Module End Project

DSML D19
VISHNU E P

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
5. 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

```
#LIBRARY MANAGEMENT SYSTEM
create database LMS;
USE LMS;

create table Branch(Branch_no int primary key,
Manager_id int,
Branch_address varchar(30),
contact_no int);
desc branch;
drop table branch;
insert into branch values (1,001,'mlptm branch',987456321);
insert into branch values (2,002,'mlptm branch',987456322);
insert into branch values (3,003,'mlptm branch',987456323);
insert into branch values (4,004,'mlptm branch',987456324);
insert into branch values (5,005,'mlptm branch',987456325);
select * from branch;

```

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

Branch_no	Manager_id	Branch_address	contact_no
1	1	mlptm branch	987456321
2	2	mlptm branch	987456322
3	3	mlptm branch	987456323
4	4	mlptm branch	987456324
5	5	mlptm branch	987456325
NULL	NULL	NULL	NULL

```
create table employee(emp_id int primary key,
emp_name varchar(30),
position varchar(30),
salary int,
branch_no int,
foreign key(branch_no) references Branch(Branch_no));
desc employee;
insert into employee values(101,'abhinav','lib assistant',17000,1);
insert into employee values(102,'abhijith','lib manager',53000,2);
insert into employee values(103,'abhiraj','librerian',20000,3);
insert into employee values(104,'abhinand','lib assistant',15000,4);
insert into employee values(105,'abhina','lib manager',45000,5);
insert into employee values(106,'pranav','librerian',17000,4);
insert into employee values(107,'naveen','lib manager',41000,2);
insert into employee values(108,'avishna','lib assistant',20500,3);
insert into employee values(109,'vishnu','lib manager',59000,4);
insert into employee values(110,'nandhana','lib manager',51000,1);
select * from employee;
```

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

emp_id	emp_name	position	salary	branch_no
101	abhinav	lib assistant	17000	1
102	abhijith	lib manager	53000	2
103	abhiraj	librerian	20000	3
104	abhinand	lib assistant	15000	4
105	abhina	lib manager	45000	5
106	pranav	librerian	17000	4
107	naveen	lib manager	41000	2
108	avishna	lib assistant	20500	3

employee 2 ×

```sql
create table Books(ISBN int PRIMARY KEY,
Book_title varchar(30),
Category varchar(28),
Rental_price float,
status char(5),
Author varchar(50),
Publisher varchar(30));
desc books;
insert into books values(9754221, 'Harry Potter', 'Fantasy', 599, 'yes', 'J.K. Rowling', 'Scholastic'),
    (9780930, 'To Kill a Mockingbird', 'classic', 499, 'yes', 'Harper lee', 'Scholastic'),
    (9780439, 'Adujeevitham', 'Fantasy', 499, 'yes', 'benniyamin', 'DC books'),
    (9780839, 'The Great Gatsby', 'Fantasy', 399, 'yes', ' F. Scott Fitzgerald ', 'Scribner'),
    (9780465, '1984', 'Fantasy', 199, 'no', 'George Orwell', 'Secker & Warburg'),
    (9780071, 'The Catcher in the Rye', 'Fiction', 299, 'yes', 'J.D. Salinger', ' Little Brown and Company'),
    (9749700, 'The Da Vinci Code', 'Fiction', 199, 'no', 'Dan Brown', 'Ss books'),
    (9783427, 'A Brief History of Humankind', 'Non Fiction', 299, 'yes', ' Yuval Noah Harar', 'Bloomsbury'),
    (9784735, 'wings of fire', 'autobiography', 499, 'yes', 'APJ abdul kalam', 'DC books'),
    (9781483, 'Alchemist', 'Fiction', 699, 'yes', 'paulo coelho', 'Bloomsbury');
select * from books;
```

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

ISBN	Book_title	Category	Rental_price	status	Author	Publisher
9749700	The Da Vinci Code	Fiction	199	no	Dan Brown	Ss books
9754221	Harry Potter	Fantasy	599	yes	J.K. Rowling	Scholastic
9780071	The Catcher in the Rye	Fiction	299	yes	J.D. Salinger	Little Brown and Company
9780439	Adujeevitham	Fantasy	499	yes	benniyamin	DC books
9780465	1984	Fantasy	199	no	George Orwell	Secker & Warburg
9780839	The Great Gatsby	Fantasy	399	yes	F. Scott Fitzgerald	Scribner
9780930	To Kill a Mockingbird	classic	499	yes	Harper lee	Scholastic
9781483	Alchemist	Fiction	699	yes	paulo coelho	Bloomsbury
9783427	A Brief History of Humankind	Non Fiction	299	yes	Yuval Noah Harar	Bloomsbury
9784735	wings of fire	autobiography	499	yes	APJ abdul kalam	DC books

books 3

```sql
create table customer(customer_id int primary key,
customer_name varchar(25),
customer_address varchar(30),
Reg_date date);
desc customer;

insert into customer values(1001,'vishnu','alavoor house','2021-06-11'),
(1002,'shanu','deva house','2022-3-21'),(1003,'vishak','pp house','2023-09-14'),
(1004,'anu','ram house','2024-04-19'),(1005,'vinod','mr house','2022-10-24');
select * from customer;
```

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

customer_id	customer_name	customer_address	Reg_date
1001	vishnu	alavoor house	2021-06-11
1002	shanu	deva house	2022-03-21
1003	vishak	pp house	2023-09-14
1004	anu	ram house	2024-04-19
1005	vinod	mr house	2022-10-24
NULL	NULL	NULL	NULL

```sql
create table IssueStatus(issue_id int primary key,
issued_cus int,
foreign key (issued_cus) references customer(customer_id),
issued_date date,
Isbn_book int,foreign key (Isbn_book) references Books(ISBN));
desc Issuestatus;

insert into IssueStatus values(10001,1001,'2020-2-14',9754221),
(10002,1002,'2024-05-22',9754221),(10003,1003,'2024-07-20',9780071),
(10004,1004,'2023-05-12',9780439),(10005,1005,'2024-05-22',9780465);
select * from Issuestatus;
```

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

issue_id	issued_cus	issued_date	Isbn_book
10001	1001	2020-02-14	9754221
10002	1002	2024-05-22	9754221
10003	1003	2024-07-20	9780071
10004	1004	2023-05-12	9780439
10005	1005	2024-05-22	9780465
NULL	NULL	NULL	NULL

```sql
create table ReturnStatus(Return_id int primary key,
Return_cus varchar(25),
Return_book_name varchar(30),
Return_date date,
Isbn_book2 int,foreign key (Isbn_book2) references Books(ISBN));
desc ReturnStatus;
drop table returnstatus;
insert into ReturnStatus values(01,'vishnu','Harry Potter','2020-01-13',9754221),
(02,'anu','Adujeevitham','2024-04-29',9780439);
select * from ReturnStatus;

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

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

Return_id	Return_cus	Return_book_name	Return_date	Isbn_book2
1	vishnu	Harry Potter	2020-01-13	9754221
2	anu	Adujeevitham	2024-04-29	9780439
NULL	NULL	NULL	NULL	NULL

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;
```

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

book_title	category	rental_price
The Da Vinci Code	Fiction	199
Harry Potter	Fantasy	599
The Catcher in the Rye	Fiction	299
Adujeevitham	Fantasy	499
1984	Fantasy	199
The Great Gatsby	Fantasy	399
To Kill a Mockingbird	classic	499
Alchemist	Fiction	699
A Brief History of Humankind	Non Fiction	299
wings of fire	autobiography	499

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

```
95      #2. List the employee names and their respective salaries in descending order of salary.
96 •    select emp_name,salary from employee order by salary desc;
97
98      #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
vishnu	59000
abhijith	53000
nandhana	51000
abhina	45000
naveen	41000
avishna	20500
abhiraj	20000
abhinav	17000
pranav	17000
abhinand	15000

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

```
98      #3. Retrieve the book titles and the corresponding customers who have issued those books.
99 •    select b.book_title,c.customer_name,i.issued_date
100     from issuestatus i join books b on b.isbn=i.isbn_book
101     join customer c on i.issued_cus=c.customer_id;
102
103     #4. Display the total count of books in each category.
104 •    select category,count(book_title) from books group by category;
```

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

book_title	customer_name	issued_date
Harry Potter	vishnu	2020-02-14
Harry Potter	shanu	2024-05-22
The Catcher in the Rye	vishak	2024-07-20
Adujeevitham	anu	2023-05-12
1984	vinod	2024-05-22

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

```
103     #4. Display the total count of books in each category.
104 •    select category,count(book_title) from books group by category;
105
106     #5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
107 •    select emp_name,salary,position from employee where salary>50000;
```

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

category	count(book_title)
Fiction	3
Fantasy	4
classic	1
Non Fiction	1
autobiography	1

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

```
105
106     #5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
107 •    select emp_name,salary,position from employee where salary>50000;
```

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

emp_name	salary	position
abhijith	53000	lib manager
vishnu	59000	lib manager
nandhana	51000	lib manager

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

```
#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_cus from issuestatus);
```

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

customer_name

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

```
#7. Display the branch numbers and the total count of employees in each branch.
select branch_no,count(emp_name) as totalEmployees from employee group by branch_no;
```

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

branch_no	totalEmployees
1	2
2	2
3	2
4	3
5	1

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

```
#8. Display the names of customers who have issued books in the month of June 2023.
select issued_cus from issuestatus where issued_date in ('2023-06-01');
```

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

issued_cus

9. Retrieve book_title from book table containing history.

```
#9. Retrieve book_title from book table containing history.
select book_title,category from books where category='history';
```

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

book_title	category

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

```
#10.Retrieve the branch numbers along with the count of employees for branches having more than 3 employees
select branch_no,count(emp_name) as TotleEmployees from employee group by branch_no having count(emp_name)>=3;
```

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

branch_no	TotleEmployees
4	3