***Library Management System***

**1 .** 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;

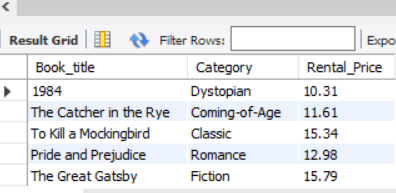
**Queries**

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

SELECT Book\_title, Category, Rental\_Price

FROM Books

WHERE Status = 'yes';

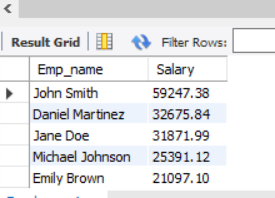


1. **List the employee names and their respective salaries in descending order of salary.**

SELECT Emp\_name, Salary

FROM Employee

ORDER BY Salary DESC;



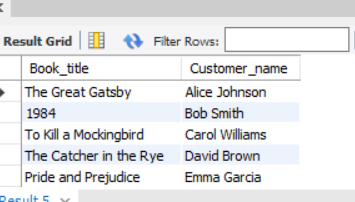
**3. Retrieve the book titles and the corresponding customers who have issued those books**

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;

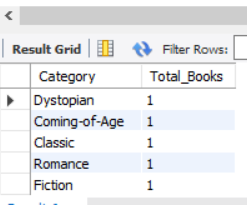
****

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

SELECT Category, COUNT(\*) AS Total\_Books

FROM Books

GROUP BY Category;

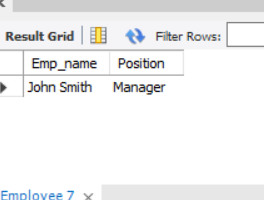


**5. Retrieve the employee names and their positions for the employees whose salaries are above Rs. 50,000.**

SELECT Emp\_name, Position

FROM Employee

WHERE Salary > 50000;



**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'

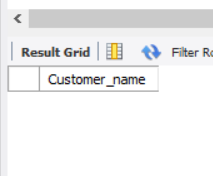
AND NOT EXISTS (

SELECT 1

FROM IssueStatus i

WHERE i.Issued\_cust = c.Customer\_Id

);

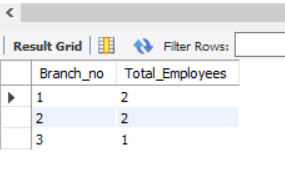


**7. Display the branch numbers and the total count of employees in each branch.**

SELECT e.Branch\_no, COUNT(\*) AS Total\_Employees

FROM Employee e

GROUP BY e.Branch\_no;



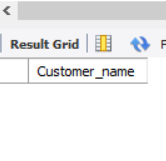
**8.** **Display the names of customers who have issued books in the month of June 2023.**

**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;**

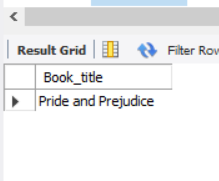
****

**9.** **Retrieve Book\_title from the Books table containing "history".**

SELECT Book\_title

FROM Books

WHERE Category LIKE '%history%';

****

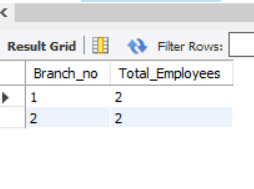
**10.** **Retrieve the branch numbers along with the count of employees for branches having more than 1 employees.**

SELECT e.Branch\_no, COUNT(\*) AS Total\_Employees

FROM Employee e

GROUP BY e.Branch\_no

HAVING COUNT(\*) > 1;

****

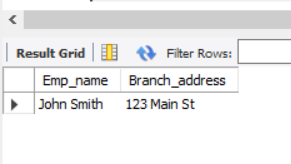
**11.** **Retrieve the names of employees who manage branches and their respective branch addresses.**

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

FROM Employee e

JOIN Branch b ON e.Branch\_no = b.Branch\_no

WHERE e.Position = 'Manager';



**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;

