

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 and insert 10 rows into respective 6 tables

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 database Library;
```

```
use Library;
```

```
Create table Branch(
```

```
Branch_No int primary key,
```

```
Manager_Id int,
```

```
Branch_Address varchar(255),
```

```
Contact_No varchar(50)
```

```
);
```

```
INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
```

```
VALUES
```

```
(1, 201, '123 Main St, Cityville', '555-1234'),
```

```
(2, 203, '456 Elm St, Townsville', '555-5678'),
```

(3, 205, '789 Oak St, Villageville', '555-9012'),
(4, 207, '321 Pine St, Hamletville', '555-3456'),
(5, 209, '654 Maple St, Boroughville', '555-7890'),
(6, 211, '987 Birch St, Suburbville', '555-2345'),
(7, 213, '159 Cedar St, Metropolis', '555-6789'),
(8, 215, '753 Walnut St, Countryside', '555-0123'),
(9, 217, '852 Cherry St, Urbanville', '555-4567'),
(10, 219, '951 Spruce St, Ruralville', '555-8901');

Select* from branch;

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)

);

Desc Employee;

Select* from Employee;

Insert into Employee (Emp_Id, Emp_Name, Position, Salary, Branch_No)

Values

(201, 'Alice Johnson', 'Librarian', 40000.00, 1),
(202, 'Bob Smith', 'Assistant Librarian', 55000.00, 1),
(203, 'Carol Davis', 'Librarian', 52000.00, 2),
(204, 'David Martin', 'Library Technician', 35000.00, 2),
(205, 'Eva Brown', 'Librarian', 55000.00, 3),
(206, 'Frank Wilson', 'Library Assistant', 53000.00, 3),
(207, 'Grace Lee', 'Librarian', 57000.00, 4),
(208, 'Hank Taylor', 'Library Assistant', 34000.00, 4),
(209, 'Ivy Clark', 'Library Technician', 36000.00, 5),
(210, 'Jack Miller', 'Assistant Librarian', 55000.00, 5),

(211, 'Karen White', 'Librarian', 56000.00, 6),
(212, 'Larry Green', 'Library Assistant', 35000.00, 6),
(213, 'Mona Black', 'Librarian', 58000.00, 7),
(214, 'Nathan King', 'Library Technician', 37000.00, 7),
(215, 'Olivia Scott', 'Assistant Librarian', 42000.00, 8),
(216, 'Paul Adams', 'Librarian', 59000.00, 8),
(217, 'Quincy Roberts', 'Library Assistant', 36000.00, 9),
(218, 'Rachel Turner', 'Librarian', 60000.00, 9),
(219, 'Steven Harris', 'Library Technician', 38000.00, 10),
(220, 'Tracy Young', 'Assistant Librarian', 41000.00, 10),
(221, 'Uma Thompson', 'Library Assistant', 37000.00, 1),
(222, 'Victor Perez', 'Library Technician', 39000.00, 1),
(223, 'Wendy Hall', 'Librarian', 60000.00, 1),
(224, 'Xander Nelson', 'Library Technician', 38000.00, 2),
(225, 'Yara Lewis', 'Library Assistant', 34000.00, 2),
(226, 'Zane Clark', 'Assistant Librarian', 42000.00, 3),
(227, 'Arthur King', 'Library Technician', 36000.00, 3),
(228, 'Bella Reed', 'Librarian', 57000.00, 4),
(229, 'Caleb Moore', 'Library Assistant', 33000.00, 4),
(230, 'Diana Foster', 'Librarian', 58000.00, 4),
(231, 'Ethan Brooks', 'Assistant Librarian', 40000.00, 5),
(232, 'Fiona Walker', 'Library Technician', 35000.00, 5),
(233, 'George Baker', 'Library Assistant', 32000.00, 6),
(234, 'Hannah Wood', 'Librarian', 61000.00, 6),
(235, 'Ian Stewart', 'Library Technician', 39000.00, 7),
(236, 'Judy Bell', 'Library Assistant', 34000.00, 7),
(237, 'Kevin Hughes', 'Librarian', 62000.00, 7),
(238, 'Linda Peterson', 'Assistant Librarian', 43000.00, 8),
(239, 'Mark Cooper', 'Library Technician', 36000.00, 8),
(240, 'Nina Murphy', 'Library Assistant', 35000.00, 8),
(241, 'Oscar Rogers', 'Librarian', 60000.00, 7),

```
(242, 'Pamela Mitchell', 'Library Assistant', 34000.00, 7),  
(243, 'Quinn Bryant', 'Assistant Librarian', 44000.00, 1),  
(244, 'Ruth Simmons', 'Library Technician', 37000.00, 1),  
(245, 'Sam Fisher', 'Library Assistant', 35000.00, 1);
```

```
Select* from Employee;
```

```
Create table Books(  
ISBN varchar(25) primary key,  
Book_Title varchar(50),  
Category varchar(20),  
Rental_Price decimal(5,2),  
Status char(3),  
Author varchar(20),  
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', 35.00, 'Yes', 'F. Scott Fitzgerald', 'Scribner'),
```

```
('978-0-14-017739-8', 'To Kill a Mockingbird', 'Fiction', 4.50, 'No', 'Harper Lee', 'J.B. Lippincott & Co.'),
```

```
('978-0-7432-7356-5', '1984', 'Dystopian', 6.00, 'Yes', 'George Orwell', 'Secker & Warburg'),
```

```
('978-0-452-28423-4', 'Pride and Prejudice', 'Romance', 44.00, 'No', 'Jane Austen', 'T. Egerton'),
```

```
('978-0-375-50420-7', 'The Catcher in the Rye', 'Fiction', 4.75, 'Yes', 'J.D. Salinger', 'Little, Brown and Company'),
```

```
('978-0-316-76948-8', 'The Hobbit', 'Fantasy', 5.25, 'No', 'J.R.R. Tolkien', 'George Allen & Unwin'),
```

```
('978-0-452-28425-8', 'Jane Eyre', 'Romance', 4.00, 'Yes', 'Charlotte Brontë', 'Smith, Elder & Co.'),
```

```
('978-0-14-044926-6', 'War and Peace', 'Historical', 27.00, 'No', 'Leo Tolstoy', 'The Russian Messenger'),
```

```
('978-0-307-27560-9', 'The Da Vinci Code', 'Thriller', 35.50, 'Yes', 'Dan Brown', 'Doubleday'),
```

```
('978-0-7432-7357-2', 'Animal Farm', 'Dystopian', 4.25, 'Yes', 'George Orwell', 'Secker & Warburg');
```

```
Select* from books;
```

```
Create table Customer(  

```

```

Customer_Id int primary key,
Customer_Name varchar(50),
Customer_Address varchar(50),
Reg_Date date
);

INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date)
VALUES
(1, 'John Doe', '123 Elm Street, Springfield, IL 62701', '2018-01-15'),
(2, 'Jane Smith', '456 Oak Avenue, Springfield, IL 62702', '2023-02-20'),
(3, 'Alice James', '789 Maple Lane, Springfield, IL 62703', '2023-03-10'),
(4, 'Bob Brown', '101 Pine Road, Springfield, IL 62704', '2019-04-25'),
(5, 'Carol White', '202 Cedar Street, Springfield, IL 62705', '2023-05-05'),
(6, 'David Green', '303 Birch Boulevard, Springfield, IL 62706', '2023-06-15'),
(7, 'Eva Blue', '404 Willow Way, Springfield, IL 62707', '2023-07-20'),
(8, 'Frank Black', '505 Walnut Drive, Springfield, IL 62708', '2020-08-30'),
(9, 'Grace Red', '606 Sycamore Circle, Springfield, IL 62709', '2023-09-25'),
(10, 'Henry Orange', '707 Poplar Court, Springfield, IL 62710', '2023-10-10');

```

```

Create table IssueStatus(
Issue_Id int primary key,
Issued_Cust int,
Issued_Book_Name varchar(50),
Issue_Date date,
Isbn_Book varchar(50),
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', '2018-06-01', '978-3-16-148410-0'),
(2, 2, '1984', '2023-07-15', '978-0-7432-7356-5'),

```

```
(3, 3, 'The Catcher in the Rye', '2023-08-10', '978-0-375-50420-7'),  
(4, 4, 'Animal Farm', '2023-09-05', '978-0-7432-7357-2'),  
(5, 5, 'The Hobbit', '2023-10-01', '978-0-316-76948-8'),  
(6, 6, 'To Kill a Mockingbird', '2020-11-20', '978-0-14-017739-8'),  
(7, 7, 'Pride and Prejudice', '2023-12-05', '978-0-452-28423-4'),  
(8, 8, 'Jane Eyre', '2024-01-10', '978-0-452-28425-8'),  
(9, 9, 'The history of the Peloponnesian War', '2019-02-15', '978-0-14-044926-6'),  
(10, 10, 'The Da Vinci Code', '2024-03-01', '978-0-307-27560-9');
```

```
Select* from IssueStatus;
```

```
Create table ReturnStatus(  
Return_Id int primary key,  
Return_Cust int,  
Return_Book_Name varchar(50),  
Return_Date date,  
Isbn_Book2 varchar(25),  
foreign key(Isbn_Book2) references Books(isbn)  
);
```

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

```
VALUES
```

```
(1, 1, 'The Great Gatsby', '2023-07-01', '978-3-16-148410-0'),
```

```
(2, 2, '1984', '2023-08-15', '978-0-7432-7356-5'),
```

```
(3, 3, 'The Catcher in the Rye', '2023-09-10', '978-0-375-50420-7'),
```

```
(4, 4, 'Animal Farm', '2023-10-05', '978-0-7432-7357-2'),
```

```
(5, 5, 'The Hobbit', '2023-11-01', '978-0-316-76948-8'),
```

```
(6, 6, 'To Kill a Mockingbird', '2023-12-20', '978-0-14-017739-8'),
```

```
(7, 7, 'Pride and Prejudice', '2024-01-05', '978-0-452-28423-4'),
```

```
(8, 8, 'Jane Eyre', '2024-02-10', '978-0-452-28425-8'),
```

```
(9, 9, 'War and Peace', '2024-03-15', '978-0-14-044926-6'),
```

```
(10, 10, 'The Da Vinci Code', '2024-04-01', '978-0-307-27560-9');
```

```
Select* from ReturnStatus;
```

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

Select Book_Title, Category, Rental_Price from Books;

Result Grid	Filter Rows:	Export:
Book_Title	Category	Rental_Price
To Kill a Mockingbird	Fiction	4.50
War and Peace	Historical	27.00
The Da Vinci Code	Thriller	35.50
The Hobbit	Fantasy	5.25
The Catcher in the Rye	Fiction	4.75
Pride and Prejudice	Romance	44.00
Jane Eyre	Romance	4.00

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
Bob Smith	550000.00
Kevin Hughes	62000.00
Hannah Wood	61000.00
Wendy Hall	60000.00
Oscar Rogers	60000.00
Rachel Turner	60000.00
Paul Adams	59000.00

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

Select IssueStatus.Issued_Book_Name, Customer.Customer_Name
from IssueStatus

JOIN Customer on Issuestatus.Issued_cust = customer.customer_id;

Issued_Book_Name	Customer_Name
The Great Gatsby	John Doe
1984	Jane Smith
The Catcher in the Rye	Alice James
Animal Farm	Bob Brown
The Hobbit	Carol White
To Kill a Mockingbird	David Green
Pride and Prejudice	Eva Blue

Result 12 x

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

Select Category, count(*) AS Total_Counts from books group by category;

	Category	Total_Counts
▶	Fiction	3
	Historical	1
	Thriller	1
	Fantasy	1
	Romance	2
	Dystopian	2

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
▶	Bob Smith	Assistant Librarian
	Carol Davis	Librarian
	Eva Brown	Librarian
	Frank Wilson	Library Assistant
	Grace Lee	Librarian
	Jack Miller	Assistant Librarian
	Karen White	Librarian

Employee 2 x

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

Select c.customer_name

from customer c

LEFT JOIN issuestatus i

on c.customer_id = i.issued_cust

where c.reg_date < '2022-01-01';

	customer_name
▶	John Doe
	Bob Brown
	Frank Black

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

Select Branch_No, count(*) As Total_Employee from employee group by branch_no;

Result Grid		Filter Rows:
	Branch_No	Total_Employee
▶	1	8
	2	4
	3	4
	4	5
	5	4
	6	4
	7	7

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

Select c.customer_name from customer c

inner join issuestatus i on c.customer_id = i.issued_cust

where issue_date between '2023-06-01' and '2023-06-30';

Result Grid		Filter Rows:
	customer_name	

9. Retrieve book_title from book table containing history.

Select Book_title from Books where Category Like '%history%';

Result Grid		Filter Rows:
	Book_title	

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

Select Branch_No, count(*) As Total_Employee from employee group by branch_no Having (Total_Employee) > 5;

Result Grid			Filter Rows:
	Branch_No	Total_Employee	
▶	1	8	
	7	7	

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

Select e.Emp_name As Manager_Name, b.Branch_Address

from Branch b

join employee e

on b.manager_id = e.emp_id;

Result Grid			Filter Rows:
	Manager_Name	Branch_Address	
▶	Alice Johnson	123 Main St, Cityville	
	Carol Davis	456 Elm St, Townsville	
	Eva Brown	789 Oak St, Villageville	
	Grace Lee	321 Pine St, Hamletville	
	Ivy Clark	654 Maple St, Boroughville	
	Karen White	987 Birch St, Suburbville	
	Mona Black	159 Cedar St, Metropolis	

Result 22 ×

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

SELECT DISTINCT C.Customer_name

FROM IssueStatus I

JOIN Books B ON I.Isbn_book = B.ISBN

JOIN Customer C ON I.Issued_cust = C.Customer_Id

WHERE B.Rental_Price > 25;

Result Grid		Filter Rows:
	Customer_name	
▶	Grace Red	
	Henry Orange	
	Eva Blue	
	John Doe	