190031187
Radhakrishna

7. $r_1(x)$; $r_3(x)$; $w_1(x)$; $r_2(x)$; $w_3(x)$;

This schedule is conflict serializable as two conflicting instuctions i.e $r_3(x)$ and $w_1(x)$ is appearing one after the other. The equivalent serial schedule is

The conflicting instructions can be swapped to make it serializable. After swapping the conflicting instruction we get

$r_1(x)$; $r_3(x)$; $r_2(x)$; $w_1(x)$; $w_3(x)$;

9.

r1(x)		
	r2(z)	
r1(z)		r3(x)
		r3(y)
w1(x)		
c1		
		w3(y)
		c3
	r2(y)	
	w2(z)	
	w2(y)	
	c2	

Recoverable Schedule

The transaction which does uncommitted read operation should not commit before the commit/ rollback of the transaction which updated that data item

Non-Recoverable Schedule

Opposite of Recoverable schedule

Cascadless Schedule

No uncommitted read is allowed

strict Recoverable Schedule

Read/ write (WR, WW) operations are not allowed by any other transactions until the transaction commit/rollback which updated a data item.

SS: Recoverable, cascadeless, strict recoverable

8. Sort - Merge Join

The steps it follows are

a) Create partitions by utilizing the shuffle process

b) Sorting the data

c) Merging reducers

d) Finally generating the output.

partition Join :- It works on similar partitions when both the relations are joined by a Join key. The no.of partitions of a relation must be multiple of that of other.

N-way Map side Join

It is generally used at data warehouse where there is a Fact table and others are dimension Tables.

Simple N-way Join

when there is a large table and two comparetively smaller tables then this type of Join is performed for getting the rows for the key of all small tables which can be bufferred in memory.