Mysql Comprehensive Assessment
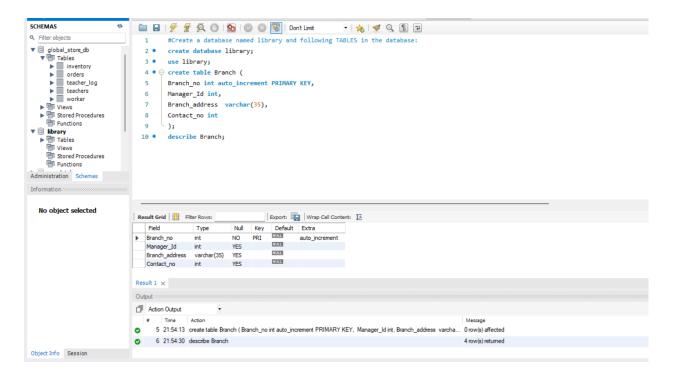
Topic : Library Management System
You are going to build a project based on the Library Management System. It keeps track of all information about books in the library, their cost, status and total number of books available in the library.

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

1. Branch
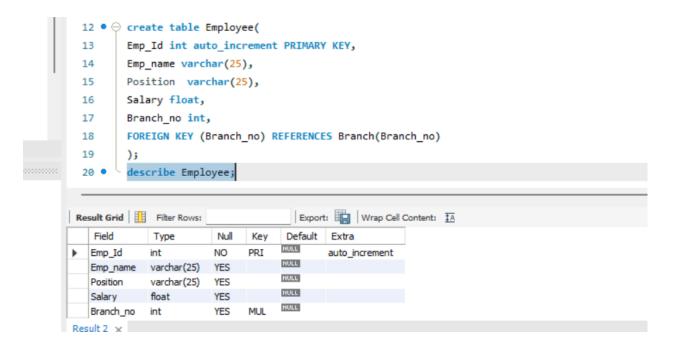2. Employee
3. Books
4. Customer
5. IssueStatus
6. ReturnStatus

Attributes for the tables:

1. Branch

• Branch_no

- ● Set as PRIMARY KEY
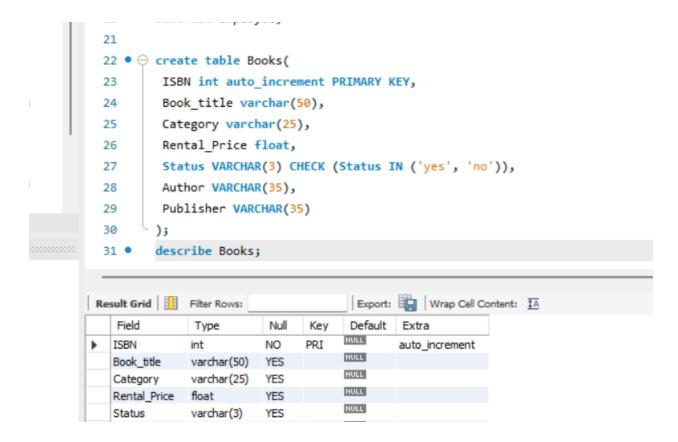 • Manager_Id
  • Branch_address
  • Contact_no

```
1    #Create a database named library and following TABLES in the database:
2  •  create database library;
3  •  use library;
4  •⊖  create table Branch (
5    Branch_no int auto_increment PRIMARY KEY,
6    Manager_Id int,
7    Branch_address  varchar(35),
8    Contact_no int
9    );
10 •  describe Branch;
```

Field	Type	Null	Key	Default	Extra
Branch_no	int	NO	PRI	NULL	auto_increment
Manager_Id	int	YES		NULL	
Branch_address	varchar(35)	YES		NULL	
Contact_no	int	YES		NULL	

Result 1 ×

Output

Action Output

#	Time	Action	Message
5	21:54:13	create table Branch (Branch_no int auto_increment PRIMARY KEY, Manager_Id int, Branch_address varcha...	0 row(s) affected
6	21:54:30	describe Branch	4 row(s) returned

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

```
12 •  ⊖  create table Employee(
13         Emp_Id int auto_increment PRIMARY KEY,
14         Emp_name varchar(25),
15         Position  varchar(25),
16         Salary float,
17         Branch_no int,
18         FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
19         );
20 •       describe Employee;
```

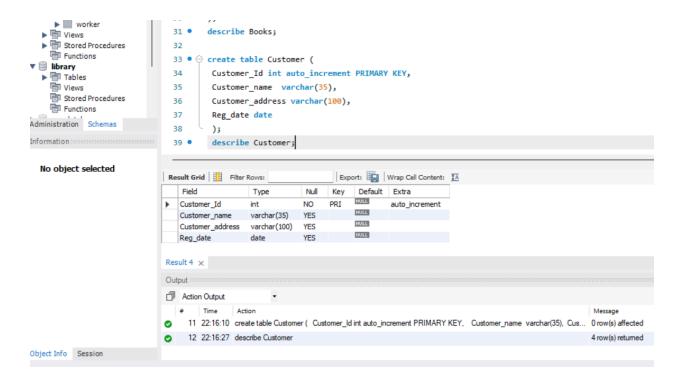| Field	Type	Null	Key	Default	Extra
Emp_Id	int	NO	PRI	NULL	auto_increment
Emp_name	varchar(25)	YES		NULL	
Position	varchar(25)	YES		NULL	
Salary	float	YES		NULL	
Branch_no	int	YES	MUL	NULL	

Result 2 ✕

3. Books

• ISBN

- Set as PRIMARY KEY
  • Book_title
  • Category
  • Rental_Price
  • Status [Give yes if book available and no if book not available]
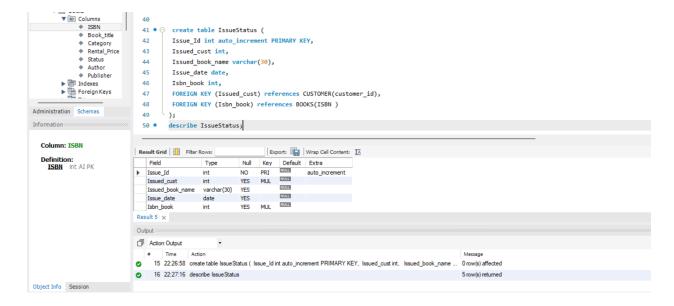 • Author
  • Publisher

```
21
22 •  ⊖  create table Books(
23           ISBN int auto_increment PRIMARY KEY,
24           Book_title varchar(50),
25           Category varchar(25),
26           Rental_Price float,
27           Status VARCHAR(3) CHECK (Status IN ('yes', 'no')),
28           Author VARCHAR(35),
29           Publisher VARCHAR(35)
30        );
31 •   describe Books;
```

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

	Field	Type	Null	Key	Default	Extra
▶	ISBN	int	NO	PRI	NULL	auto_increment
	Book_title	varchar(50)	YES		NULL	
	Category	varchar(25)	YES		NULL	
	Rental_Price	float	YES		NULL	
	Status	varchar(3)	YES		NULL	

4. Customer

• Customer_Id

- Set as PRIMARY KEY
  • Customer_name
  • Customer_address
  • Reg_date

```
31 •    describe Books;
32
33 • ⊖  create table Customer (
34          Customer_Id int auto_increment PRIMARY KEY,
35          Customer_name varchar(35),
36          Customer_address varchar(100),
37          Reg_date date
38        );
39 •    describe Customer;
```

Field	Type	Null	Key	Default	Extra
Customer_Id	int	NO	PRI	NULL	auto_increment
Customer_name	varchar(35)	YES		NULL	
Customer_address	varchar(100)	YES		NULL	
Reg_date	date	YES		NULL	

Result 4 ×

Output

Action Output

#	Time	Action	Message
✅	11 22:16:10	create table Customer (Customer_Id int auto_increment PRIMARY KEY, Customer_name varchar(35), Cus...	0 row(s) affected
✅	12 22:16:27	describe Customer	4 row(s) returned

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
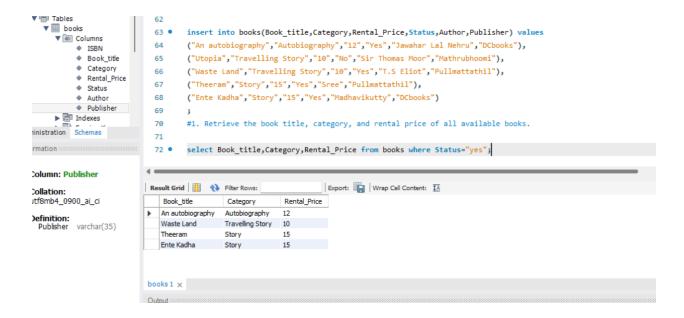


```
40
41 • ⊖  create table IssueStatus (
42          Issue_Id int auto_increment PRIMARY KEY,
43          Issued_cust int,
44          Issued_book_name varchar(30),
45          Issue_date date,
46          Isbn_book int,
47          FOREIGN KEY (Issued_cust) references CUSTOMER(customer_id),
48          FOREIGN KEY (Isbn_book) references BOOKS(ISBN )
49        );
50 •    describe IssueStatus;
```

Column: ISBN

Definition:
ISBN int AI PK

Field	Type	Null	Key	Default	Extra
Issue_Id	int	NO	PRI	NULL	auto_increment
Issued_cust	int	YES	MUL	NULL	
Issued_book_name	varchar(30)	YES		NULL	
Issue_date	date	YES		NULL	
Isbn_book	int	YES	MUL	NULL	

Result 5 ×

Output

Action Output

#	Time	Action	Message
✅	15 22:26:58	create table IssueStatus (Issue_Id int auto_increment PRIMARY KEY, Issued_cust int, Issued_book_name ...	0 row(s) affected
✅	16 22:27:16	describe IssueStatus	5 row(s) returned

6. ReturnStatus

• Return_Id

- ● Set as PRIMARY KEY
  - • Return_cust
  - • Return_book_name
  - • Return_date
  - • Isbn_book2
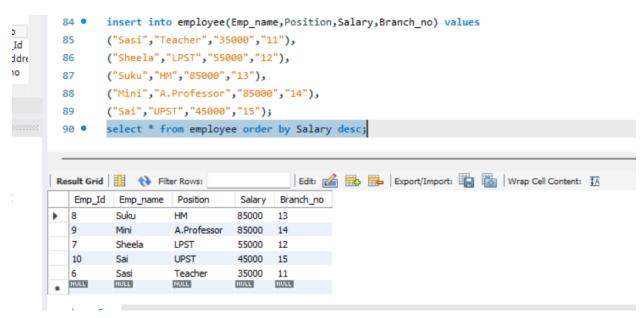- ● Set as FOREIGN KEY and it should refer isbn in BOOKS table

```
52 •  create table ReturnStatus (
53       Return_Id int auto_increment PRIMARY KEY ,
54       Return_cust int,
55       Return_book_name varchar(45),
56       Return_date date,
57       Isbn_book2 int,
58       FOREIGN KEY(Isbn_book2) references BOOKS(isbn),
59       FOREIGN KEY(Return_cust) references customer(Customer_Id)
60     );
61 •   describe ReturnStatus;
```

Field	Type	Null	Key	Default	Extra
Return_Id	int	NO	PRI	NULL	auto_increment
Return_cust	int	YES	MUL	NULL	
Return_book_name	varchar(45)	YES		NULL	
Return_date	date	YES		NULL	
Isbn_book2	int	YES	MUL	NULL	

#	Time	Action	Message
19	22:33:37	describe ReturnStatus	5 row(s) returned
20	22:33:54	describe ReturnStatus	5 row(s) returned

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

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

```
62
63 ●   insert into books(Book_title,Category,Rental_Price,Status,Author,Publisher) values
64       ("An autobiography","Autobiography","12","Yes","Jawahar Lal Nehru","DCbooks"),
65       ("Utopia","Travelling Story","10","No","Sir Thomas Moor","Mathrubhoomi"),
66       ("Waste Land","Travelling Story","10","Yes","T.S Eliot","Pullmattathil"),
67       ("Theeram","Story","15","Yes","Sree","Pullmattathil"),
68       ("Ente Kadha","Story","15","Yes","Madhavikutty","DCbooks")
69       ;
70       #1. Retrieve the book title, category, and rental price of all available books.
71
72 ●   select Book_title,Category,Rental_Price from books where Status="yes";
```

Tables
  books
    Columns
      ◆ ISBN
      ◆ Book_title
      ◆ Category
      ◆ Rental_Price
      ◆ Status
      ◆ Author
      ◆ Publisher
    ▶ Indexes
inistration   Schemas

rmation

Column: Publisher

Collation:
tf8mb4_0900_ai_ci

Definition:
  Publisher   varchar(35)

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

Book_title	Category	Rental_Price
An autobiography	Autobiography	12
Waste Land	Travelling Story	10
Theeram	Story	15
Ente Kadha	Story	15

books 1 ×

Output

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

```
84 ●   insert into employee(Emp_name,Position,Salary,Branch_no) values
85       ("Sasi","Teacher","35000","11"),
86       ("Sheela","LPST","55000","12"),
87       ("Suku","HM","85000","13"),
88       ("Mini","A.Professor","85000","14"),
89       ("Sai","UPST","45000","15");
90 ●   select * from employee order by Salary desc;
```

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

Emp_Id	Emp_name	Position	Salary	Branch_no
8	Suku	HM	85000	13
9	Mini	A.Professor	85000	14
7	Sheela	LPST	55000	12
10	Sai	UPST	45000	15
6	Sasi	Teacher	35000	11
NULL	NULL	NULL	NULL	NULL

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

```
117 •   select * from issuestatus;
118
119 •   SELECT *
120     FROM issuestatus full JOIN customer;
121
122 •   SELECT i.Issued_book_name,c.Customer_name
123     FROM issuestatus i
124     left JOIN customer c ON c.Customer_Id = i.Issued_cust;
125
```

Column: Customer_name

Collation:
utf8mb4_0900_ai_ci

Definition:
Customer_name   varchar(35)

Issued_book_name	Customer_name
An autobiography	Meera
Theeram	Rose
Waste Land	Sree
Ente Kadha	Mini
Ente Kadha	Rose
An autobiography	Rose
Theeram	Meera

Result 37 ×

```
114
115 •   SELECT *
116     FROM issuestatus i
117     left JOIN customer c ON c.Customer_Id = i.Issued_cust;
118
```

Issue_Id	Issued_cust	Issued_book_name	Issue_date	Isbn_book	Customer_Id	Customer_name	Customer_address	Reg_date
1	16	An autobiography	2024-02-02	6	16	Meera	Meera villa	2012-06-06
2	20	Theeram	2024-02-02	9	20	Rose	Rose villa	2012-02-06
3	18	Waste Land	2024-02-12	8	18	Sree	Sreevalsam	2012-10-10
4	17	Ente Kadha	2024-12-02	10	17	Mini	Mini villa	2014-10-06
5	20	Ente Kadha	2024-12-02	7	20	Rose	Rose villa	2012-02-06

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

```
119     #4. Display the total count of books in each category.
120 •   select Category,count(Category) as Tot_No from books group by Category;
```

Category	Tot_No
Autobiography	1
Travelling Story	2
Story	2

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

```
122      #5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
123 •    select Emp_Id,Emp_name,Position,Salary from  employee where Salary > 50000 order by salary desc;
```

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

Emp_Id	Emp_name	Position	Salary
8	Suku	HM	85000
9	Mini	A.Professor	85000
7	Sheela	LPST	55000
NULL	NULL	NULL	NULL

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

```
125      #6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
126
127 •    select * from customer c where c.Reg_date < "2022-01-01" and (select count(*) from issuestatus d where d.Issued_cust= c.Customer_Id) =0;
128
129
```

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

Customer_Id	Customer_name	Customer_address	Reg_date
19	Ami	Aami villa	2017-07-08
NULL	NULL	NULL	NULL

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

```
133
134      #7. Display the branch numbers and the total count of employees in each branch.
135
136 •    SELECT b.Branch_no, COUNT(e.Emp_Id) AS total_employees
137      FROM branch b
138      LEFT JOIN employee e ON b.Branch_no = e.Branch_no
139      GROUP BY b.Branch_no;
140
```

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

Branch_no	total_employees
11	2
12	2
13	1
14	1
15	1

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

```
140
141        #8. Display the names of customers who have issued books in the month of June 2023.
142
143  ●    insert into issuestatus(Issued_cust,Issued_book_name,Issue_date,Isbn_book) values
144        (20,"An autobiography","2023-06-02",6),
145        (16,"Theeram","2023-06-02",9),
146        (19,"Waste Land","2023-06-12",8);
147
148  ●    SELECT DISTINCT c.Customer_name
149        FROM customer c
150        JOIN issuestatus i ON c.Customer_Id = i.Issued_cust
151        WHERE MONTH(i.Issue_date) = 6 AND YEAR(i.Issue_date) = 2023;
152
153
```

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

Customer_name
▶ Meera
Ami
Rose

9. Retrieve book_title from a book table containing history.

```
155
156        #9. Retrieve book_title from a book table containing history.
157
158  ●    insert into books(Book_title,Category,Rental_Price,Status,Author,Publisher) values
159        ("Utopia2","History","10","No","Sir Thomas Moor","Mathrubhoomi"),
160        ("Columbus","History","10","Yes","T.S Eliot","Pullmattathil");
161
162  ●    select Book_title from books where Category="History";
163
```

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

Book_title
▶ Utopia2
Columbus

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

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

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

Score Distribution:
1 point for correctly formulating the query of each question (12 x 1 = 12).
1 point for providing screenshots of the output for each question (12 x 1 = 12).
1 point for timely submission.

Total = 25.

PS : After completing the project upload your project with screenshots in the github and share the link.

Queries Used
#Create a database named library and following TABLES in the database:
create database library;
use library;
create table Branch (
Branch_no int auto_increment PRIMARY KEY,
Manager_Id int,
Branch_address varchar(35),
Contact_no int
);
describe Branch;

create table Employee(
Emp_Id int auto_increment PRIMARY KEY,
Emp_name varchar(25),
Position varchar(25),
Salary float,
Branch_no int,
FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
);
describe Employee;

create table Books(
** ISBN int auto_increment PRIMARY KEY,**

```sql
 Book_title varchar(50),
 Category varchar(25),
 Rental_Price float,
 Status VARCHAR(3) CHECK (Status IN ('yes', 'no')),
 Author VARCHAR(35),
 Publisher VARCHAR(35)
);
describe Books;

create table Customer (
 Customer_Id int auto_increment PRIMARY KEY,
 Customer_name  varchar(35),
 Customer_address varchar(100),
 Reg_date date
 );
 describe Customer;

 create table IssueStatus (
 Issue_Id int auto_increment PRIMARY KEY,
 Issued_cust int,
 Issued_book_name varchar(30),
 Issue_date date,
 Isbn_book int,
 FOREIGN KEY (Issued_cust) references
CUSTOMER(customer_id),
 FOREIGN KEY (Isbn_book) references BOOKS(ISBN )
);
describe IssueStatus;

create table ReturnStatus (
```

```sql
  Return_Id int auto_increment PRIMARY KEY ,
  Return_cust int,
  Return_book_name varchar(45),
  Return_date date,
  Isbn_book2 int,
  FOREIGN KEY(Isbn_book2) references BOOKS(isbn),
  FOREIGN KEY(Return_cust) references
customer(Customer_Id)
);
describe ReturnStatus;

insert into
books(Book_title,Category,Rental_Price,Status,Author,Publis
her) values
("An autobiography","Autobiography","12","Yes","Jawahar
Lal Nehru","DCbooks"),
("Utopia","Travelling Story","10","No","Sir Thomas
Moor","Mathrubhoomi"),
("Waste Land","Travelling Story","10","Yes","T.S
Eliot","Pullmattathil"),
("Theeram","Story","15","Yes","Sree","Pullmattathil"),
("Ente
Kadha","Story","15","Yes","Madhavikutty","DCbooks")
;
```

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

```sql
select Book_title,Category,Rental_Price from books where
Status="yes";
```

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

```sql
insert into branch(Manager_Id,Branch_address,Contact_no)
values
(101,"Kadalikad Jn",1234567810),
(102,"Janatha Library Jn",1111111111),
(103,"SALPS",222222222),
(104,"Matha Library",333333333),
("105","Kerala",444444444);
select * from branch;
delete from branch where Branch_no>15;
insert into employee(Emp_name,Position,Salary,Branch_no)
values
("Sasi","Teacher","35000","11"),
("Sheela","LPST","55000","12"),
("Suku","HM","85000","13"),
("Mini","A.Professor","85000","14"),
("Sai","UPST","45000","15");

insert into employee(Emp_name,Position,Salary,Branch_no)
values
("Shinny","Teacher","35000","11"),
("Shubha","LPST","55000","12");

select * from employee order by Salary desc;
```

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

```sql
describe issuestatus;
insert into
customer(Customer_name,Customer_address,Reg_date)
values
("Meera","Meera villa","2012-06-06"),
("Mini","Mini villa","2014-10-06"),
("Sree","Sreevalsam","2012-10-10"),
("Ami","Aami villa","2017-07-08"),
("Rose","Rose villa","2012-02-06");
select * from customer;


insert into
issuestatus(Issued_cust,Issued_book_name,Issue_date,Isbn
_book) values
(16,"An autobiography","2024-02-02",6),
(20,"Theeram","2024-02-02",9),
(18,"Waste Land","2024-02-12",8),
(17,"Ente Kadha","2024-12-02",10),
(20,"Ente Kadha","2024-12-02",7);

select * from issuestatus;

SELECT *
FROM issuestatus full JOIN customer;

SELECT i.Issued_book_name,c.Customer_name
FROM issuestatus i
```

left JOIN customer c ON c.Customer_Id = i.Issued_cust;

#4. Display the total count of books in each category.
select Category,count(Category) as Tot_No 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_Id,Emp_name,Position,Salary from employee where Salary > 50000 order by salary desc;

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

select * from customer c where c.Reg_date < "2022-01-01" and (select count(*) from issuestatus d where d.Issued_cust= c.Customer_Id) =0;

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

SELECT b.Branch_no, COUNT(e.Emp_Id) AS total_employees
FROM branch b
LEFT JOIN employee e ON b.Branch_no = e.Branch_no
GROUP BY b.Branch_no;

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

```sql
insert into
issuestatus(Issued_cust,Issued_book_name,Issue_date,Isbn
_book) values
(20,"An autobiography","2023-06-02",6),
(16,"Theeram","2023-06-02",9),
(19,"Waste Land","2023-06-12",8);

SELECT DISTINCT c.Customer_name
FROM customer c
cross JOIN issuestatus i ON c.Customer_Id = i.Issued_cust
WHERE MONTH(i.Issue_date) = 6 AND YEAR(i.Issue_date) =
2023;
```

#9. Retrieve book_title from a book table containing history.

```sql
insert into
books(Book_title,Category,Rental_Price,Status,Author,Publis
her) values
("Utopia2","History","10","No","Sir Thomas
Moor","Mathrubhoomi"),
("Columbus","History","10","Yes","T.S Eliot","Pullmattathil");

select Book_title from books where Category="History";
```

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

```sql
insert into employee(Emp_name,Position,Salary,Branch_no)
values
```

```sql
("Salu","Teacher","35000","11"),
("Sulu","LPST","55000","11"),
("Mini","HM","85000","11"),
("Meera","A.Professor","85000","1"),
("Chithra","UPST","45000","15");

SELECT b.Branch_no,b.Branch_address, COUNT(e.Emp_Id)
AS employee_count
FROM branch b
JOIN employee e ON b.Branch_no = e.Branch_no
GROUP BY b.Branch_no
HAVING COUNT(e.Emp_Id) > 5;
```

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

```sql
select e.Emp_name , b.Branch_address from employee e left
join branch b on b.Branch_no = e.Branch_no;
```

#12. Display the names of customers who have issued books with a rental price higher than Rs. 25.
```sql
insert into
books(Book_title,Category,Rental_Price,Status,Author,Publisher) values
("English","Sbject","25","Yes","Jawahar Lal
Nehru","DCbooks"),
("Maths","Subject","25","No","Sir Thomas
Moor","Mathrubhoomi"),
("History","History","25","Yes","T.S Eliot","Pullmattathil");
```

```sql
insert into
issuestatus(Issued_cust,Issued_book_name,Issue_date,Isbn
_book) values
(20,"English","2023-06-02",13),
(16,"Maths","2023-06-02",14),
(19,"Maths","2023-06-12",14);
update books set Rental_Price=30 where Rental_Price=25;
SET SQL_SAFE_UPDATES = 0;
select
b.Book_title,b.Rental_Price,i.Issued_cust,c.Customer_name
from books b
 left join issuestatus i on i.Isbn_book=b.ISBN
 left join customer c on c.Customer_Id=i.Issued_cust  where
b.Rental_Price>25 ;
select b.Book_title,c.Customer_name from books b
left join issuestatus i on i.Isbn_book=b.ISBN
left join customer c on c.Customer_Id=i.Issued_cust
 where b.Rental_Price>25 and c.Customer_name is not null;
```