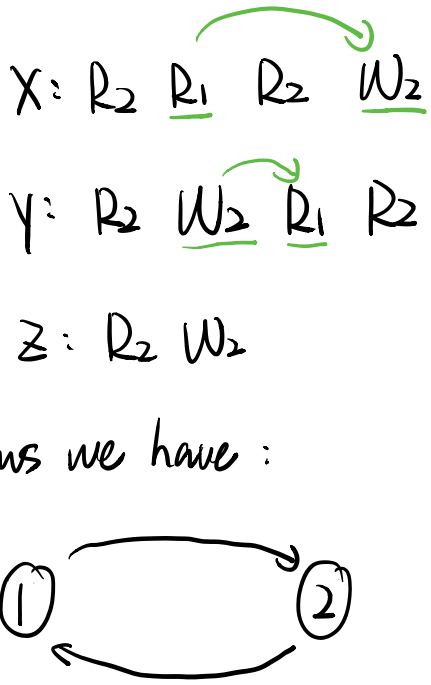
Part 1:

T_1	T_2
	R(X)
	R(Y)
	W(Y)
R(X)	
R(Y)	
	R(X)
	R(Y)
W(Y)	
	W(X)
	R(Z)
	W(Z)



S: $R_2(X)$; $R_2(Y)$; $W_2(Y)$; $R_1(X)$; $R_1(Y)$; $R_2(X)$; $R_2(Y)$; $W_1(Y)$; $W_2(X)$; $R_2(Z)$; $W_2(Z)$

(B)

If we have multiple transaction and one of the transaction only contains a single query, we may encounter the situation where this single-query transaction is working on a table that is also used by other transactions. If we do not wrap the single query statement in a transaction and utilize lock to insure the conflict serializability, the whole schedule may run into conflicts and further become non conflict serializable. Thus we do need a transaction for a single query statement.