Conflict Serializable Schedule

| -- Transaction T1 BEGIN TRANSACTION; UPDATE product SET Price = 10 WHERE categoryID = 1; UPDATE product SET Price = 20 WHERE categoryID = 2; COMMIT;  -- Transaction T2 BEGIN TRANSACTION; UPDATE product SET Price = 40 WHERE categoryID = 1; UPDATE product SET Price = 30 WHERE categoryID = 2; COMMIT; |
| --- |

In this schedule, T1 and T2 can be executed in either order, and the result will be the same. This schedule is conflict serializable.

Non Conflict Serializable Schedule

Transaction 1:

| BEGIN TRANSACTION; UPDATE orders SET order\_total = quantity \* (SELECT price FROM products WHERE product\_id = orders.product\_id); COMMIT; |
| --- |

Transaction 2:

| BEGIN TRANSACTION; UPDATE products SET price = 10.00 WHERE product\_id = 1; COMMIT; |
| --- |

The two transactions can be executed concurrently without causing any conflicts or inconsistency in the data. Since there is no overlap in the data accessed by the two transactions, they can both be executed concurrently without issue.Hence, these are

non conflict serializable.