

Project Title: Library Management System (LMS)

Project Duration - 1 months

Project Overview:

The Library Management System (LMS) is a MySQL-based project that manages library resources such as books, members, borrowing records, and staff details. It incorporates relational database concepts, DML/DDL operations, and advanced SQL features like views, stored procedures, triggers, and transactions.

Key Features:

Database Design

- A normalized schema for efficient storage and retrieval.
- Create Tables for books, members, staff, borrow records, and book categories.

Data Integrity

- Constraints like FOREIGN KEY, CHECK, and UNIQUE.
- Triggers for consistent updates (e.g., decrementing book stock on borrowing).

CRUD Operations

- End-to-end operations for adding, updating, and deleting books, members, and borrowing records.

Advanced SQL

- Use of stored procedures for routine tasks.
- Views for simplified reporting.
- Transactions
- Ensures atomic operations for borrowing and returning books.
- Reports
- Aggregated insights like most borrowed books or overdue fines.

Project Task -

Library Management System (LMS) Project Tasks

1. Database Design and Setup

- Design the database schema for the LMS.
- Create tables for Books, Members, Staff, Borrowing Records, and Categories.
- Implement appropriate relationships between tables using Foreign Keys.
- Ensure data integrity by setting up Constraints like UNIQUE, NOT NULL, CHECK, and DEFAULT.
- Insert sample data into the tables.
- Populate the Books table with sample book details (e.g., title, author, category, stock).
- Add sample Members (e.g., student, staff), Staff details, and Borrowing Records for testing.

2. CRUD Operations

- Implement CRUD operations for the Books table.
- Add functionality to Add, Update, Delete, and View book details.
- Implement CRUD operations for the Members table.
- Add functionality for managing member records (Add, Edit, Delete, View).
- Implement CRUD operations for the Borrowing Records table.

3. Additional Features -

- Add functionality for borrowing and returning books (issue a book, update borrowing records when a book is returned).
- Book Borrowing and Returning Logic
- Implement book borrowing functionality.
- When a member borrows a book, decrement the book stock in the Books table.
- If the book stock is zero, prevent further borrowing.
- Implement book returning functionality.

- When a book is returned, increment the book stock and update the Borrowing Records table.
- Reporting and Insights
 - Generate a report of most borrowed books.
 - Use aggregate functions like COUNT and GROUP BY to identify the most borrowed books.
 - Generate a report of overdue fines.
- Calculate fines based on the overdue period and generate a report for overdue books.

4. Advanced SQL Features

- Create a View for simplified access to borrowing records.
- Create a view that joins Books, Members, and Borrowing Records for easy reporting of book borrowings.
- Implement Stored Procedures for routine tasks.

5. Create a stored procedure to add a new book

- Create a stored procedure to update a member's information.
- Implement Triggers for automatic updates.
- Create a trigger to automatically update the stock when a book is borrowed or returned.

6. Transaction Management

- Implement transactions for borrowing and returning books.
- Ensure atomic operations to maintain data consistency, using BEGIN TRANSACTION, COMMIT, and ROLLBACK.

7. Documentation

- Write a README file.
- Document the steps to set up and run the project.
- Include a section for database setup and instructions for running stored procedures, views, and triggers.
- Provide example queries.
- Include example SQL queries for common operations like borrowing a book, returning a book, and generating reports.

Table

Categories (CategoryID, CategoryName)

(1, 'Fiction'),
(2, 'Non-Fiction'),
(3, 'Science'),
(4, 'Mathematics'),
(5, 'History'),
(6, 'Biographies'),
(7, 'Philosophy'),
(8, 'Technology'),
(9, 'Psychology'),
(10, 'Literature'),
(11, 'Business'),
(12, 'Art'),
(13, 'Politics'),
(14, 'Travel'),
(15, 'Cooking'),
(16, 'Religion'),
(17, 'Self-Help'),
(18, 'Health'),
(19, 'Education'),
(20, 'Sports');

Books (BookID, Title, Author, CategoryID, Stock)

(1, 'Wings of Fire', 'A.P.J. Abdul Kalam', 6, 10),
(2, 'The Alchemist', 'Paulo Coelho', 1, 8),
(3, 'Sapiens', 'Yuval Noah Harari', 2, 12),
(4, 'Brief History of Time', 'Stephen Hawking', 3, 7),
(5, 'Ramayana', 'Valmiki', 16, 15),
(6, 'The Great Indian Novel', 'Shashi Tharoor', 1, 9),

(7, 'India After Gandhi', 'Ramachandra Guha', 5, 6),
(8, 'Think and Grow Rich', 'Napoleon Hill', 11, 14),
(9, 'The Art of War', 'Sun Tzu', 7, 10),
(10, 'Rich Dad Poor Dad', 'Robert Kiyosaki', 11, 8),
(11, 'The Monk Who Sold His Ferrari', 'Robin Sharma', 17, 11),
(12, 'You Can Win', 'Shiv Khera', 17, 13),
(13, 'Cooking for Beginners', 'Tarla Dalal', 15, 5),
(14, 'The Blue Umbrella', 'Ruskin Bond', 10, 20),
(15, 'A Suitable Boy', 'Vikram Seth', 1, 7),
(16, 'Mahabharata', 'Vyasa', 16, 9),
(17, 'Panchatantra', 'Vishnu Sharma', 10, 8),
(18, 'A Journey to the Center of the Earth', 'Jules Verne', 14, 6),
(19, 'The Secret', 'Rhonda Byrne', 17, 18),
(20, 'Gitanjali', 'Rabindranath Tagore', 10, 10);

Members (MemberID, Name, Email, MemberType, Phone)

(1, 'Aarav Gupta', 'aarav.gupta@example.com', 'Student',
'9876543210'),
(2, 'Riya Sharma', 'riya.sharma@example.com', 'Student',
'9876543211'),
(3, 'Vikram Singh', 'vikram.singh@example.com', 'Staff',
'9876543212'),
(4, 'Ananya Iyer', 'ananya.iyer@example.com', 'Student',
'9876543213'),
(5, 'Rahul Mehta', 'rahul.mehta@example.com', 'Staff', '9876543214'),
(6, 'Diya Patel', 'diya.patel@example.com', 'Student', '9876543215'),
(7, 'Arjun Nair', 'arjun.nair@example.com', 'Staff', '9876543216'),
(8, 'Neha Desai', 'neha.desai@example.com', 'Student',
'9876543217'),
(9, 'Rohan Joshi', 'rohan.joshi@example.com', 'Staff', '9876543218'),

(10, 'Meera Reddy', 'meera.reddy@example.com', 'Student', '9876543219'),
(11, 'Ishaan Malhotra', 'ishaan.malhotra@example.com', 'Student', '9876543220'),
(12, 'Tara Kapoor', 'tara.kapoor@example.com', 'Student', '9876543221'),
(13, 'Aman Choudhary', 'aman.choudhary@example.com', 'Staff', '9876543222'),
(14, 'Anushka Jain', 'anushka.jain@example.com', 'Student', '9876543223'),
(15, 'Harsh Bansal', 'harsh.bansal@example.com', 'Staff', '9876543224'),
(16, 'Priya Saxena', 'priya.saxena@example.com', 'Student', '9876543225'),
(17, 'Karan Verma', 'karan.verma@example.com', 'Staff', '9876543226'),
(18, 'Sneha Bhatt', 'sneha.bhatt@example.com', 'Student', '9876543227'),
(19, 'Ravi Kumar', 'ravi.kumar@example.com', 'Staff', '9876543228'),
(20, 'Sanjana Deshmukh', 'sanjana.deshmukh@example.com', 'Student', '9876543229');

Staff (StaffID, Name, Email, Position, Phone)

(1, 'Rajesh Kumar', 'rajesh.kumar@example.com', 'Librarian', '9876543230'),
(2, 'Sneha Reddy', 'sneha.reddy@example.com', 'Assistant Librarian', '9876543231'),
(3, 'Amit Sharma', 'amit.sharma@example.com', 'Clerk', '9876543232'),
(4, 'Pooja Iyer', 'pooja.iyer@example.com', 'Manager', '9876543233'),

(5, 'Rakesh Mehta', 'rakesh.mehta@example.com', 'Data Entry', '9876543234'),
(6, 'Ishita Desai', 'ishita.desai@example.com', 'Assistant', '9876543235'),
(7, 'Nikhil Joshi', 'nikhil.joshi@example.com', 'Supervisor', '9876543236'),
(8, 'Rohit Kapoor', 'rohit.kapoor@example.com', 'Library Assistant', '9876543237'),
(9, 'Sakshi Verma', 'sakshi.verma@example.com', 'Book Organizer', '9876543238'),
(10, 'Anil Gupta', 'anil.gupta@example.com', 'Library Technician', '9876543239'),
(11, 'Komal Bhatt', 'komal.bhatt@example.com', 'Assistant', '9876543240'),
(12, 'Manoj Choudhary', 'manoj.choudhary@example.com', 'System Administrator', '9876543241'),
(13, 'Divya Saxena', 'divya.saxena@example.com', 'Event Manager', '9876543242'),
(14, 'Aakash Jain', 'aakash.jain@example.com', 'Research Assistant', '9876543243'),
(15, 'Kavya Malhotra', 'kavya.malhotra@example.com', 'Library Cataloger', '9876543244'),
(16, 'Uday Bansal', 'uday.bansal@example.com', 'Maintenance', '9876543245'),
(17, 'Naina Bhatt', 'naina.bhatt@example.com', 'Clerk', '9876543246'),
(18, 'Mohit Kumar', 'mohit.kumar@example.com', 'Coordinator', '9876543247'),
(19, 'Simran Kapoor', 'simran.kapoor@example.com', 'Supervisor', '9876543248'),
(20, 'Aditya Desai', 'aditya.desai@example.com', 'Manager', '9876543249');

BorrowingRecords (RecordID, BookID, MemberID, BorrowDate, ReturnDate)

(1, 1, 1, '2024-12-01', '2024-12-15'),
(2, 2, 2, '2024-12-02', '2024-12-16'),
(3, 3, 3, '2024-12-03', '2024-12-17'),
(4, 4, 4, '2024-12-04', '2024-12-18'),
(5, 5, 5, '2024-12-05', '2024-12-19'),
(6, 6, 6, '2024-12-06', '2024-12-20'),
(7, 7, 7, '2024-12-07', '2024-12-21'),
(8, 8, 8, '2024-12-08', '2024-12-22'),
(9, 9, 9, '2024-12-09', '2024-12-23'),
(10, 10, 10, '2024-12-10', '2024-12-24'),
(11, 11, 11, '2024-12-11', '2024-12-25'),
(12, 12, 12, '2024-12-12', '2024-12-26'),
(13, 13, 13, '2024-12-13', '2024-12-27'),