Using Triggers in PostgreSQL to Maintain User Transactions and Logs

Objective

Create a PostgreSQL database that uses triggers to maintain a log of user transactions. The goal is to track all changes (INSERT, UPDATE, DELETE) made to a `Transactions` table and store the details in a separate `TransactionLog` table for auditing purposes.

Assignment Instructions

1. Create the Schema

1. Create a `Users` table to store user details.
2. Create a `Transactions` table to store user transactions (e.g., purchases, deposits, withdrawals).
3. Create a `TransactionLog` table to store logs of all operations performed on the `Transactions` table.

2. Set Up the Tables

Use the following SQL commands as a reference:

```sql
-- Create Users Table
CREATE TABLE Users (
    UserId SERIAL PRIMARY KEY,
    FullName VARCHAR(100),
    Email VARCHAR(100)
);

-- Create Transactions Table
CREATE TABLE Transactions (
    TransactionId SERIAL PRIMARY KEY,
    UserId INT REFERENCES Users(UserId),
    TransactionType VARCHAR(50), -- e.g., 'Deposit', 'Withdrawal'
    Amount NUMERIC(10, 2),
    TransactionDate TIMESTAMP DEFAULT NOW()
);
```

```
-- Create TransactionLog Table
CREATE TABLE TransactionLog (
    LogId SERIAL PRIMARY KEY,
    TransactionId INT,
    OperationType VARCHAR(10), -- 'INSERT', 'UPDATE', 'DELETE'
    OldValues JSONB,
    NewValues JSONB,
    OperationDate TIMESTAMP DEFAULT NOW()
);
```