MYSQL Comprehensive Assessment

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

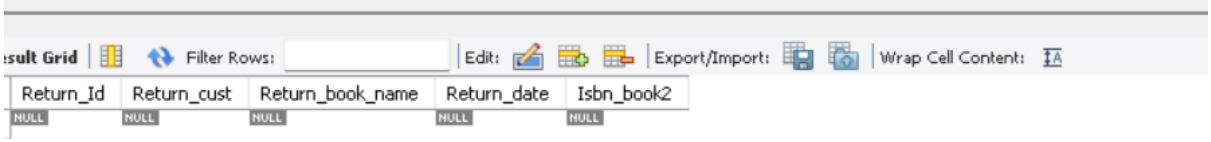
3. Books

4. Customer

5. IssueStatus

6. ReturnStatus

```
create table if not exists  ReturnStatus (
    Return_Id int primary key,
    Return_cust int,
    Return_book_name varchar(255),
    Return_date date,
    Isbn_book2 varchar(20),
    foreign key (Return_cust) references Customer(Customer_Id),
    foreign key (Isbn_book2) references Books(ISBN)
);
select * from ReturnStatus;
```



Attributes for the tables:

1. Branch

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

```sql
create database if not exists library;
use library;

-- create brach table
create table if not exists Branch (
    Branch_no int primary key,
    Manager_Id int,
    Branch_address varchar(255),
    Contact_no varchar(20)
);
```

Output

Action Output

#	Time	Action	Message
1	13:59:11	create database if not exists library	1 row(s) affected
2	14:01:55	use library	0 row(s) affected

```sql
insert into Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
values
(200, 101, '123  Street', '123-456-7890'),
(201, 102, '456  Street', '987-654-3210'),
(202, 103, '789  Street', '555-555-5555'),
(203, 104, '321  Street', '111-222-3333'),
(204, 105, '567  Street', '444-333-2222'),
(205, 107, '234  Street', '999-888-7777'),
(206, 108, '876  Street', '333-222-1111'),
(207, 109, '432  Street', '777-999-1111'),
(208, 110, '678 Street', '111-222-3333');
```

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

Branch_no	Manager_Id	Branch_address	Contact_no
1	101	123 Street	123-456-7890
2	102	456 Street	987-654-3210
3	103	789 Street	555-555-5555
4	104	321 Street	111-222-3333
5	105	567 Street	444-333-2222
200	101	123 Street	123-456-7890
201	102	456 Street	987-654-3210
202	103	789 Street	555-555-5555
203	104	321 Street	111-222-3333

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 Employee table
create table if not exists Employee (
    Emp_Id int primary key,
    Emp_name varchar(100),
    Position varchar(100),
    Salary decimal(10, 2),
    Branch_no int,
    foreign key (Branch_no) references Branch(Branch_no)
);
select * from Employee;
```

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

Emp_Id	Emp_name	Position	Salary	Branch_no
NULL	NULL	NULL	NULL	NULL

```sql
insert into Employee (Emp_Id, Emp_name, Position, Salary, Branch_no)
values
(20, 'Usha', 'Manager', 60000, 200),
(21, 'sannitha k', 'Assistant Manager', 45000, 200),
(22, 'rahul tv', 'Librarian', 40000, 201),
(23, 'lekha ck', 'Clerk', 35000, 202),
(24, 'biju nair', 'Clerk', 30000, 203),
(25, 'shivanand kb', 'Manager', 58000, 204),
(26, 'devakrishnan', 'Assistant Manager', 42000, 205),
(27, 'devananda', 'Librarian', 38000, 206),
(28, 'Sophia ', 'Clerk', 32000, 207),
(29, 'alice', 'Clerk', 28000, 208);
select * from Employee;
```

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

Emp_Id	Emp_name	Position	Salary	Branch_no
20	Usha	Manager	60000.00	200
21	sannitha k	Assistant Manager	45000.00	200
22	rahul tv	Librarian	40000.00	201
23	lekha ck	Clerk	35000.00	202
24	biju nair	Clerk	30000.00	203
25	shivanand kb	Manager	58000.00	204
26	devakrishnan	Assistant Manager	42000.00	205

Employee 5

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 the Books table
create table if not exists Books (
    ISBN varchar(20) primary key,
    Book_title varchar(255),
    Category varchar(100),
    Rental_Price decimal(10, 2),
    status enum('yes', 'no'),
    Author varchar(100),
    Publisher varchar(100)
);
select * from Books;
```

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

ISBN	Book_title	Category	Rental_Price	status	Author	Publisher
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
insert into Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)
values
('978-0143126560', '1984', 'Fiction', 10.99, 'yes', 'George Orwell', 'Penguin Books'),
('978-1400079983', 'The Kite Runner', 'Fiction', 12.99, 'yes', 'Khaled Hosseini', 'Riverhead Books'),
('978-0143038254', 'To Kill a Mockingbird', 'Fiction', 9.99, 'no', 'Harper Lee', 'Harper Perennial'),
('978-0743273565', 'The Catcher in the Rye', 'Fiction', 11.99, 'yes', 'J.D. Salinger', 'Little, Brown and Company'),
('978-0446310789', 'To Kill a Mockingbird', 'Fiction', 9.99, 'yes', 'Harper Lee', 'Harper Perennial'),
('978-0670020553', 'The Goldfinch', 'Fiction', 14.99, 'yes', 'Donna Tartt', 'Little, Brown and Company'),
('978-0385537858', 'All the Light We Cannot See', 'Fiction', 13.99, 'yes', 'Anthony Doerr', 'Scribner'),
('978-0316769174', 'The Bell Jar', 'Fiction', 10.99, 'yes', 'Sylvia Plath', 'Harper Perennial Modern Classics'),
('978-0061120080', 'To Kill a Mockingbird', 'Fiction', 9.99, 'yes', 'Harper Lee', 'Harper Perennial'),
('978-0735219090', 'Where the Crawdads Sing', 'Fiction', 15.99, 'yes', 'Delia Owens', 'G.P. Putnam\'s Sons');
select * from Books;
```

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

ISBN	Book_title	Category	Rental_Price	status	Author	Publisher
978-0061120080	To Kill a Mockingbird	Fiction	9.99	yes	Harper Lee	Harper Perennial
978-0143038254	To Kill a Mockingbird	Fiction	9.99	no	Harper Lee	Harper Perennial
978-0143126560	1984	Fiction	10.99	yes	George Orwell	Penguin Books
978-0316769174	The Bell Jar	Fiction	10.99	yes	Sylvia Plath	Harper Perennial Modern Classics
978-0385537858	All the Light We Cannot See	Fiction	13.99	yes	Anthony Doerr	Scribner
978-0446310789	To Kill a Mockingbird	Fiction	9.99	yes	Harper Lee	Harper Perennial
978-0670020553	The Goldfinch	Fiction	14.99	yes	Donna Tartt	Little, Brown and Company

Books 6

4. Customer

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

```sql
35
36      -- Create  Customer table
37  •  create table if not exists Customer (
38          Customer_Id int primary key,
39          Customer_name varchar(100),
40          Customer_address varchar(255),
41          Reg_date date
42      );
43  •  select * from Customer;
```

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

Customer_Id	Customer_name	Customer_address	Reg_date
NULL	NULL	NULL	NULL

```sql
116 •  insert into Customer (Customer_Id, Customer_name, Customer_address, Reg_date)
117    values
118    (201, 'arun', '1234 Park Ave', '2021-12-15'),
119    (202, 'Sarah ', '5678 Elm St', '2020-11-20'),
120    (203, 'David ', '9876 Pine St', '2022-01-05'),
121    (204, 'nike', '5432 Oak St', '2023-03-10'),
122    (205, 'albart', '8765 Maple St', '2022-05-25'),
123    (206, 'jane', '234 Cedar St', '2022-08-15'),
124    (207, 'Jessica ', '890 Rose St', '2021-06-30');
125 •  select * from Customer;
126
```

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

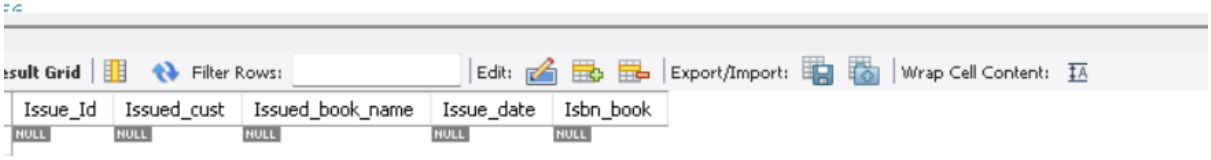
Customer_Id	Customer_name	Customer_address	Reg_date
201	arun	1234 Park Ave	2021-12-15
202	Sarah	5678 Elm St	2020-11-20
203	David	9876 Pine St	2022-01-05
204	nike	5432 Oak St	2023-03-10
205	albart	8765 Maple St	2022-05-25
206	jane	234 Cedar St	2022-08-15
207	Jessica	890 Rose St	2021-06-30

Customer 7

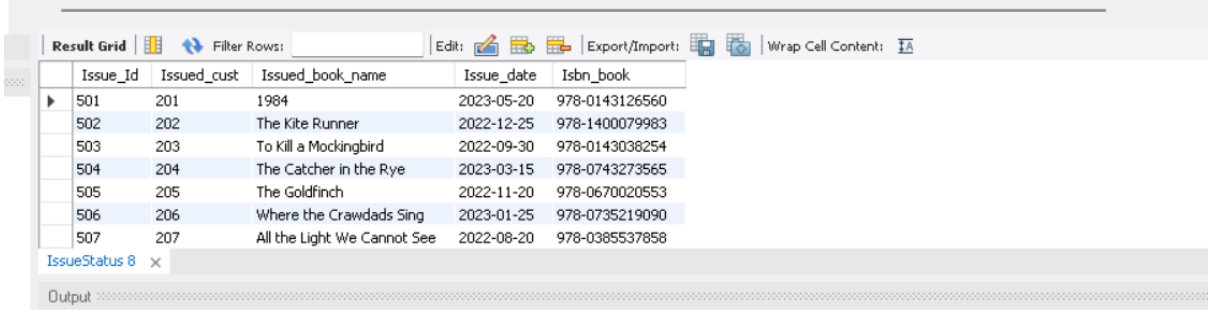
5. IssueStatus

- 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  IssueStatus table
create table if not exists  IssueStatus (
    Issue_Id int primary key,
    Issued_cust int,
    Issued_book_name varchar(255),
    Issue_date date,
    Isbn_book varchar(20),
    foreign key (Issued_cust) references Customer(Customer_Id),
    foreign key (Isbn_book) references Books(ISBN)
);
select * from IssueStatus;
```




```sql
insert into  IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book)
values
(501, 201, '1984', '2023-05-20', '978-0143126560'),
(502, 202, 'The Kite Runner', '2022-12-25', '978-1400079983'),
(503, 203, 'To Kill a Mockingbird', '2022-09-30', '978-0143038254'),
(504, 204, 'The Catcher in the Rye', '2023-03-15', '978-0743273565'),
(505, 205, 'The Goldfinch', '2022-11-20', '978-0670020553'),
(506, 206, 'Where the Crawdads Sing', '2023-01-25', '978-0735219090'),
(507, 207, 'All the Light We Cannot See', '2022-08-20', '978-0385537858'),
(508, 201, 'The Bell Jar', '2023-04-30', '978-0316769174'),
(509, 202, 'To Kill a Mockingbird', '2022-11-05', '978-0061120080'),
(510, 203, 'To Kill a Mockingbird', '2023-04-10', '978-0061120080');
select * from IssueStatus;
```



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 if not exists  ReturnStatus (
    Return_Id int primary key,
    Return_cust int,
    Return_book_name varchar(255),
    Return_date date,
    Isbn_book2 varchar(20),
    foreign key (Return_cust) references Customer(Customer_Id),
    foreign key (Isbn_book2) references Books(ISBN)
);
select * from ReturnStatus;
```

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

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
NULL	NULL	NULL	NULL	NULL

```sql
insert into ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)
values
(601, 201, '1984', '2023-06-10', '978-0143126560'),
(602, 202, 'The Kite Runner', '2023-01-05', '978-1400079983'),
(603, 203, 'To Kill a Mockingbird', '2022-10-10', '978-0143038254'),
(604, 204, 'The Catcher in the Rye', '2023-04-05', '978-0743273565'),
(605, 205, 'The Goldfinch', '2023-01-30', '978-0670020553'),
(606, 206, 'Where the Crawdads Sing', '2023-02-05', '978-0735219090'),
(607, 207, 'All the Light We Cannot See', '2022-09-05', '978-0385537858'),
(608, 201, 'The Bell Jar', '2023-05-15', '978-0316769174'),
(609, 202, 'To Kill a Mockingbird', '2023-01-20', '978-0061120080'),
(610, 203, 'To Kill a Mockingbird', '2023-05-25', '978-0061120080');
select * from ReturnStatus;
```

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

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
601	201	1984	2023-06-10	978-0143126560
602	202	The Kite Runner	2023-01-05	978-1400079983
603	203	To Kill a Mockingbird	2022-10-10	978-0143038254
604	204	The Catcher in the Rye	2023-04-05	978-0743273565
605	205	The Goldfinch	2023-01-30	978-0670020553
606	206	Where the Crawdads Sing	2023-02-05	978-0735219090
607	207	All the Light We Cannot See	2022-09-05	978-0385537858

ReturnStatus 9

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

```
show tables;
```

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

Tables_in_library
books
branch
customer
employee
issuestatus
returnstatus

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

```
127
128     #1
129
130 •   select Book_title, Category, Rental_Price
131     from Books
132     where status = 'yes';
133
134
```

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

Book_title	Category	Rental_Price
All the Light We Cannot See	Fiction	13.99
To Kill a Mockingbird	Fiction	9.99
The Goldfinch	Fiction	14.99
Where the Crawdads Sing	Fiction	15.99
The Catcher in the Rye	Fiction	11.99
The Kite Runner	Fiction	12.99

Books 3

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

```
163
164     #2
165
166 •   select emp_name, salary
167     from employee
168     order by salary desc;
```

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

emp_name	salary
Usha	60000.00
shivanand kb	58000.00
sannitha k	45000.00
devakrishnan	42000.00
rahul tv	40000.00
devananda	38000.00
lekha ck	35000.00

employee 10

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

```
170     #3
171 •   select b.book_title, c.customer_name
172     from issuestatus i
173     join books b on i.isbn_book = b.isbn
174     join customer c on i.issued_cust = c.customer_id;
```

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

book_title	customer_name
1984	arun
The Kite Runner	Sarah
To Kill a Mockingbird	David
The Catcher in the Rye	nike
The Goldfinch	albart
Where the Crawdads Sing	jane
All the Light We Cannot See	Jessica

Result 11

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

```
174     join customer c on i.issued_cust = c.customer_id;
175
176     #4
177 •   select Category, COUNT(*) as Total_Count
178     from Books
179     group by Category;
180
181
182
183
184
185
```

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

Category	Total_Count
Fiction	10

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

```
181
182     #5
183 •   select emp_name, position
184     from employee
185     where salary > 50000;
```

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

emp_name	position
Usha	Manager
shivanand kb	Manager

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

```
187     #6
188 •   select customer_name
189     from customer
190     where reg_date < '2022-01-01'
191     and customer_id not in (select issued_cust from issuestatus);
```

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

customer_name

(Here no customer registered before 2022-01-01)

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

```
193     #7
194
195 •   select branch_no, count(*) as total_employees
196     from employee
197     group by branch_no;
```

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

branch_no	total_employees
200	2
201	1
202	1
203	1
204	1
205	1
206	1

Result 17

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

```
198
199     #8
200 •   select distinct c.customer_name
201     from customer c
202     join issuestatus i on c.customer_id = i.issued_cust
203     where year(i.issue_date) = 2023 and month(i.issue_date) = 6;
```

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

customer_name

9. Retrieve book_title from book table containing history.

```
205     #9
206 •   insert into Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)
207     values('978-0735219097',' Gandhi','history',10.99, 'yes', 'Mahatma Gandhi.', 'Penguin Books');
208 •   select book_title
209     from books
210     where category = 'history';
```

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

book_title
Gandhi

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

```
212     #10
213 •   select branch_no, count(*) as total_employees
214     from employee
215     group by branch_no
216     having count(*) > 5;
217
218
219
220
221
222
223
```

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.

```
218     #11
219 •   select e.emp_name, b.branch_address
220     from employee e
221     join branch b on e.branch_no = b.branch_no
222     where e.position = 'manager';
223
```

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

emp_name	branch_address
Usha	123 Street
shivanand kb	567 Street

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

```
224    #12
225 •  select distinct c.customer_name
226    from customer c
227    join issuestatus i on c.customer_id = i.issued_cust
228    join books b on i.isbn_book = b.isbn
229    where b.rental_price > 25;
```

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

customer_name