Syed Huzaifa Ali
23K-0004

Database Systems

Assignment # 04

Q1.

a). The ACID properties violated are Atomicity and Consistency. Atomicity is violated because the transaction was partially completed whereas, Consistency is violated because the database is left in an inconsistent state.

b). The violation happened because the transaction was not treated as an all-or-nothing unit. In a database system, transactions should use commit or rollback mechanisms. Here, the server crash occurred mid-transaction, after some writes (driver acceptance and wallet deduction) 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 reserves 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 operational inefficiencies.

CamScanner

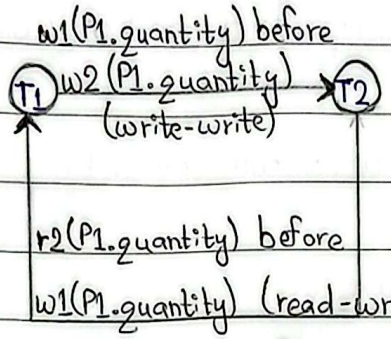
23K-0004

Q2.

a). non-serial schedule:

r1(P1.quantity), r2(P1.quantity), w1(P1.quantity - sold_units), w2(P1.quantity + returned_units)

b).



c). The schedule is not conflict-serializable because the precedence graph contains a cycle, indicating no equivalent serial order without conflicts.

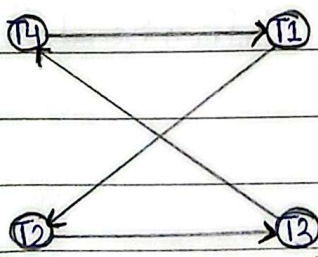
d). Corrected schedule:

r1(P1.quantity), w1(P1.quantity - sold_units), r2(P1.quantity), w2(P1.quantity + returned_units)

e). Inventory inaccuracies could lead to overstocking/understocking, causing lost sales or excess holding costs. This might result in financial losses from return/refunds, damaged customer relationships, supply chain disruptions, and reputational harm if products are frequently out of stock.

Q3.

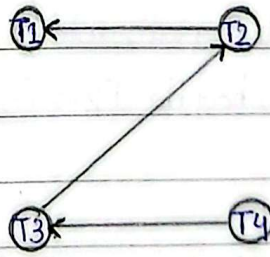
a).



b). Not conflict-serializable (cycle)

c). Not serializable.

b).



Conflic-serializable (acyclic)

The serial schedule is

T4 → T3 → T2 → T1

23K-0004

Q4.

a). The concurrency problems are Lost Update and Dirty Read (or 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/- after both transactions finish.

c). T1 → T2:

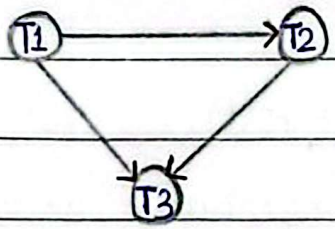
2000 - 300 => 1700 + 800 => 2500

T2 → T1:

2000 + 800 => 2800 - 300 => 2500

Q5.

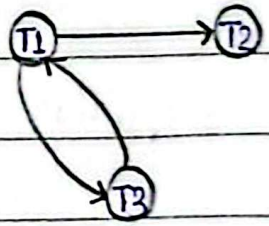
a)



Conflict Serializable (acyclic)

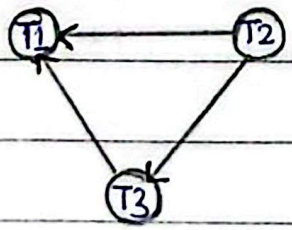
Serial schedule: T1 → T2 → T3

b)



Not Conflict Serializable (cycle)

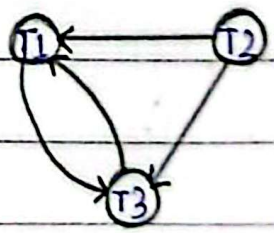
c)



Conflict Serializable (acyclic)

Serial schedule: T2 → T3 → T1

d)



Not Conflict Serializable (cyclic)

CamScanner