

## DB assignment # 4

Q1.

- a. The ACID properties violated are Atomicity and Consistency. Atomicity is violated because the transaction was partially completed whereas, consistency is violated because DB is left in an inconsistent state.
- b. The violation happened because the transaction was not treated as an all-or-nothing unit. In a DBMS transactions should use commit or rollbacks mechanisms. Here, the server crash occurred, mid-transaction, after some writes but before the final commit that would include the customer confirmation. This led to breaking atomicity and inconsistency.
- c. A customer books a ride, which initiates a transaction. The system receives the payment from the wallet, records the driver's acceptance, and prepares the confirmation. All operations are wrapped in a single atomic transaction. If all steps succeed, the system issues COMMIT, showing the confirmation to the customer. If the server crashes before commit, upon recovery the transaction is rolled back using logs: wallet balance is restored, driver acceptance is undone, and no confirmation is sent. This ensures all ACID properties.
- d. Frequent violations could lead to loss of customer trust. financial discrepancies might cause revenue loss from refunds or disputes, legal issues from payment errors, reputational damage via negative reviews, and

23K-0086

Pitted horizon line

Date:

M T W T F S S

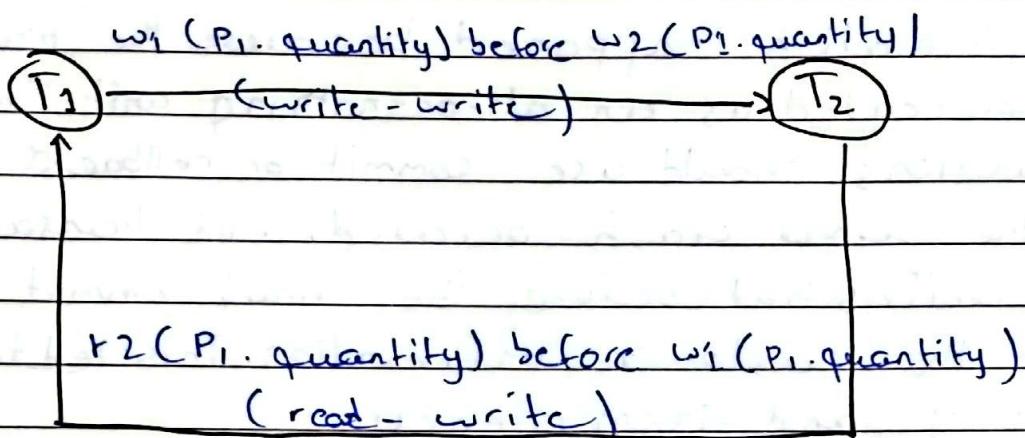
operational inefficiencies.

Q2

a. non-Serial Schedule:

$r_1(P_1 \cdot \text{quantity})$ ,  $r_2(P_1 \cdot \text{quantity})$ ,  $w_1(P_1 \cdot \text{quantity} - \text{sold units})$   
 $w_2(P_1 \cdot \text{quantity} + \text{returned units})$

b,



c, The Schedule is not Conflict-Serializable because the procedure graph contains a cycle, indicating no equivalent serial order without conflicts,

d, Corrected Schedule:

$r_1(P_1 \cdot \text{quantity})$ ,  $w_1(P_1 \cdot \text{quantity} - \text{sold units})$ ,  $r_2(P_1 \cdot \text{quantity})$ ,  
 $w_2(P_1 \cdot \text{quantity} + \text{returned units})$

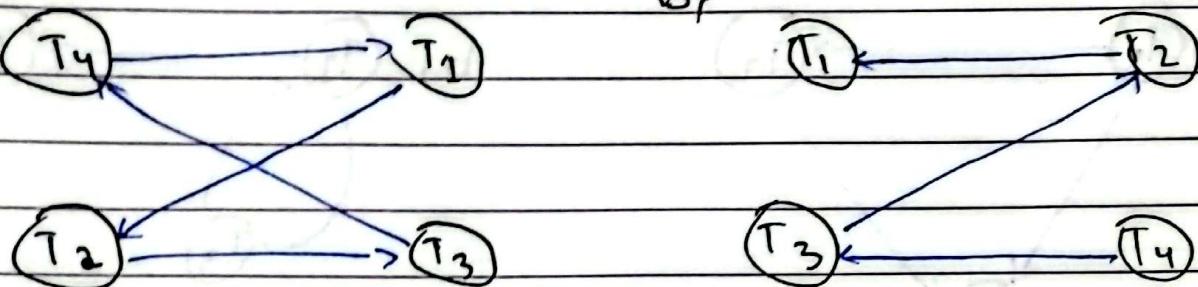
e, Inventory inaccuracies could lead to overstocking under causing lost sales or even excess holding costs. This might result in financial losses from return/refunds, damaged customer relationships, supply chain disruptions and reputational harm

23K.0080

Date:  
MTWTFSS

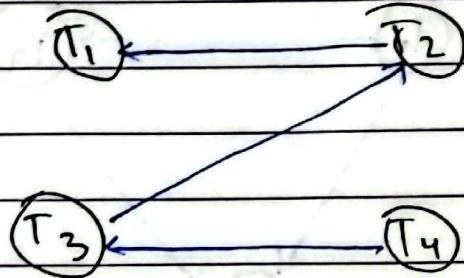
if products are frequently out of stock.

Q3



a, b,  
b, not conflict-serializable(cycle)

c not serializable



conflict-Serializable (acyclic)  
The serial schedule is  
 $T_4 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1$

Q4.a. The concurrency problems are lost update and dirty Read (as more precisely, a write-write conflict) leading to lost update). They arise because both transactions read the same initial value (2000). without isolation, compute independently and overwrite each other.

b. The final amount is 2800 l. after both transactions finish,

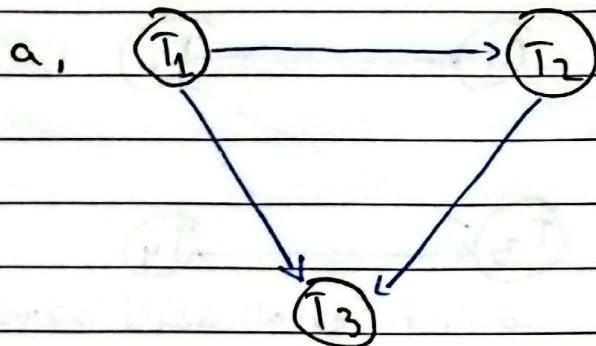
c,  $T_1 \rightarrow T_2$ :

$$2000 - 300 = 1700 + 600 = 2500$$

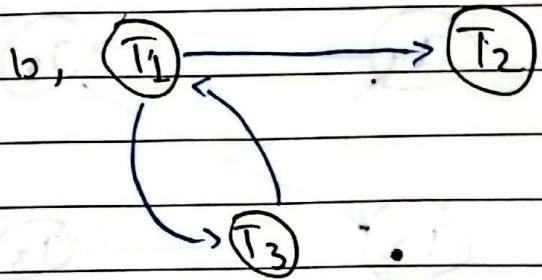
$T_2 \rightarrow T_3$ :

$$2000 + 800 = 2800 - 300 = 2500$$

Q5.

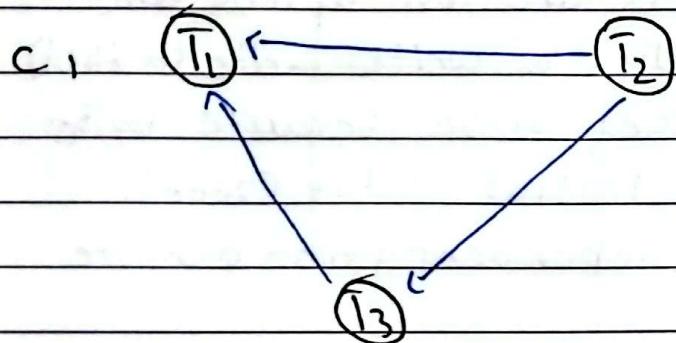


Conflict serializable (acyclic)

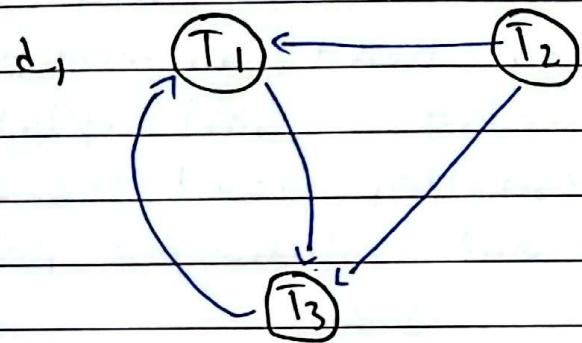
Serial schedule:  $T_1 \rightarrow T_2 \rightarrow T_3$ 

Not conflict serializable

(cycle)



Conflict serializable (acyclic)

Serial schedule:  $T_2 \rightarrow T_3 \rightarrow T_1$ 

Not conflict

serializable (cyclic)