Transactions

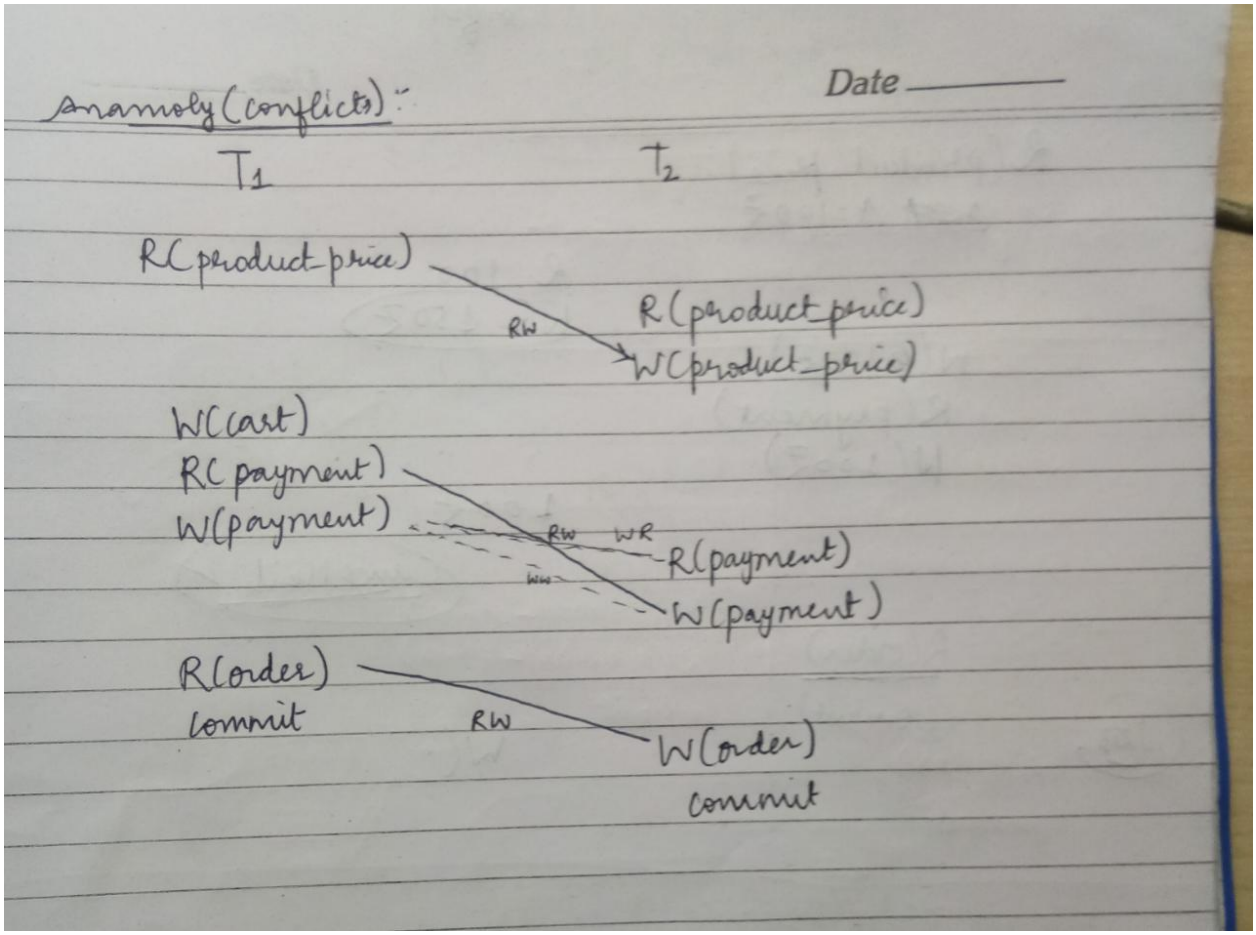
Submitted by->Shyama Goel(2021492),shivam(2021489)

Conflicting serializability schedule

T1	T2
R(product_price)	
	R(product_price) W(product_price)
W(cart) R(payment) W(payment)	
	R(payment) W(payment)
R(order) commit	
	W(order) commit

Let the product price for a product A read by the user is 200 Rs at the after that the admin reads the mrp of the same product as 200 Rs and then He changes(write the price)to be 250Rs (200+50) then customers reads the product price as 200 Rs and add that product into his cart and read the payment and proceed for payment process after paying that amount admin reads the amount as 200 rs he quickly rejects the payment after that he reads his order and commit then after order red by customer admin cancels his order(write his order).

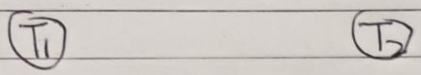
Anomaly(conflicts):



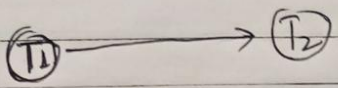
There are two anomaly present in the schedule
There is RW conflict present in product price from T1 to T2
RW conflict is present in the payment from T1 to T2
WR, WW conflict also present in the payment from T1 to T2
And after that RW conflict present in the order from T1 to T2.

Precedence graph:

for making precedence graph we have to choose the the transactions as node of the graph and conflict that taken as the edges of the directed graph. If the cycle is found in the graph then graph will be called as conflicting serializable schedule. Here the node is T_1 and T_2.



There is the RW in product price from $T_1 \rightarrow T_2$



Now every conflicts RW(payment) WR(payment) and RW (in order) goes from T_1 to T_2 Hence, the edge will be in one dir^n from $T_1 \rightarrow T_2$. Hence, no cycle forms that's why this is conflict Serializable schedule.

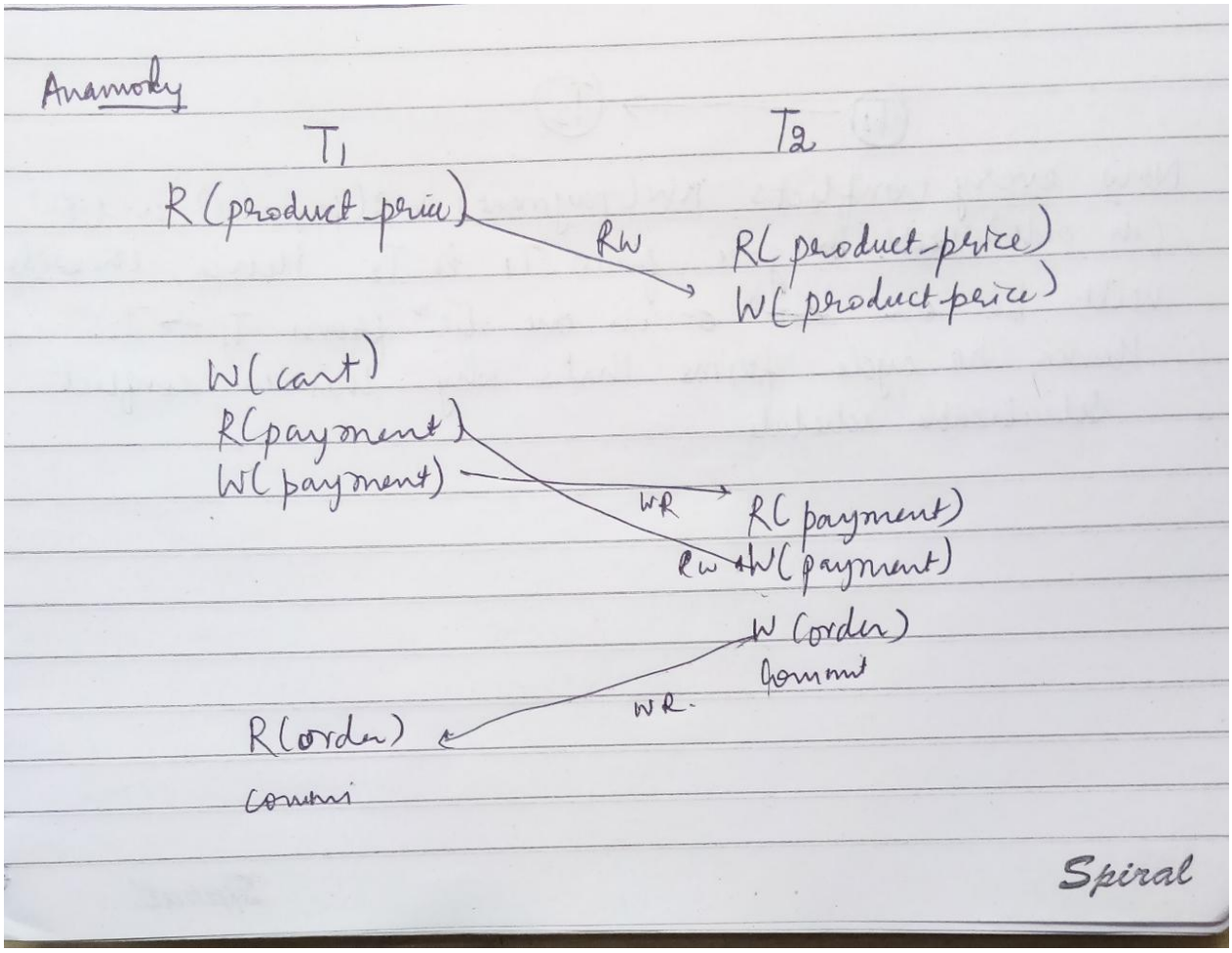
Non- conflicting serializable schedule

T1	T2
R(product_price)	
	R(product_price) W(product_price)
W(cart) R(payment) W(payment)	
	R(payment) W(payment)
	W(order) commit
R(order) commit	

Change from conflicting serializable to non-conflict serializable By swapping(R(order) and W(order)

Let the product price for a product A read by the user is 200 Rs at the after that the admin reads the mrp of the same product as 200 Rs and then He changes(write the price)to be 250Rs (200+50) then customers reads the product price as 200 Rs and add that product into his cart and read the payment and proceed for payment process after paying that amount admin reads the amount as 200 rs he quickly rejects the payment after that admin cancel his order and then the customer reads his order.

Anomaly (conflict):



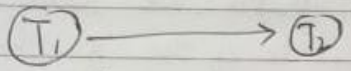
There are two anomaly present in the schedule
There is RW conflict present in product price from T1 to T2
RW conflict is present in the payment from T1 to T2
WR, WW conflict also present in the payment from T1 to T2
And after that WR conflict present in the order from T2 to T1.

Precedence graph:

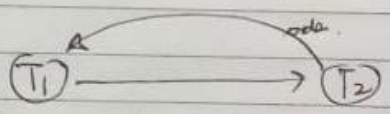
Date _____

Similarly in this case also we are taking T_1 and T_2 as node of the graph and conflicts will be taken as the directed edge of the graph.

In first RW (price) from $T_1 \rightarrow T_2$.



Now RW (payment) same as in the previous dir^n no need to show again. WR (payment) follows same no need to show one more edge in the same dir^n from $T_1 \rightarrow T_2$ after this there is ~~an edge~~ conflict of WR in order from $T_2 \rightarrow T_1$ dir^n.



Hence, there is cycle present in this graph and hence. this ~~non~~ is a non-conflicting ~~Tran~~ serializable schedule.