

## Transaction Pair-II

Suppose the Admin adds a particular product to the stock, which means the quantity increases. At the same time, a customer buys the same product whose quantity was increased, meaning that its quantity should be decreased after successfully purchasing that product.

Variable Notations meaning:-

X represents the quantity (in stock) of Product 1.

x1 represents the quantity of Product 1, which the admin increased.

x2 represents the quantity of Product 1, which the customer purchased.

Bal represents the balance of the customer.

BalNew represents the new balance of the customer.

Ord represent the customer's order.

| Transaction-1 (T-1)                           | Transaction-2 (T-2)   |
|---|---|
| Read(X)<br>$X = X + x1$<br>Write(X)<br>Commit | Read(X)<br>$X = X - x2$<br>Write(X)<br>Read(Bal)<br>Write(BalNew)<br>Write(Order Ord)<br>Commit |

## A Serial Schedule of the two transactions

| Transaction-1 (T-1)                           | Transaction-2 (T-2)   |
|---|---|
| Read(X)<br>$X = X + x1$<br>Write(X)<br>Commit | Read(X)<br>$X = X - x2$<br>Write(X)<br>Read(Bal)<br>Write(BalNew)<br>Write(Order Ord)<br>Commit |

## Conflict Serializable Schedule

| Transaction-1 (T-1)                           | Transaction-2 (T-2)   |
|---|---|
| Read(X)<br>$X = X + x1$<br>Write(X)<br>Commit | Read(X)<br>$X = X - x2$<br>Write(X)<br><br>Read(Bal)<br>Write(BalNew)<br>Write(Order Ord)<br>Commit |

This Schedule is conflict serializable as we can easily move down the three statements of T-1, which results in a serial schedule (T-2 =>T-1).

## Non-Conflict Serializable Schedule

| Transaction-1 (T-1)                           | Transaction-2 (T-2)   |
|---|---|
| Read(X)<br>$X = X + x1$<br>Write(X)<br>Commit | Read(X)<br><br>$X = X - x2$<br>Write(X)<br>Read(Bal)<br>Write(BalNew)<br>Write(Order Ord)<br>Commit |

This schedule is non-conflict serializable as there exists a loop in the precedence graph between T-2 and T-1 on data item Q.

This is because T1 writes to Q before and after a read and write, respectively, which are executed by T2.

## Non-Conflict Serializable Schedule With Locks

| Transaction-1 (T-1)   | Transaction-2 (T-2)   |
|---|---|
| Lock-X(X)<br>Read(X)<br>$X = X + x1$<br>Write(X)<br>Unlock(X)<br>Commit | Lock-S(X)<br>Read(X)<br>Unlock(X)<br><br>Lock-X(X)<br>$X = X - x2$<br>Write(X)<br>Unlock(X)<br>Lock-X(Bal)<br>Read(Bal)<br>Write(BalNew)<br>Unlock(Bal)<br>Lock-X(Order Ord)<br>Write(Order Ord)<br>Unlock(Order Ord)<br>Commit |

In this schedule, we add locks to each data item to prevent conflicts and dirty reads. Adding locks makes sure that no two transactions simultaneously manipulate the same data item.