Case Study: Library Management System

Section 1: Python Standalone Console Application

Design and implement a standalone console application for a Library Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage the inventory of books.

Requirements:

1. **Book Management**:
    - Implement the functionality to add, update, and delete books.
    - Each book should have attributes such as book_id, title, author, genre, price, and quantity_in_stock.
2. **Library Operations**:
    - Implement the functionality to check the availability of a book.
    - Implement the functionality to update the quantity of books when borrowed or returned.
3. **Reporting**:
    - Implement the functionality to generate a report of books that are low in stock (quantity_in_stock < 5).

Business Functionalities:

1. **Add/Update/Delete Books**:
    - Create a class Book with attributes book_id, title, author, genre, price, and quantity_in_stock.
    - Implement methods to add a new book, update existing book details, and delete a book from the library.
2. **Check Availability and Update Quantity**:
    - Implement a method to check the availability of a book by book_id.
    - Implement a method to update the quantity of books when borrowed or returned.
3. **Low Stock Report**:
    - Implement a method to generate a list of books that are low in stock (quantity_in_stock < 5).

Section 2: MySQL Database Management

Design a MySQL database schema to support the Library Management System and provide solutions for the problem statements.

Table Structures:

1. **Books Table**:
    - book_id: INT, Primary Key
    - title: VARCHAR(100)
    - author: VARCHAR(50)
    - genre: VARCHAR(50)
    - price: DECIMAL(10, 2)
    - quantity_in_stock: INT
2. **Borrowers Table**:
    - borrower_id: INT, Primary Key
    - name: VARCHAR(100)
    - contact: VARCHAR(50)
3. **Borrowing Table**:
    - borrowing_id: INT, Primary Key
    - book_id: INT, Foreign Key References Books(book_id)
    - borrower_id: INT, Foreign Key References Borrowers(borrower_id)
    - quantity_borrowed: INT
    - borrowing_date: DATE

Problem Statements:

1. Write a query to find the total quantity of each book borrowed.
2. Write a query to find the book title and total quantity borrowed for each book.
3. Write a query to find the titles of books that have never been borrowed.
4. Write a query to find the books that have been borrowed more than 10 times.
5. Write a query to find the book titles and their current stock levels for books that have been borrowed more than 20 times.