DBMS PROJECT

Group No. : 108

DealsVerge

Aryan Dhull | 2021520

Deepanshu IIITD | 2021524

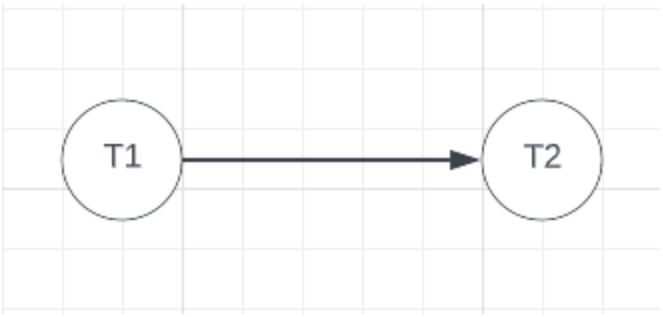
TRANSACTION SET 1

Transaction 1 : Increases the price of a product A by 150 and B by 450
Transaction 2 : Decreases the price of a product A by 360.

Conflict Serializable Schedule:

T1	T2
R(A) : Read the current price of product A	
W(A) : Write new price (A=A+150)	
	R(A) : Read the current price of product A
	W(A) : Write new price (A=A-360)
R(B) : Read the current price of product B	
W(B) : Write new price (B=B+450)	

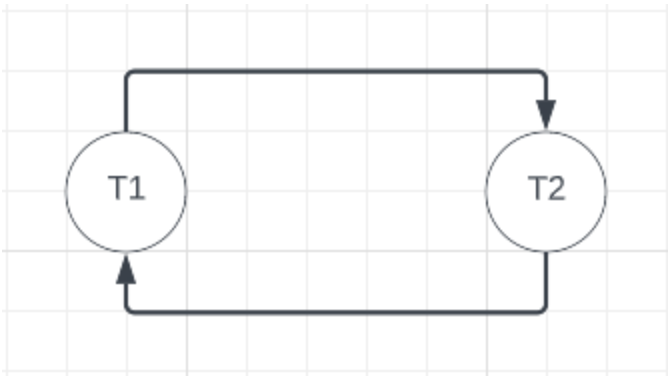
There is a type of conflict called WR conflict, where a transaction T2 reads data that was written by transaction T1. If we represent T1 and T2 as nodes and draw an arrow from T1 to T2 to indicate the WR conflict, the resulting graph is a directed acyclic graph (DAG). Since a DAG represents a conflict serializable transaction schedule, we can conclude that this type of conflict is also conflict serializable.



Non Conflict Serializable Schedule:

T1	T2
	R(A) : Read the current price of product A
R(A) : Read the current price of product A	
W(A) : Write new price (A=A+150)	
	W(A) : Write new price (A=A-360)
R(B) : Read the current price of product B	
W(B) : Write new price (B=B+450)	

There is a conflict involving both RW and WW operations on product A between two transactions, T1 and T2. If we construct a precedence graph for these transactions, we will observe a loop in the graph, indicating that the schedule is non-conflict serializable. This means that the transactions cannot be rearranged to create a serializable schedule. The loop is formed due to a RW conflict from T2 to T1 and a combination of RW and WW conflicts from T1 to T2.



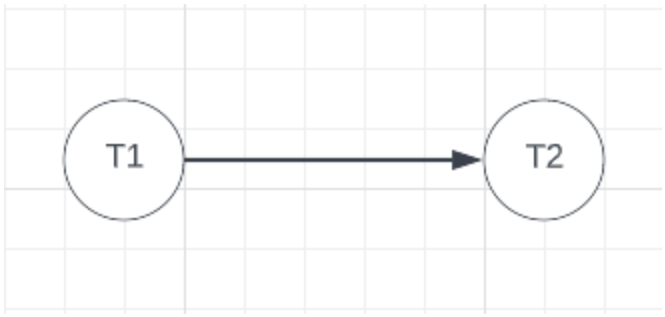
TRANSACTION SET 2

Transaction 1 : Update stock of a product and add it to cart
Transaction 2 : Update cart table (add another product or change quantity if existing product)

Conflict Serializable Schedule:

T1	T2
R(A) : Read the current stock of product	
W(A) : Write new stock/ Update stock	
R(B) : Read current state of cart	
W(B) : Add product to cart	
	R(B) : Read current state of cart
	W(B) : Update cart (add new/ update exiting)

There is a type of conflict called WR conflict, where a transaction T2 reads data that was written by transaction T1. If we represent T1 and T2 as nodes and draw an arrow from T1 to T2 to indicate the WR conflict, the resulting graph is a directed acyclic graph (DAG). Since a DAG represents a conflict serializable transaction schedule, we can conclude that this type of conflict is also conflict serializable.



Non Conflict Serializable Schedule:

T1	T2
R(A) : Read the current stock of product	
W(A) : Write new stock/ Update stock	
R(B) : Read current state of cart	
	R(B) : Read current state of cart
	W(B) : Update cart (add new/ update exiting)
W(B) : Add product to cart	

There is a conflict involving both RW and WW operations on cart between two transactions, T1 and T2. If we construct a precedence graph for these transactions, we will observe a loop in the graph, indicating that the schedule is non-conflict serializable. This means that the transactions cannot be rearranged to create a serializable schedule. The loop is formed due to a RW conflict from T2 to T1 and a combination of RW and WW conflicts from T1 to T2.