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

```
3  CREATE TABLE  Branch (
4      Branch_no INT PRIMARY KEY,
5      Manager_Id INT,
6      Branch_address VARCHAR(100),
7      Contact_no VARCHAR(20)
8  );
9  INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
10 VALUES
11     (1, 101, '123 Main St, City A', '123-456-7890'),
12     (2, 102, '456 Oak Ave, City B', '456-789-0123'),
13     (3, 103, '111 Oak Ave, City B', '234-876-0123'),
14     (4, 104, '222 Oak Ave, City B', '567-532-0123');
15
16     select * from branch;
17
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Branch_no	Manager_Id	Branch_address	Contact_no
1	101	123 Main St, City A	123-456-7890
2	102	456 Oak Ave, City B	456-789-0123
3	103	111 Oak Ave, City B	234-876-0123
4	104	222 Oak Ave, City B	567-532-0123
NULL	NULL	NULL	NULL

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

```sql
CREATE TABLE  Employee (
    Emp_Id INT PRIMARY KEY,
    Emp_name VARCHAR(100),
    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 Doe', 'Manager', 60000, 1),
    (2, 'Jane Smith', 'Assistant', 45000, 1),
    (3, 'Michael Brown', 'Manager', 62000, 2),
    (4, 'Emily Johnson', 'Librarian', 48000, 1),
    (5, 'John Joseph', 'Assistant', 50000, 1),
    (6, 'Mathew Johnson', 'Librarian', 38000, 1),
    (7, 'Edison Mathew', 'Assistant', 45000, 1);
 select * from   Employee;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Emp_Id	Emp_name	Position	Salary	Branch_no
1	John Doe	Manager	60000.00	1
2	Jane Smith	Assistant	45000.00	1
3	Michael Brown	Manager	62000.00	2
4	Emily Johnson	Librarian	48000.00	1
5	John Joseph	Assistant	50000.00	1
6	Mathew Johnson	Librarian	38000.00	1
7	Edison Mathew	Assistant	45000.00	1
NULL	NULL	NULL	NULL	NULL

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

```sql
CREATE TABLE  Books (
    ISBN VARCHAR(20) PRIMARY KEY,
    Book_title VARCHAR(200),
    Category VARCHAR(50),
    Rental_Price DECIMAL(10, 2),
    Status ENUM('yes', 'no'),
    Author VARCHAR(100),
    Publisher VARCHAR(100)
);

INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)
VALUES
    ('978-0-306-40615-7', 'Introduction to Algorithms', 'Computer Science', 30.50, 'yes', 'Thomas H. Cormen', 'MIT Press'),
    ('978-0-321-99278-9', 'Database System Concepts', 'Computer Science', 28.00, 'yes', 'Abraham Silberschatz', 'McGraw-Hill'),
    ('978-0-13-213080-6', 'Artificial Intelligence: A Modern Approach', 'Computer Science', 32.75, 'no', 'Stuart Russell', 'Pearson'),
    ('978-1-4493-8091-5', 'JavaScript: The Good Parts', 'Programming history of Java', 25.00, 'yes', 'Douglas Crockford', 'O Reilly');
select * from books;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
978-0-13-213080-6	Artificial Intelligence: A Modern Approach	Computer Science	32.75	no	Stuart Russell	Pearson
978-0-306-40615-7	Introduction to Algorithms	Computer Science	30.50	yes	Thomas H. Cormen	MIT Press
978-0-321-99278-9	Database System Concepts	Computer Science	28.00	yes	Abraham Silberschatz	McGraw-Hill
978-1-4493-8091-5	JavaScript: The Good Parts	Programming history of Java	25.00	yes	Douglas Crockford	O Reilly
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

4. Customer

• Customer_Id

- Set as PRIMARY KEY
  • Customer_name
  • Customer_address
  • Reg_date

```
54
55  CREATE TABLE  Customer (
56      Customer_Id INT PRIMARY KEY,
57      Customer_name VARCHAR(100),
58      Customer_address VARCHAR(100),
59      Reg_date DATE
60  );
61  INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date)
62  VALUES
63      (1, 'Alice Johnson', '789 Elm St, City A', '2022-03-15'),
64      (2, 'Bob Williams', '567 Pine Ave, City B', '2020-12-10'),
65      (3, 'Eve Brown', '890 Cedar Rd, City A', '2023-01-20');
66      select * from Customer;
67
68  CREATE TABLE IssueStatus (
69      Issue_Id INT PRIMARY KEY,
70      Issued_cust INT,
71      Issued_book_name VARCHAR(200),
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Customer_Id	Customer_name	Customer_address	Reg_date
1	Alice Johnson	789 Elm St, City A	2022-03-15
2	Bob Williams	567 Pine Ave, City B	2020-12-10
3	Eve Brown	890 Cedar Rd, City A	2023-01-20
NULL	NULL	NULL	NULL

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

```sql
CREATE TABLE IssueStatus (
    Issue_Id INT PRIMARY KEY,
    Issued_cust INT,
    Issued_book_name VARCHAR(200),
    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, 'Introduction to Algorithms', '2023-05-10', '978-0-306-40615-7'),
    (2, 2, 'Database System Concepts', '2023-06-18', '978-0-321-99278-9'),
    (3, 3, 'JavaScript: The Good Parts', '2023-07-05', '978-1-4493-8091-5');
    select * from IssueStatus;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Issue_Id	Issued_cust	Issued_book_name	Issue_date	Isbn_book
1	1	Introduction to Algorithms	2023-05-10	978-0-306-40615-7
2	2	Database System Concepts	2023-06-18	978-0-321-99278-9
3	3	JavaScript: The Good Parts	2023-07-05	978-1-4493-8091-5

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

```sql
CREATE TABLE  ReturnStatus (
    Return_Id INT PRIMARY KEY,
    Return_cust INT,
    Return_book_name VARCHAR(200),
    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
    (1, 1, 'Introduction to Algorithms', '2023-05-30', '978-0-306-40615-7'),
    (2, 2, 'Database System Concepts', '2023-07-01', '978-0-321-99278-9');
   select * from  ReturnStatus;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
1	1	Introduction to Algorithms	2023-05-30	978-0-306-40615-7
2	2	Database System Concepts	2023-07-01	978-0-321-99278-9
NULL	NULL	NULL	NULL	NULL

Display all the tables and Write the queries for the following :

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

```sql
#1. Retrieve the book title, category, and rental price of all available books.
SELECT Book_title, Category, Rental_Price
FROM Books WHERE Status = 'yes';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_title	Category	Rental_Price
Introduction to Algorithms	Computer Science	30.50
Database System Concepts	Computer Science	28.00
JavaScript: The Good Parts	Programming history of Java	25.00

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

```
102     #2. List the employee names and their respective salaries in descending order of salary.
103 •   SELECT Emp_name, Salary
104     FROM Employee ORDER BY Salary DESC;
105
106     #3. Retrieve the book titles and the corresponding customers who have issued those books.
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Emp_name	Salary
Michael Brown	62000.00
John Doe	60000.00
John Joseph	50000.00
Emily Johnson	48000.00
Jane Smith	45000.00
Edison Mathew	45000.00
Mathew Johnson	38000.00

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

```
105
106     #3. Retrieve the book titles and the corresponding customers who have issued those books.
107 •   SELECT b.Book_title, c.Customer_name
108     FROM Books b
109     JOIN IssueStatus i ON b.ISBN = i.Isbn_book
110     JOIN Customer c ON i.Issued_cust = c.Customer_Id;
111
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_title	Customer_name
Introduction to Algorithms	Alice Johnson
Database System Concepts	Bob Williams
JavaScript: The Good Parts	Eve Brown

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

```
112     #4. Display the total count of books in each category.
113 •   SELECT Category, COUNT(*) AS Total_Count
114     FROM Books GROUP BY Category;
115
116     #5. Retrieve the employee names and their positions for the employees
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Category	Total_Count
Computer Science	3
Programming history of Java	1

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

```
116    #5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
117    SELECT Emp_name, Position
118    FROM Employee WHERE Salary > 50000;
119
120    #6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Emp_name	Position
John Doe	Manager
Michael Brown	Manager

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

```
1    #6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
2    SELECT Customer_name FROM Customer
3    WHERE Reg_date < '2022-01-01'
4    AND Customer_Id NOT IN (
5        SELECT Issued_cust    FROM IssueStatus
6    );
7
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_name

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

```
127
128    #7. Display the branch numbers and the total count of employees in each branch.
129    SELECT Branch_no, COUNT(*) AS Total_Employees
130    FROM Employee GROUP BY Branch_no;
131
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Branch_no	Total_Employees
1	6
2	1

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

```
132    #8. Display the names of customers who have issued books in the month of June 2023.
133    SELECT DISTINCT c.Customer_name
134    FROM Customer c
135    JOIN IssueStatus i ON c.Customer_Id = i.Issued_cust
136    WHERE YEAR(i.Issue_date) = 2023 AND MONTH(i.Issue_date) = 6;
137
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_name

9. Retrieve book_title from book table containing history.

```
138     #9. Retrieve book_title from book table containing history.
139 •   SELECT Book_title
140     FROM Books WHERE Category LIKE '%history%';
141
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_title
JavaScript: The Good Parts

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

```
141
142     #10.Retrieve the branch numbers along with the count of employees for branches having more than 5 employees
143 •   SELECT Branch_no, COUNT(*) AS Total_Employees
144     FROM Employee
145     GROUP BY Branch_no HAVING COUNT(*) > 5;
146
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Branch_no	Total_Employees
1	6

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

```
146
147     #11. Retrieve the names of employees who manage branches and their respective branch addresses.
148 •   SELECT e.Emp_name, b.Branch_address
149     FROM Employee e
150     JOIN Branch b ON e.Branch_no = b.Branch_no
151     WHERE e.Position = 'Manager';
152
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Emp_name	Branch_address
John Doe	123 Main St, City A
Michael Brown	456 Oak Ave, City B

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

```
153     #12.  Display the names of customers who have issued books with a rental price higher than Rs. 25.
154 •   SELECT DISTINCT c.Customer_name
155     FROM Customer c
156     JOIN IssueStatus i ON c.Customer_Id = i.Issued_cust
157     JOIN Books b ON i.Isbn_book = b.ISBN
158     WHERE b.Rental_Price > 25;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_name
Alice Johnson
Bob Williams