\_name,b.Branch\_address

FROM Employee e JOIN Branch b ON e.Emp\_id=b.Manager\_Id;

1. Display the names of customers who have issued books with a rental price higher than Rs. 25.
   1. Description: Lists customers who have borrowed books that cost more than Rs. 25 to rent.
   2. Advantages: Helps in identifying customers who are willing to spend more, useful for targeted marketing and service enhancements.
   3. Query:

SELECT c.Customer\_name FROM Books b JOIN IssueStatus i on b.ISBN=i.Isbn\_book

JOIN Customer c on i.Issued\_cust=c.Customer\_Id

WHERE b.Rental\_Price>25;