

# 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: Inventory Stock Management**

Consider a scenario where multiple users are updating the stock levels of different products concurrently in an inventory management system. Each product has a unique identifier, and users can increase or decrease the stock level of a product. However, the stock level should never go below zero.

Assume the following table structure for the bank account:

```
CREATE TABLE SeatReservation (SeatID INT PRIMARY KEY, ReservedBy  
VARCHAR(50));
```

- 1) Create a sample inventory with products and initial stock levels.
- 2) Write a SQL transaction that simulates concurrent stock updates for different products. The transaction should ensure that the stock levels are correctly updated while preventing negative stock values.
- 3) Execute multiple instances of the transaction concurrently and observe any concurrency issues that arise. Discuss potential solutions to address these issues.

### **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.