LAB: 4

Aim: To learn Concurrency Control Concerns in Transactions

Through this lab, you will solve:

1. Dirty Read Concurrency Problem in SQL Server
2. Last Update Concurrency Problem
3. Non-Repeatable Read Concurrency Problem
4. Phantom Read Concurrency Problem in SQL Server

Assignment: Bank Account Transactions

Consider a scenario where multiple users are performing transactions on a bank account concurrently. Each transaction involves withdrawing a certain amount of money from the account. However, the balance of the account should never go below zero.

Assume the following table structure for the bank account:

```
CREATE TABLE BankAccount ( AccountNumber INT PRIMARY KEY, Balance
DECIMAL(10, 2) );
```

1) Create a sample account with an initial balance of $1000.

2) Write a SQL transaction that simulates concurrent withdrawals from the account. The transaction should ensure that the balance doesn't go below zero.

3) Execute multiple instances of the transaction concurrently and observe the results. Discuss any concurrency problems you encounter and possible solutions.

Hints: Use appropriate SQL statements and clauses (e.g., BEGIN TRANSACTION, COMMIT, ROLLBACK, UPDATE, etc.) to handle transactions. Consider using locking mechanisms or isolation levels to prevent concurrent access issues.