SQL Assignment

Database Schema:

You will be working with a database that models a library. Here are the relevant tables:

books - Contains information about books in the library.

Columns: book\_id (Primary Key), title, author, publication\_year, isbn.

users - Contains information about library users.

Columns: user\_id (Primary Key), first\_name, last\_name, email, registration\_date.

borrowed\_books - Records when a user borrows a book.

Columns: borrow\_id (Primary Key), user\_id (Foreign Key), book\_id (Foreign Key), borrow\_date, return\_date.

**Test Questions:**

Write a SQL query to retrieve the top 10 most borrowed books, along with the number of times each book has been borrowed.  
  
**SELECT books.title, COUNT(borrowed\_books.book\_id) AS borrow\_count**

**FROM books JOIN borrowed\_books ON books.book\_id = borrowed\_books.book\_id**

**GROUP BY books.title ORDER BY borrow\_count DESC LIMIT 10;**

Create a stored procedure that calculates the average number of days a book is borrowed before being returned. The procedure should take a book\_id as input and return the average number of days.

**DELIMITER //**

**CREATE PROCEDURE CalculateAverageBorrowDuration(IN p\_book\_id INT, OUT p\_average\_duration FLOAT)**

**BEGIN SELECT AVG(DATEDIFF(return\_date, borrow\_date)) INTO p\_average\_duration**

**FROM borrowed\_books WHERE book\_id = p\_book\_id;**

**END //**

**DELIMITER ;**

**CALL CalculateAverageBorrowDuration(1, @average\_duration);**

**SELECT @average\_duration AS average\_duration;**

Write a query to find the user who has borrowed the most books from the library.

**SELECT users.user\_id, users.first\_name, users.last\_name, COUNT(borrowed\_books.book\_id) AS books\_borrowed\_count**

**FROM users JOIN borrowed\_books ON users.user\_id = borrowed\_books.user\_id**

**GROUP BY users.user\_id, users.first\_name, users.last\_name**

**ORDER BY books\_borrowed\_count DESC**

**LIMIT 1;**

Create an index on the publication\_year column of the books table to improve query performance.

**CREATE INDEX index\_publication\_year ON books(publication\_year);**

Write a SQL query to find all books published in the year 2020 that have not been borrowed by any user.

**SELECT books.book\_id, books.title, books.author, books.publication\_year**

**FROM books WHERE books.publication\_year = 2020 AND books.book\_id NOT IN (SELECT book\_id FROM borrowed\_books);**

Design a SQL query that lists users who have borrowed books published by a specific author (e.g., "J.K. Rowling").

**SELECT DISTINCT users.user\_id, users.first\_name, users.last\_name**

**FROM users JOIN borrowed\_books ON users.user\_id = borrowed\_books.user\_id**

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

**WHERE books.author = 'John Green';**

Create a trigger that automatically updates the return\_date in the borrowed\_books table to the current date when a book is returned.

**DELIMITER //**

**CREATE TRIGGER update\_return\_date**

**BEFORE UPDATE ON borrowed\_books**

**FOR EACH ROW**

**BEGIN**

**IF NEW.return\_date IS NOT NULL THEN**

**SET NEW.return\_date = CURRENT\_DATE;**

**END IF;**

**END //**

**DELIMITER ;**

Please remember to create and use any necessary tables, procedures, or triggers in the database to answer these questions