Library management system using MySQL

Topic:

Library management system using MySQL- Develop a system to manage library resources,track book availability,and monitor transactions.-Key features include data retrieval by category automated issue/return tracking, and employee distribution monitoring

Introduction

The Library Management System (LMS) using MySQL is a database-driven project aimed at providing a streamlined, automated solution for managing library resources and operations. Traditional library management can be challenging due to the need for meticulous record-keeping and frequent updates for book inventory, member details, and transactions. This project addresses these challenges by creating a digital system powered by MySQL, a widely-used relational database management system known for its reliability and efficiency in handling large datasets.

In this system, MySQL serves as the core for storing and managing data related to books, members, and loan transactions. The LMS allows library staff to easily add, update, and track information, such as book availability, member details, and borrowing history. Members can benefit from features like book search and reservation, making their library experience more convenient and engaging.

Overall, the Library Management System using MySQL not only simplifies administrative tasks for library staff but also enhances the accessibility and efficiency of library services for users. This system is designed to modernize library management, minimize human errors, and support a more organized and user-friendly library environment.

Key Features:

1. **Book Management**
   - Add, update, delete books.
   - Search and retrieve books by category (e.g., author, genre, year, etc.).
2. **Automated Issue/Return System**
   - Track books that are issued and returned by members.
   - Automated due date calculation and fine system for late returns.
3. **Member Management**
   - Add and update member details.
   - Track each member's borrowing history.
4. **Employee Monitoring**
   - Track employees responsible for book issuance, return handling, and other tasks.
5. **Report Generation**
   - Generate reports on available books, borrowed books, late returns, and employee activity.

Database Design:

1. Books Table

- book_id (INT, Primary Key, Auto Increment)
- title (VARCHAR)
- author (VARCHAR)
- genre (VARCHAR)
- publisher (VARCHAR)
- publication_year (YEAR)
- status (ENUM: Available, Borrowed)

2. Members Table

- member_id (INT, Primary Key, Auto Increment)
- name (VARCHAR)
- address (VARCHAR)
- phone (VARCHAR)
- email (VARCHAR)

3. Employees Table

- employee_id (INT, Primary Key, Auto Increment)
- name (VARCHAR)
- role (ENUM: Librarian, Clerk)

5. Categories Table

- category_id (INT, Primary Key, Auto Increment)
- category_name (VARCHAR)
- description (TEXT)

4. Transactions Table

- transaction_id (INT, Primary Key, Auto Increment)
- book_id (INT, Foreign Key)
- member_id (INT, Foreign Key)
- employee_id (INT, Foreign Key)
- issue_date (DATE)
- return_date (DATE)
- status (ENUM: Issued, Returned)

Tables

Books:

book_id	title	author	genre	publisher	publication_year	status
1	The Silent Patient	Alex Michaelides	Thriller	Celadon Books	2019	Borrowed
2	Educated	Tara Westover	Memoir	Random House	2018	Borrowed
3	The Alchemist	Paulo Coelho	Adventure	HarperOne	1988	Available
4	Sapiens	Yuval Noah Harari	History	Harper	2011	Available
5	Becoming	Michelle Obama	Biography	Crown Publishing	2018	Borrowed
6	The Midnight Library	Matt Haig	Fantasy	Canongate Books	2020	Available
7	The Road	Cormac McCarthy	Post-apocalyptic	Alfred A. Knopf	2006	Borrowed
8	Where the Crawdads Sing	Delia Owens	Mystery	G.P. Putnam's Sons	2018	Available
9	The Subtle Art of Not Giving a F*ck	Mark Manson	Self-help	HarperOne	2016	Available
10	The Catcher in the Rye	J.D. Salinger	Fiction	Little, Brown and Company	1951	Borrowed
NULL	NULL	NULL	NULL	NULL	NULL	NULL

categories:

category_id	category_name	description
1	Fiction	Narrative literature created from the imaginatio...
2	Non-Fiction	Literature based on facts, real events, and real...
3	Mystery	Fictional works that involve a suspenseful plot, ...
4	Science Fiction	Fiction that explores futuristic concepts, advan...
5	Fantasy	Fiction that features magical elements and imagi...
6	Biography	A detailed description of a person's life written ...
7	Self-Help	Books designed to help readers improve their liv...
8	History	Literature that examines past events and their ...
9	Poetry	A literary form that uses rhythmic and aesthetic...
10	Cookbooks	Books containing recipes and cooking tips.
NULL	NULL	NULL

Employees:

employee_id	name	role
1	Sarah Thompson	Librarian
2	David Miller	Clerk
3	Emily Johnson	Librarian
4	James Smith	Clerk
5	Laura Brown	Librarian
6	Michael Wilson	Clerk
7	Emma Davis	Librarian
8	Robert Garcia	Clerk
9	Sophia Martinez	Librarian
10	William Lee	Clerk
NULL	NULL	NULL

Members:

member_id	name	address	phone	email
1	John Doe Jr.	456 Oak Street	555-5678	john.doe.jr@example.com
2	Jane Smith	456 Oak Ave, Lincoln, NE	555-987-6543	janesmith@example.com
3	Michael Johnson	789 Pine Rd, Austin, TX	555-456-7890	michaelj@example.com
4	Emily Davis	321 Maple St, Denver, CO	555-789-1234	emilyd@example.com
5	William Brown	654 Cedar Ln, Seattle, WA	555-321-9876	williamb@example.com
6	Olivia Wilson	876 Birch Dr, Boston, MA	555-654-3210	oliviaw@example.com
7	Sophia Miller	987 Elm St, Portland, OR	555-789-4561	sophiam@example.com
8	James Taylor	543 Walnut St, Phoenix, AZ	555-123-9870	jamest@example.com
9	Charlotte Anderson	123 Cherry Ln, Miami, FL	555-654-7890	charlottea@example.com
10	Liam Martinez	456 Spruce St, Chicago, IL	555-321-6547	liamm@example.com
NULL	NULL	NULL	NULL	NULL

Transactions:

transaction_id	book_id	member_id	employee_id	issue_date	return_date	status
1	1	1	1	2024-01-15	NULL	Issued
2	2	2	2	2024-02-10	2024-02-20	Returned
3	3	3	1	2024-03-05	NULL	Issued
4	4	4	3	2024-03-15	2024-03-25	Returned
5	5	5	2	2024-04-01	NULL	Issued
6	6	6	4	2024-04-10	2024-04-17	Returned
7	7	7	1	2024-05-01	NULL	Issued
8	8	8	3	2024-05-15	NULL	Issued
9	9	9	2	2024-06-05	2024-06-15	Returned
10	10	10	4	2024-06-20	NULL	Issued
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database Design

create tables in MySQL for each entity (Books, Users, Transactions, etc.). Here's the basic structure

Books Table:

```
create table Books(book_id INT Primary Key Auto_Increment,
title VARCHAR(100),
author VARCHAR(100),
genre VARCHAR(100),
publisher VARCHAR(100),
publication_year YEAR,
status ENUM('Available', 'Borrowed'
);
```

MEMBERS Table:

```
create table members(member_id INT Primary Key Auto_Increment,
name VARCHAR(100),
address text,
phone VARCHAR(15),
email VARCHAR(40)
);
```

EMPLOYEES Table:

```
create table employees(employee_id INT Primary
Key Auto_Increment,
name VARCHAR(100),
role ENUM ('Librarian', 'Clerk')
);
```

Transactions Table:

```
create table transactions(transaction_id INT Primary Key
Auto_Increment,
book_id INT,
member_id INT,
employee_id INT,
issue_date DATE,
return_date DATE,
status ENUM ('Issued', 'Returned'),
foreign key(book_id) references books(book_id),
foreign key(member_id) references members(member_id),
foreign key(employee_id) references employees(employee_id)
);
```

CATEGORIES Table :

```
create table categories(category_id INT Primary Key Auto_Increment,
category_name VARCHAR(100),
description TEXT
);
```

BASIC QUERIES

- Add a New Book:

```
insert into  books(title, author, genre, publisher, publication_year, status) values('The Great Gatsby', 'F. Scott Fitzgerald',' Literary Fiction','Charles Scribners Sons',1925,  'Available' );
```

- Issue a Book:

```
update books set status='borrowed'where book_id=1 ;
```

- SEARCH BOOKS:

```
select title, author, genre, publisher, publication_year from books where status='available';
```

- Track books that are issued and returned by members:

```
select
m.name AS member_name,
```

```
b.title AS book_title,
t.issue_date,
t.return_date,
t.status
from transactions as t inner join books as b on t.book_id=b.book_id
inner join members as m on t.member_id = m.member_id
order by t.issue_date desc;
```

- Automated due date calculation and fine system for late returns:

```
SELECT
m.name AS member_name,
b.title AS book_title,
t.issue_date,
DATE_ADD(t.issue_date, INTERVAL 14 DAY) AS due_date,  -- Calculates due date 14 days after issue
date
t.return_date,
CASE
WHEN t.return_date IS NULL AND CURDATE() > DATE_ADD(t.issue_date, INTERVAL 14 DAY) THEN
DATEDIFF(CURDATE(), DATE_ADD(t.issue_date, INTERVAL 14 DAY)) * 1  -- Fine calculation: $1 for each
overdue day
WHEN t.return_date > DATE_ADD(t.issue_date, INTERVAL 14 DAY) THEN
DATEDIFF(t.return_date, DATE_ADD(t.issue_date, INTERVAL 14 DAY)) * 1  -- Fine if returned late
ELSE 0
END AS fine_amount
FROM
transactions t
JOIN
books as b ON t.book_id = b.book_id
JOIN
members as m ON t.member_id = m.member_id
WHERE
t.status = 'Issued' OR t.status = 'Returned';
```

- Updating an Existing Member's Details:

```
UPDATE members
SET
name = 'John Doe Jr.',
address = '456 Oak Street',
phone = '555-5678',
email = 'john.doe.jr@example.com'
WHERE
member_id = 1;
select * from members;
```

- Query to Track Member’s Borrowing History

```
SELECT
m.name AS member_name,
b.title AS book_title,
t.issue_date,
t.return_date,
t.status
FROM
```

```
transactions  as t
JOIN
books as  b ON t.book_id = b.book_id
JOIN
members as m ON t.member_id = m.member_id
WHERE
m.member_id =4   -- Replace with the actual member's ID
ORDER BY
t.issue_date DESC;
```

- Track employees responsible for book issuance, return handling, and other tasks:

```
SELECT
   e.name AS employee_name,
   b.title AS book_title,
   t.issue_date,
   t.return_date,
   t.status,
   t.transaction_id
FROM
   transactions t
JOIN
   books  as b ON t.book_id = b.book_id
JOIN
   employees as  e ON t.employee_id = e.employee_id
ORDER BY
   t.issue_date DESC;
```

- Report on Borrowed Books

```
SELECT
   b.book_id, b.title, b.author, t.issue_date, m.name AS borrowed_by
FROM
   books as  b
JOIN
   transactions as  t ON b.book_id = t.book_id
JOIN
   members as  m ON t.member_id = m.member_id
WHERE
   t.status = 'Issued';
```

- Report on Late Returns

```
SELECT
   m.name AS member_name,
   b.title AS book_title,
   t.issue_date,
   DATE_ADD(t.issue_date, INTERVAL 14 DAY) AS due_date,
   CURDATE() AS `current_date`,  -- Escaped the alias `current_date`
   DATEDIFF(CURDATE(), DATE_ADD(t.issue_date, INTERVAL 14 DAY)) AS days_late
FROM
   transactions t
JOIN
   books b ON t.book_id = b.book_id
JOIN
```

```
    members m ON t.member_id = m.member_id
WHERE
    t.return_date IS NULL
    AND CURDATE() > DATE_ADD(t.issue_date, INTERVAL 14 DAY);
```

- Report on Employee Activity

```
SELECT
    e.name AS employee_name,
    COUNT(t.transaction_id) AS total_transactions_handled
FROM
    employees e
LEFT JOIN
    transactions t ON e.employee_id = t.employee_id
GROUP BY
    e.employee_id;
```

conclusion

Library Management System (LMS) using MySQL effectively modernizes and simplifies the process of managing a library's resources. By leveraging MySQL as the backbone database, the system provides secure, organized, and efficient data management for books, members, and transactions. This project demonstrates how database management systems can transform traditional operations, minimizing manual effort, reducing human errors, and enhancing the user experience for both library staff and patrons.

The LMS fulfills the goal of creating a centralized, easily accessible platform that can manage library tasks like inventory tracking, loan management, and report generation with accuracy and efficiency. This system not only improves administrative workflow but also empowers library users by offering quick access to information, thus fostering a more engaging and productive library environment. Overall, this project highlights the value of technology in creating structured, effective solutions for managing library operations.