

Library Management System (MYSQL PROJECT)

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

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

. Branch .










- Branch_no - Set as PRIMARY KEY
- Manager_Id
- Branch_address
- Contact_no

```
create database Library;
```





```
use Library;
```

```
create table Branch( Branch_no int auto_increment primary key,  
Manage_ID int,Address varchar  
(200), Contact_no int);
```

SQL File 6

 Limit to 1000 rows

```
6 • insert into Branch(Manage_ID,Address,Contact_no)
7 values(1001,'Dubai-public',05588789),
8 (1002,'Al Tawar',023659),
9 (1005,'Al mamzar beach',012356),
10 (10023,'Al mankhool',056148),
11 (563,'Al rashidiya',05698745),
```

Result Grid  Filter Rows: Edit:    Ex

	Branch_no	Manage_ID	Address	Contact_no
▶	1	1001	Dubai-public	5588789
	2	1002	Al Tawar	23659
	3	1005	Al mamzar beach	12356
	4	10023	Al mankhool	56148
	5	563	Al rashidiya	5698745
	6	5623	DWC knowledge	4569325
	7	897	UMM suquien	6347895
	8	5671	House of wisdom	3156400
	9	7836	Khalifa park library	5632470
	10	830	Al Oqoud	2300456
•	NULL	NULL	NULL	NULL

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

```






create table Employee(
Emp_Id int auto_increment primary key,
Emp_name varchar(50),
Position_area varchar(50),
Salary decimal(10,3),
Branch_no int,foreign key (Branch_no) references Branch(Branch_no));

```

```

44      ('Sajna','Librarian',25000,10);
45 •    select*from Employee;

```

Result Grid					
Filter Rows:		Edit:    Export/Import:   Wrap Cell Content:			
	Emp_Id	Emp_name	Position_area	Salary	Branch_no
▶	1	Basheer-Uv	Manager	75000.000	1
	2	Maimoona_nazar	Asst_Manager	55000.000	2
	3	Sijas s	Branch Manager	45000.000	3
	4	Arunjith	Librarian	45000.000	4
	5	Luca	Childrens_librarian	52000.000	5
	6	Faizal	Teeans_manager	65000.000	6
	7	Aslam	Branch_coordinator	360000.000	7
	8	Kiran	Classsic_librarian	96000.000	8
	9	Jaffer	Branch_manager	450000.000	9
	10	Sajna	Librarian	25000.000	10
*	NULL	NULL	NULL	NULL	NULL

Employee 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

create table Books(

ISBN int primary key,

Book_title varchar(250) not null,

Category varchar(50),

Rental_Price decimal(10,2),

Status enum('yes','no') default 'yes',

Author varchar(100),

Publisher varchar (255));

68 • desc Books;

69

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

Field	Type	Null	Key	Default	Extra
ISBN	varchar(255)	NO	PRI	NULL	
Book_title	varchar(250)	NO		NULL	
Category	varchar(50)	YES		NULL	
Rental_Price	decimal(10,2)	YES		NULL	
Status	enum('yes','no')	YES		yes	
Author	varchar(100)	YES		NULL	
Publisher	varchar(255)	YES		NULL	

Result 2 x Read Only

Output

Action Output

#	Time	Action
✓ 1	10:29:08	use Library
✓ 2	10:29:14	select * from Branch LIMIT 0, 1000
✗ 3	18:14:06	insert into Books(ISBN,Book_title,Category,Rental_Price,Status,Author,Publisher) values ('789-456-321','The G...
✓ 4	18:14:45	insert into Books(ISBN,Book_title,Category,Rental_Price,Status,Author,Publisher) values ('789-456-321','The G...
✓ 5	18:15:05	desc Books

Customer

- Customer_Id - Set as PRIMARY KEY
- Customer_name
- Customer_address
- Reg_date

create table Customer (Customer_id int primary key ,

Customer_Name varchar (250),

Customer_Add varchar (100), Reg_Date date);

```

71 • create table Customer (Customer_id int primary key ,
72   Customer_Name varchar (250),
73   Customer_Add varchar (100),
74   Reg_Date date);
75 • desc Customer;

```

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

Field	Type	Null	Key	Default	Extra
Customer_id	int	NO	PRI	NUL	
Customer_Name	varchar(250)	YES		NUL	
Customer_Add	varchar(100)	YES		NUL	
Reg_Date	date	YES		NUL	

Result 4 x Read Only

Output

Action Output

#	Time	Action	Me
3	18:14:06	insert into Books(ISBN,Book_title,Category,Rental_Price,Status,Author,Publisher) values ('789-456-321','The ...	Em
4	18:14:45	insert into Books(ISBN,Book_title,Category,Rental_Price,Status,Author,Publisher) values ('789-456-321','The ...	9 r
5	18:15:05	desc Books	7 r
6	18:17:13	select* from Books LIMIT 0, 1000	9 r
7	18:19:53	create table Customer (Customer_id int primary key , Customer_Name varchar (250). Customer_Add varchar (1...	0 r
8	18:20:11	desc Customer	4 r

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

```
create table IssueStatus(Issue_Id INT PRIMARY KEY, Issued_custid int, Issued_book_name
VARCHAR(255), Issue_date DATE, Isbn_book varchar (250), FOREIGN KEY (Issued_custid)
references Customer (Customer_id), FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN));

desc IssueStatus;
```

```

89      Issued_custid int,
90      Issued_book_name VARCHAR(255),
91      Issue_date DATE,
92      Isbn_book varchar (250),
93      FOREIGN KEY (Issued_custid) references Customer (Customer_id),
94      FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN));
95 •    desc Customer;
96 •    desc IssueStatus;_

```

Field	Type	Null	Key	Default	Extra
Issue_Id	int	NO	PRI	NONE	
Issued_custid	int	YES	MUL	NONE	
Issued_book_name	varchar(255)	YES		NONE	
Issue_date	date	YES		NONE	
Isbn_book	varchar(250)	YES	MUL	NONE	

Result 7 ×

Output

Action Output

#	Time	Action
✗ 16	18:44:01	create table IssueStatus(Issue_Id INT PRIMARY KEY, Issued_custid int, Issued_book_name VARCHAR(255...
✗ 17	18:45:53	create table IssueStatus(Issue_Id INT PRIMARY KEY, Issued_custid int, Issued_book_name VARCHAR(255...
✗ 18	18:47:25	create table IssueStatus(Issue_Id INT PRIMARY KEY, Issued_custid int, Issued_book_name VARCHAR(255...
✓ 19	18:47:52	create table IssueStatus(Issue_Id INT PRIMARY KEY, Issued_custid int, Issued_book_name VARCHAR(255...
✓ 20	18:48:12	desc Customer
✓ 21	18:48:48	desc IssueStatus

Limit to 1000 rows

```

107 values
108 (808,156,'A Boy at Seven', '2024-05-07 ', '1-86092-022-5');
109 • insert into IssueStatus(Issue_id,Issued_custid,Issued_book_name,Issue_date,Isbn_book
110 values
111 (809,157,'The Hostage','2021-12-12','1-86092-014-4');
112 • insert into IssueStatus(Issue_id,Issued_custid,Issued_book_name,Issue_date,Isbn_book
113 values
114 (810,160,'An Occurrence at Owl Creek Bridge One of the Missing','2024-11-12','1-860

```

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

	Issue_Id	Issued_custid	Issued_book_name	Issue_date	Isbn_book
▶	805	151	The Rocking-Horse Winner	2023-12-03	1-86092-007-1
	806	152	The Dancing Partner: Clocks	2024-01-03	1-86092-050-0
	807	160	The Grass is Always Greener	2024-06-11	789-456-321
	808	156	A Boy at Seven	2024-05-07	1-86092-022-5
	809	157	The Hostage	2021-12-12	1-86092-014-4
	810	160	An Occurrence at Owl Creek Bridge One of the ...	2024-11-12	1-86092-006-3
*	NULL	NULL	NULL	NULL	NULL

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 table Return_status(Return_Id INT PRIMARY KEY, Return_cust INT,
Return_book_name VARCHAR(255), Return_date DATE, Isbn_book2 varchar (250),
FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_id), FOREIGN KEY
(Isbn_book2) REFERENCES Books(ISBN));
```

```
desc Return_status;
```

The screenshot shows a SQL IDE with a toolbar at the top. The main editor contains the following SQL code:

```

119 Return_cust INT,
120 Return_book_name VARCHAR(255),
121 Return_date DATE,
122 Isbn_book2 varchar (250),
123 FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_id),
124 FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN));
125 • desc Return_status;
126
127 • insert into Return_status(Return_Id,Return_cust,Return_book_name,Return_date,Isbn_b

```

Below the editor is the 'Result Grid' tab, which displays the structure of the 'Return_status' table:

	Field	Type	Null	Key	Default	Extra
▶	Return_Id	int	NO	PRI	<small>NULL</small>	
	Return_cust	int	YES	MUL	<small>NULL</small>	
	Return_book_name	varchar(255)	YES		<small>NULL</small>	
	Return_date	date	YES		<small>NULL</small>	
	Isbn_book2	varchar(250)	YES	MUL	<small>NULL</small>	

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

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

```

SELECT Book_title, Category, Rental_Price
FROM Books
WHERE Status = 'yes';

```


138

139 • **SELECT** Book_title, Category, Rental_Price

140 **FROM** Books

141 **WHERE** Status = 'yes';

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Book_title	Category	Rental_Price	
An Occurrence at Owl Creek Bridge One of the ...	Adventure	6.50	
The Dancing Partner: Clocks	Comedy	5.00	
The Hostage	Adventure	8.00	
The Open Boat	Classic	5.00	
The Grass is Always Greener	Modern Times	10.00	

Books 7 x

Read Only

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

SELECT Emp_name, Salary

FROM Employe

ORDER BY Salary DESC;

```

143 • SELECT Emp_name, Salary
144 FROM Employee
145 ORDER BY Salary DESC;
146

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Emp_name	Salary			
▶	Jaffer	450000.000			
	Aslam	360000.000			
	Kiran	96000.000			
	Basheer-Uv	75000.000			
	Faizal	65000.000			
	Maimoona_nazar	55000.000			
	Luca	52000.000			
	Sijas s	45000.000			
	Arunjith	45000.000			
	Sajna	25000.000			

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

```

SELECT Books.Book_title, Customer.Customer_Name
FROM Books
JOIN IssueStatus ON Books.ISBN = IssueStatus.Isbn_book
JOIN Customer ON IssueStatus.Issued_custid = Customer.Customer_Id;

```

```

146
147 • SELECT Books.Book_title, Customer.Customer_Name
148 FROM Books
149 JOIN IssueStatus ON Books.ISBN = IssueStatus.Isbn_book
150 JOIN Customer ON IssueStatus.Issued_custid = Customer.Customer_Id;
151

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Book_title	Customer_Name			
▶	The Rocking-Horse Winner	Malathi_s			
	The Dancing Partner: Clocks	Aira Dilshad			
	The Grass is Always Greener	Kalyani			
	A Boy at Seven	Maimoona			
	The Hostage	Sijas			
	An Occurrence at Owl Creek Bridge One of the ...	Kalyani			

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




```

SELECT Category, COUNT(*) AS Total_Books
FROM Books
GROUP BY Category;

```

151

```
152 • SELECT Category, COUNT(*) AS Total_Books
153 FROM Books
154 GROUP BY Category;
```




Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 		
	Category	Total_Books
▶	Adventure	1
	Classic	3
	Crime	2
	Comedy	1
	Adventure	1
	Modern Times	1

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

```
SELECT Emp_name, Position_area
FROM Employe
WHERE Salary > 50000;
```

155

```
156 • SELECT Emp_name, Position_area
157 FROM Employe
158 WHERE Salary > 50000;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 		
	Emp_name	Position_area
▶	Basheer-Uv	Manager
	Maimoona_nazar	Asst_Manager
	Luca	Childrens_librarian
	Faizal	Teeans_manager
	Aslam	Branch_coordinator
	Kiran	Classsic_librarian
	Jaffer	Branch_manager

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

```
SELECT Customer_Name
FROM Customer
WHERE Reg_date < '2022-01-01'
AND Customer_id NOT IN (SELECT Issued_custid FROM IssueStatus);
```

The screenshot displays a database management interface. At the top, a SQL query is entered in a text area:

```
160 • SELECT Customer_Name
161 FROM Customer
162 WHERE Reg_date < '2022-01-01'
163 AND Customer_id NOT IN (SELECT Issued_custid FROM IssueStatus);
```

Below the query editor, there is a toolbar with options like "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content". A dropdown menu shows "Customer_Name".

At the bottom, the "Output" pane is visible, showing a table of execution results:

#	Time	Action	Message
✓ 35	01:36:15	SELECT Category, COUNT(*) AS Total_Books FROM Books GROUP BY Category LIMIT 0, 1000	6 row(s) returned
✓ 36	01:38:42	select*from Employee LIMIT 0, 1000	10 row(s) returned
✓ 37	01:38:59	SELECT Emp_name, Position_area FROM Employee WHERE Salary > 50000 LIMIT 0, 1000	7 row(s) returned
✓ 38	01:41:50	SELECT Customer_Name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_id NOT IN (SEL...	0 row(s) returned
✓ 39	01:42:08	SELECT Customer_Name FROM Customer WHERE Reg_date > '2022-01-01' AND Customer_id NOT IN (SEL...	3 row(s) returned
✓ 40	01:43:04	SELECT Customer_Name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_id NOT IN (SEL...	0 row(s) returned

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

```
SELECT Branch_no, COUNT(*) AS Total_Employs
FROM Employee
GROUP BY Branch_no;
```

```

165 • SELECT Branch_no, COUNT(*) AS Total_Employs
166 FROM Employee
167 GROUP BY Branch_no;

```

Result Grid		
	Filter Rows:	Export: Wrap Cell Content:
	Branch_no	Total_Employs
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1

Result 18			
Output			
Action Output			
#	Time	Action	
36	01:38:42	select*from Employee LIMIT 0, 1000	
37	01:38:59	SELECT Emp_name, Position_area FROM Employee WHERE Salary > 50000 LIMIT 0, 1000	
38	01:41:50	SELECT Customer_Name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_id NOT IN (SEL...	
39	01:42:08	SELECT Customer_Name FROM Customer WHERE Reg_date > '2022-01-01' AND Customer_id NOT IN (SEL...	
40	01:43:04	SELECT Customer_Name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_id NOT IN (SEL...	
41	01:45:53	SELECT Branch_no, COUNT(*) AS Total_Employs FROM Employee GROUP BY Branch_no LIMIT 0, 1000	

8. Display the names of customers who have issued books in the month of June 2024.

```

SELECT Customer.Customer_Name
FROM Customer
JOIN IssueStatus ON Customer.Customer_id = IssueStatus.Issued_custid
WHERE Issue_date BETWEEN '2024-06-01' AND '2024-06-30';

```

```
169 • SELECT Customer.Customer_Name
170 FROM Customer
171 JOIN IssueStatus ON Customer.Customer_id = IssueStatus.Issued_custid
172 WHERE Issue_date BETWEEN '2024-06-01' AND '2024-06-30';
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	Customer_Name
▶	Kalyani

9. Retrieve book_title from book table containing history.

```
SELECT Book_title
FROM Books
WHERE Book_title LIKE '%history%';
```

174 • `SELECT Book_title`
 175 `FROM Books`
 176 `WHERE Book_title LIKE '%history%';`

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

Book_title

Result Grid
Form Editor
Field Types

Books 23 x | Read Only | Conte

Output

Action Output

#	Time	Action	Message
✓ 41	01:45:53	SELECT Branch_no, COUNT(*) AS Total_Employs FROM Employee GROUP BY Branch_no LIMIT 0, 1000	10 row(s) returned
✓ 42	01:50:47	SELECT Customer.Customer_Name FROM Customer JOIN IssueStatus ON Customer.Customer_id = IssueStat...	0 row(s) returned
✓ 43	01:51:12	SELECT Customer.Customer_Name FROM Customer JOIN IssueStatus ON Customer.Customer_id = IssueStat...	0 row(s) returned
✓ 44	01:51:20	SELECT Customer.Customer_Name FROM Customer JOIN IssueStatus ON Customer.Customer_id = IssueStat...	4 row(s) returned
✓ 45	01:51:55	SELECT Customer.Customer_Name FROM Customer JOIN IssueStatus ON Customer.Customer_id = IssueStat...	1 row(s) returned
✓ 46	01:55:53	SELECT Book_title FROM Books WHERE Book_title LIKE "%history%" LIMIT 0, 1000	0 row(s) returned

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

```
SELECT Branch_no, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Branch_no
HAVING COUNT(*) > 5;
```



```

177
178 • SELECT Branch_no, COUNT(*) AS Total_Employees
179 FROM Employee
180 GROUP BY Branch_no
181 HAVING COUNT(*) > 5;

```

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.

```

SELECT e.Emp_name, b.Address
FROM Employee e
JOIN Branchs b ON e.Emp_Id = b.Manage_ID;

```

```

182
183 • SELECT e.Emp_name, b.Address
184 FROM Employee e
185 JOIN Branchs b ON e.Emp_Id = b.Manage_ID;

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Emp_name	Address		
Basheer-Uv	Dubai-public		
Maimoona_nazar	Al Tawar		
Luca	Al mamzar beach		
Arunjith	Al mankhool		
Faizal	Al rashidiya		
Aslam	DWC knowledge		
Kiran	UMM suquien		
Jaffer	House of wisdom		
Sajna	Khalifa park library		
Basheer-Uv	Al Oqoud		

12. Display the names of customers who have issued books with a rental price higher than Rs. 6.5.

```

SELECT DISTINCT Customer.Customer_Name
FROM Customer
JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_custid
JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
WHERE Books.Rental_Price > 6.5;

```

186

187 • SELECT DISTINCT Customer.Customer_Name

188 FROM Customer

189 JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_custid

190 JOIN Books ON IssueStatus.Isbn_book = Books.ISBN

191 WHERE Books.Rental_Price > 6.5;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Result Grid

Form Editor

Field Types

Result 33 x

Read Only

Context Help

Snippets

Output

Action Output

#	Time	Action	Message
✓ 54	02:10:34	insert into Branchs(Manage_ID,Address,Contact_no) values(1,'Dubai-public',05588789), (2,'Al Tawar',023659),...	10 row(s) affected Records: 10 Duplic
✓ 55	02:10:43	select *from Branchs LIMIT 0, 1000	10 row(s) returned
✓ 56	02:10:56	SELECT e.Emp_name, b.Address FROM Employe e JOIN Branchs b ON e.Emp_Id = b.Manage_ID LIMIT 0, 1...	10 row(s) returned
✓ 57	02:15:43	select *from Books LIMIT 0, 1000	9 row(s) returned
✓ 58	02:16:09	SELECT DISTINCT Customer.Customer_Name FROM Customer JOIN IssueStatus ON Customer.Customer_Id ...	1 row(s) returned
✓ 59	02:16:37	SELECT DISTINCT Customer.Customer_Name FROM Customer JOIN IssueStatus ON Customer.Customer_Id ...	3 row(s) returned

end##