**V I S H W A S C P**

**Project-based practice assignment on MySQL:**

**Project**: Online Library Management System

**Description**: In this project, you will create a database for an online library management system that will keep track of books, authors, and users. The system will allow users to borrow and return books, and the librarian to manage the inventory of the library.

**Requirements**:

Create a database named 'library'.

Create tables for books, authors, and users.

**The 'books' table should have the following columns:**

book\_id (integer, primary key)

title (varchar)

author\_id (integer, foreign key)

publisher (varchar)

publish\_date (date)

quantity (integer)

**The 'authors' table should have the following columns:**

author\_id (integer, primary key)

first\_name (varchar)

last\_name (varchar)

email (varchar)

**The 'users' table should have the following columns:**

user\_id (integer, primary key)

first\_name (varchar)

last\_name (varchar)

email (varchar)

password (varchar)

**Create a table for borrowed books.**

The 'borrowed\_books' table should have the following columns:

id (integer, primary key)

book\_id (integer, foreign key)

user\_id (integer, foreign key)

borrowed\_date (date)

due\_date (date)

returned\_date (date)

**Insert some sample data into the tables.**

**Write SQL queries to perform the following operations:**

1. Display all the books in the library.

SELECT \* FROM books;

1. Display all the authors in the library.

SELECT \* FROM authors;

1. Display all the users in the library.

SELECT \* FROM users;

1. Display all the borrowed books.

SELECT \* FROM borrowed\_books;

1. Display all the books borrowed by a particular user.

SELECT books.\*

FROM books

INNER JOIN borrowed\_books

ON books.book\_id = borrowed\_books.book\_id

WHERE borrowed\_books.user\_id = 3;

1. Display all the books written by a particular author.

SELECT \* FROM books WHERE author\_id = 1;

1. Display the number of available copies for a particular book.

SELECT (b.quantity - COUNT(bb.book\_id)) AS available\_copies

FROM books b

LEFT JOIN borrowed\_books bb ON b.book\_id = bb.book\_id

WHERE b.book\_id = 123;

1. Add a new book to the library.

INSERT INTO books (book\_id, title, author\_id, publisher, publish\_date, quantity)

VALUES (12345, 'The Catcher in the Rye', 1, 'Little, Brown and Company', '1951-07-16', 5);

1. Update the quantity of a book in the library.

UPDATE books

SET quantity = 10

WHERE book\_id = 12345;

1. Delete a book from the library.

DELETE FROM books

WHERE book\_id = 12345;

1. Add a new user to the library.

INSERT INTO users (user\_id, first\_name, last\_name, email, password)

VALUES (123, 'John', 'Doe', 'johndoe@example.com', 'mypassword');

1. Update the password of a user.

UPDATE users

SET password = 'newpassword'

WHERE user\_id = 123;

1. Delete a user from the library.

DELETE FROM users

WHERE user\_id = 123;

1. Borrow a book.

INSERT INTO borrowed\_books (id, book\_id, user\_id, borrowed\_date, due\_date)

VALUES (4, 2, 2, '2023-03-28', '2023-04-04');

1. Return a book.

UPDATE borrowed\_books

SET returned\_date = '2023-04-01'

WHERE id = 1;

Test your queries to ensure they are working properly.

**Create a web interface for the library management system using ReactJS/Node JS and MySQL.**