**FINAL PROJECT: KULIAH BASISDATA LANJUT**

**1 Group max. 3 orang, tugas membuat aplikasi (boleh web, mobile, atau apapun). Terkait studi kasus dengan contoh berikut (bagian warna orange boleh ada boleh tidak).**

**Case Study: Library Management System with CRUD Operations**

Scenario: You are tasked with developing a library management software system for a public library using Microsoft SQL Server. The system should allow librarians to manage books, patrons, loans, and fines efficiently. The database should include at least six tables with their attributes, and you are required to create functions, stored procedures, and triggers to facilitate CRUD operations.

**Database Tables and Attributes**:

1. **Book**
   - BookID (Primary Key)
   - ISBN
   - Title
   - Author
   - Genre
   - PublicationYear
   - QuantityAvailable
   - QuantityTotal
2. **Patron**
   - PatronID (Primary Key)
   - FirstName
   - LastName
   - Email
   - PhoneNumber
3. **Loan**
   - LoanID (Primary Key)
   - BookID (Foreign Key)
   - PatronID (Foreign Key)
   - LoanDate
   - DueDate
   - ReturnDate (Nullable)
4. **Fine**
   - FineID (Primary Key)
   - PatronID (Foreign Key)
   - Amount
   - PaymentStatus
   - DueDate
5. **Reservation**

- ReservationID (Primary Key)
- BookID (Foreign Key)
- PatronID (Foreign Key)
- ReservationDate

6. **LibraryStaff**
   - StaffID (Primary Key)
   - FirstName
   - LastName
   - Email
   - PhoneNumber

**Functional Requirements**:

1. Librarians should be able to add new books to the library, specifying the ISBN, title, author, genre, publication year, and the total quantity available.
2. Librarians should be able to view a list of all books in the library, including details such as the current quantity available.
3. Librarians should be able to add new patrons to the library system, providing their first name, last name, email, and phone number.
4. Librarians should be able to loan books to patrons, recording the loan date, due date, and the book's status.
5. Librarians should be able to mark books as returned, updating the return date and quantity available.
6. Librarians should be able to assess fines for overdue books and mark fines as paid when patrons make payments.
7. Patrons should be able to reserve books that are currently unavailable, with the system recording the reservation date.

**DB Functions**:

1. Create a function to calculate the total fines owed by a specific patron.
2. Create a function to check whether a book is available for loan or reserved.

**DB Stored Procedures**:

1. Implement a stored procedure to add a new book to the library.
2. Develop a stored procedure to loan books to patrons.
3. Create a stored procedure to mark books as returned and update the available quantity.
4. Implement a stored procedure to assess fines for overdue books and record payments.

**DB Triggers**:

Create a trigger that automatically updates the quantity available in the Book table whenever a book is loaned or returned.

**Requirements**:

1. The system should have user authentication and authorization mechanisms to distinguish between librarians and patrons.
2. Librarians should have the ability to search for books by title, author, or ISBN.
3. Patrons should receive notifications via email for due dates and fines.
4. The software should generate reports for librarians to track overdue books, fines collected, and inventory.
5. Ensure data consistency and integrity, preventing the loan of more copies of a book than are available.
6. Implement a reservation system that allows patrons to be notified when a reserved book becomes available.

This case study challenges you to design a comprehensive library management software system with a robust database schema, including tables, attributes, functions, stored procedures, and triggers. It also outlines key functional requirements for the software.