

Database triggers are specialized stored programs. These triggers fire when a triggering event occurs. Consider the following scenario of an online bookstore database:

Scenario: Online Bookstore Database

Consider the online bookstore database that you created for Lab 8. Beside selling books, the bookstore has decided to lend books as well.

To implement this scenario, update the 'books' table and add another table called 'borrowed_books' like the following:

Tables:

a. Books:

Columns:

- book_id (Primary Key)
- title (Title of the book)
- author_id (Foreign Key referencing the Authors table)
- price (Price of the book)
- publication_date (Date of publication)
- availability (Number of books available in the store)

b. borrowed_books:

Columns:

- transaction_id (identifier for borrowing transaction-Primary Key)
- book_id (foreign_key referencing books table)
- borrower_name
- borrowed_date
- returning_date (the date the book is promised to be returned)

Data:

Insert some books, and borrowed_books data into their respective tables. Create relationships between the tables, ensuring that foreign keys are properly set to maintain data integrity.

1. Create a trigger 'update_book_availability' to automatically update the availability status of a book in the books table when a book is borrowed or returned.

Logic:

If a new record is inserted into borrowed_books, the trigger decreases the availability of the corresponding book by 1.

If an existing record is updated, or the returning_date is behind of the PRESENT_DATE, the trigger increases the availability of the corresponding book by 1.

Trigger Type:

The trigger is an AFTER INSERT OR UPDATE trigger on the

borrowed_books table.

2. In the previous scenario, create another trigger that logs changes to the 'borrowed_books' table in a separate 'book_transaction_log' table.

In this connection, create a function called 'log_book_transaction' that encapsulates this trigger to insert records into the 'book_transaction_log' table based on the operations (INSERT or UPDATE) in the borrowed_books table.

3. Assume that both triggers (in Questions 1 & 2) are set to fire AFTER INSERT OR UPDATE on the borrowed_books table. Specify the ideal firing order for these triggers to ensure proper functionality and data consistency. Provide specific examples of the chosen firing order of the triggers.

If the firing order is changed, discuss how it might impact the behavior of the system.