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
-

```sql
SELECT Emp_name, Position
FROM Employee
WHERE Salary > 50000;
```

Result Grid | Filter Rows:

Emp_name	Position
deepu	Manager
abin	Manager
asok	Manager

```sql
);
INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book)
VALUES
(1, 1, 'Introduction to Algorithms', '2023-06-05', '978-3-16-148410-0'),
(2, 2, 'Artificial Intelligence', '2023-06-15', '978-0-12-374856-0'),
(3, 4, 'The History of the Ancient World', '2023-06-20', '978-0-452-28423-4');

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	2023-06-05	978-3-16-148410-0
2	2	Artificial Intelligence	2023-06-15	978-0-12-374856-0
3	4	The History of the Ancient World	2023-06-20	978-0-452-28423-4
NULL	NULL	NULL	NULL	NULL

```sql
CREATE TABLE ReturnStatus (
    Return_Id INT PRIMARY KEY,
    Return_cust INT,
    Return_book_name VARCHAR(255),
    Return_date DATE,
    Isbn_book2 VARCHAR(130),
    FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
);
```

```sql
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
(1, 'John Doe', '789 Elm St', '2021-05-15'),
(2, 'Jane Smith', '456 Maple Ave', '2021-12-20'),
(3, 'Sam Brown', '123 Pine Rd', '2022-01-10'),
(4, 'Emily Davis', '321 Cedar Blvd', '2023-03-25');
select * from Customer;
```

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

Customer_Id	Customer_name	Customer_address	Reg_date
1	John Doe	789 Elm St	2021-05-15
2	Jane Smith	456 Maple Ave	2021-12-20
3	Sam Brown	123 Pine Rd	2022-01-10
4	Emily Davis	321 Cedar Blvd	2023-03-25
NULL	NULL	NULL	NULL

```sql
CREATE TABLE IssueStatus (
    Issue_Id INT PRIMARY KEY,
    Issued_cust INT,
    Issued_book_name VARCHAR(255),
    Issue_date DATE,
    Isbn_book VARCHAR(130),
    FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
);

CREATE DATABASE library;
USE library;
```

```sql
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, '123 Main St', '123-456-7890'),
(2, 102, '456 Oak Ave', '234-567-8901'),
(3, 103, '789 Pine Rd', '345-678-9012');
select * from Branch;
```

esult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Conte

Branch_no	Manager_Id	Branch_address	Contact_no
1	101	123 Main St	123-456-7890
2	102	456 Oak Ave	234-567-8901
3	103	789 Pine Rd	345-678-9012
NULL	NULL	NULL	NULL

```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)
);
```

```sql
INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no)
VALUES
(101, 'deepu', 'Manager', 75000, 1),
(102, 'abin', 'Manager', 72000, 2),
(103, 'asok', 'Manager', 71000, 3),
(104, 'nancy', 'Staff', 50000, 1),
(105, 'aswethy', 'Staff', 48000, 2),
(106, 'zainaba', 'Staff', 45000, 3);
select  * from Employee;
```

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

Emp_Id	Emp_name	Position	Salary	Branch_no
101	deepu	Manager	75000.00	1
102	abin	Manager	72000.00	2
103	asok	Manager	71000.00	3
104	nancy	Staff	50000.00	1
105	aswethy	Staff	48000.00	2
106	zainaba	Staff	45000.00	3
NULL	NULL	NULL	NULL	NULL

```sql
CREATE TABLE Books (
    ISBN VARCHAR(50)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)
);
```

```
46
47 •  INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)
48    VALUES
49    ('978-3-16-148410-0', 'Introduction to Algorithms', 'Computer Science', 35.00, 'yes', 'Thomas
50    ('978-0-12-374856-0', 'Artificial Intelligence', 'Computer Science', 45.00, 'yes', 'Stuart Rus
51    ('978-0-321-57351-3', 'Design Patterns', 'Software Engineering', 30.00, 'no', 'Erich Gamma',
52    ('978-1-56619-909-4', 'The Art of Computer Programming', 'Computer Science', 55.00, 'yes', 'Do
53    ('978-0-7432-7356-5', 'A Brief History of Time', 'History', 20.00, 'no', 'Stephen Hawking', 'B
54    ('978-0-452-28423-4', 'The History of the Ancient World', 'History', 25.00, 'yes', 'Susan Wise
55 •  select * from Books;
```

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

ISBN	Book_title	Category	Rental_Price	Status	Author
978-0-12-374856-0	Artificial Intelligence	Computer Science	45.00	yes	Stuart Russell
978-0-321-57351-3	Design Patterns	Software Engineering	30.00	no	Erich Gamma
978-0-452-28423-4	The History of the Ancient World	History	25.00	yes	Susan Wise Bauer
978-0-7432-7356-5	A Brief History of Time	History	20.00	no	Stephen Hawking
978-1-56619-909-4	The Art of Computer Programming	Computer Science	55.00	yes	Donald Knuth

```
95    );
96 •  INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)
97    VALUES
98    (1, 1, 'Introduction to Algorithms', '2023-06-25', '978-3-16-148410-0'),
99    (2, 2, 'Artificial Intelligence', '2023-06-30', '978-0-12-374856-0');
100 •  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	2023-06-25	978-3-16-148410-0
2	2	Artificial Intelligence	2023-06-30	978-0-12-374856-0

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

```
102 •  SELECT Book_title, Category, Rental_Price
103    FROM Books
104    WHERE Status = 'yes';
```

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

Book_title	Category	Rental_Price
Artificial Intelligence	Computer Science	45.00
The History of the Ancient World	History	25.00
The Art of Computer Programming	Computer Science	55.00
Introduction to Algorithms	Computer Science	35.00

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

```
106 •   SELECT Emp_name, Salary
107     FROM Employee
108     ORDER BY Salary DESC;
109
```

Result Grid | Filter Rows:

Emp_name	Salary
deepu	75000.00
abin	72000.00
asok	71000.00
nancy	50000.00
aswethy	48000.00
zainaba	45000.00

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

```
99
10 •   SELECT Books.Book_title, Customer.Customer_name
11     FROM IssueStatus
12     JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
13     JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
14
```

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

Book_title	Customer_name
Introduction to Algorithms	John Doe
Artificial Intelligence	Jane Smith
The History of the Ancient World	Emily Davis

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

```
114
115 •  SELECT Category, COUNT(*) AS Total_Books
116    FROM Books
117    GROUP BY Category;
118
```

Result Grid | Filter Rows: | Export: | Wrap

Category	Total_Books
Computer Science	3
Software Engineering	1
History	2

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

```
118
119 •  SELECT Emp_name, Position
120    FROM Employee
121    WHERE Salary > 50000;
122
```

Result Grid | Filter Rows:

Emp_name	Position
deepu	Manager
abin	Manager
asok	Manager

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

```
122
123 •  SELECT Customer_name
124    FROM Customer
125    WHERE Reg_date < '2022-01-01' AND Customer_Id NOT IN (SELECT Issued_cust FROM IssueStatus);
126
```

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

Customer_name

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

```
127 •   SELECT Branch_no, COUNT(*) AS Total_Employees
128     FROM Employee
129     GROUP BY Branch_no;
130
```

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

Branch_no	Total_Employees
1	2
2	2
3	2

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

```
131 •   SELECT Customer_name
132     FROM IssueStatus
133     JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id
134     WHERE Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
135
```

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

Customer_name
John Doe
Jane Smith
Emily Davis

9. Retrieve book_title from book table containing history.

```
136 •   SELECT Book_title
137     FROM Books
138     WHERE Book_title LIKE '%history%';
139
```

Result Grid | Filter Rows: | Export: | Wrap

Book_title
The History of the Ancient World
A Brief History of Time

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

```
140  SELECT Branch_no, COUNT(*) AS Total_Employees
141  FROM Employee
142  GROUP BY Branch_no
143  HAVING COUNT(*) > 5;
144
```

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

Branch_no	Total_Employees

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

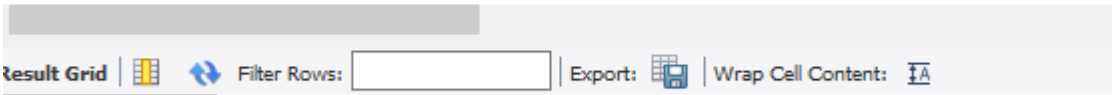
```
145  SELECT Employee.Emp_name, Branch.Branch_address
146  FROM Employee
147  JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;
148
```

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

Emp_name	Branch_address
deepu	123 Main St
abin	456 Oak Ave
asok	789 Pine Rd

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

```
148
149 •  SELECT DISTINCT Customer.Customer_name
150    FROM IssueStatus
151    JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
152    JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id
153    WHERE Books.Rental_Price > 25;
154
```



Customer_name
Jane Smith
John Doe