DB TRANSACTIONS (Deadline 6)

Conflict Serializable

T1 : · Increase the fare by 1.5 times (A)
· Update Driver contact (B)

T2 : · Increase the fare by 200. (A)
· Update the driver's ~~contact~~ rating. (B)

Schedule 1:

T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	
	R(B)
	W(B)

⇒

T1	T2
R(A)	
W(A)	
R(B)	
W(B)	
	R(A)
	W(A)
	R(B)
	W(B)

Here we have finally got a serial schedule after swapping non-conflicting ~~transactions~~ operations. So, we can say that this is Conflict serializable schedule.

Non-Conflict Serializable

T1 :- Increase fare by 1.5x (A)
· Update driver's contact (B)

T2 : · Increase Fare by 200 (A)
· Update driver's rating (B)

Schedule:

T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
	R(B)
	W(B)
R(B)	
W(B)	

To convert this schedule into a serial schedule. we have to swap {R(B), W(B)} operation of T1 with {R(A), W(A), R(B), W(B)} operation of T2.
However we cannot swap these two operations as the {R(B), W(B)} of T1 and {R(B), W(B)} of T2 are conflicting operations.

Thus, the schedule is non-conflict serializable.

Vivek Jain
2021218
, Anmol Kaw
2021234.

Transactions

Transaction 1:

```
BEGIN TRANSACTION;
-- Increasing fare by 1.5x.
   Update trip
   Set fare = fare x 1.5
   where trip_id = 10;

-- Update driver's contact
   Update driver
   Set contact = 9876543210
   Where driver_id = 1 ;

COMMIT;
```

Transaction 2:

```
BEGIN TRANSACTION;

-- Increase fare by 200.
   Update trip
   Set fare = fare+200
   Where trip_id = 10.

-- Update driver's rating
   Update driver
   Set rating = 9
   where driver_id = 1 ;

COMMIT;
```