**Task 12: Develop micro project based on business scenarios**

Title: Online Book Store Manager System

Aim:

To design and implement an Online Book Store Manager System that allows customers to browse, search, and purchase books online, while enabling the administrator to manage book inventory, handle orders, and maintain records efficiently through a relational database using SQL.

Required SQL Queries:

1. Create Database

CREATE DATABASE OnlineBookStore;

USE OnlineBookStore;

2. Create Tables (DDL Commands)

CREATE TABLE Customer (

Customer\_ID INT PRIMARY KEY,

Name VARCHAR(50),

Email VARCHAR(50),

Password VARCHAR(20)

);

CREATE TABLE Book (

Book\_ID INT PRIMARY KEY,

Title VARCHAR(100),

Author VARCHAR(50),

Category VARCHAR(40),

Price DECIMAL(8,2),

Stock INT CHECK (Stock >= 0)

);

CREATE TABLE Orders (

Order\_ID INT PRIMARY KEY,

Order\_Date DATE,

Customer\_ID INT,

Total\_Amount DECIMAL(10,2),

FOREIGN KEY (Customer\_ID) REFERENCES Customer(Customer\_ID)

);

CREATE TABLE Order\_Details (

Order\_ID INT,

Book\_ID INT,

Quantity INT CHECK (Quantity > 0),

FOREIGN KEY (Order\_ID) REFERENCES Orders(Order\_ID),

FOREIGN KEY (Book\_ID) REFERENCES Book(Book\_ID)

);

3. Insert Sample Records (DML Commands)

INSERT INTO Customer VALUES (101, 'Arjun', 'arjun@gmail.com', 'arj123');

INSERT INTO Customer VALUES (102, 'Meena', 'meena@gmail.com', 'mnn456');

INSERT INTO Book VALUES (1, 'Data Structures', 'Reema Thareja', 'Education', 450.00, 20);

INSERT INTO Book VALUES (2, 'Rich Dad Poor Dad', 'Robert Kiyosaki', 'Finance', 350.00, 15);

INSERT INTO Book VALUES (3, 'Wings of Fire', 'APJ Abdul Kalam', 'Biography', 300.00, 10);

4. Display All Books

SELECT \* FROM Book;

5. Search Books by Category

SELECT Title, Author, Price FROM Book WHERE Category = 'Education';

6. Update Book Stock

UPDATE Book SET Stock = Stock - 2 WHERE Book\_ID = 1;

7. Delete a Book

DELETE FROM Book WHERE Book\_ID = 3;

8. Create an Order

INSERT INTO Orders VALUES (501, '2025-02-01', 101, 900.00);

INSERT INTO Order\_Details VALUES (501, 1, 1);

INSERT INTO Order\_Details VALUES (501, 2, 1);

9. View Customer Purchase History (JOIN Query)

SELECT Customer.Name, Book.Title, Order\_Details.Quantity, Orders.Order\_Date

FROM Customer

JOIN Orders ON Customer.Customer\_ID = Orders.Customer\_ID

JOIN Order\_Details ON Orders.Order\_ID = Order\_Details.Order\_ID

JOIN Book ON Book.Book\_ID = Order\_Details.Book\_ID;

10. Show Books with Low Stock

SELECT Title, Stock

FROM Book

WHERE Stock < 5;

Result:

The Online Book Store Management System was successfully implemented using SQL queries for creating tables, inserting records, and retrieving information from the database. The system allows efficient storage, management, and retrieval of book details, customer details, orders, and payment data. All required operations such as searching books, placing orders, viewing order history, and managing inventory are performed accurately through SQL commands.