10_Transaction_Schedule_Properties

Due Mar 17 at 9pm **Points** 5 **Questions** 5 **Time Limit** None **Allowed Attempts** 3

Attempt History

	Attempt	Time	Score
KEPT	Attempt 3	less than 1 minute	5 out of 5
LATEST	Attempt 3	less than 1 minute	5 out of 5
	Attempt 2	less than 1 minute	2 out of 5
	Attempt 1	5 minutes	2 out of 5

Score for this attempt: **5** out of 5
Submitted Mar 18 at 11:49am
This attempt took less than 1 minute.

Question 1 — 1 / 1 pts

TRUE or FALSE: The following schedule is conflict-serializable:

```
r3(B);r1(A);w1(A);c1;r3(A);w3(A);w3(B);c3;w2(A);c2;
```

Correct!

- (•) True

False

Question 2

1 / 1 pts

TRUE or FALSE: The following schedule is non-recoverable and conflict-serializable:

```
r3(B);r1(A);w1(A);r3(A);w3(A);w3(B);c3;c1;w2(A);c2;
```

Correct!

True

False

A non-recoverable schedule occurs when a transaction is allowed to commit its work before a transaction that it read a data value from commits. In this case, T3 reads A after T1 writes A and T3 commits before the write of w1(A) is committed. This is a non-recoverable action. **Note that the transaction read from (T1) does not have to abort for the schedule to be considered non-recoverable!**

Question 3

1 / 1 pts

TRUE or FALSE: The following schedule is cascade-free:

```
r1(A);w1(A);r3(B);r3(A);w3(A);w3(B);c1;c3;w2(A);c2;
```

- ○ True
- ◉ False

Correct!

A cascade-free schedule occurs when cascading rollback is prevented. We do not allow the read an uncommitted data item written by another transaction. The schedule is recoverable because T1 was committed before T3. However, even though T3 committed it is still not cascade-free as a failure may occur before the commit for T1 is logged, which would require cascading aborts during recovery. Cascading aborts during normal operation is only possible if the transaction that wrote the value read from aborts.

Question 4

1 / 1 pts

TRUE or FALSE: The following schedule is strict:

```
r1(A);w1(A);w2(A);c2;c1;r3(B);r3(A);w3(A);w3(B);c3;
```

- ○ True

Correct!

- False

A strict schedule implies we never read or write a non-committed value. If it is cascade-free but not strict, this requires to write a data value without reading it on a value already read or written (a blind write as in T2).

Question 5

1 / 1 pts

TRUE or FALSE: The following schedule is view serializable but not conflict-serializable:

```
r1(A);w2(A);c2;r3(B);r3(A);w1(A);c1;w3(A);w3(B);c3;
```

Correct!

- True
- False

A schedule that is view serializable but not conflict serializable is only possible if there exists a blind write as in T2. In this case, the schedule is not conflict serializable because the w2(A) conflicts with r1(A) and w1(A). However, it is view serializable to the serial schedule T1, T2, T3 because:

1. T1 is the first to read A in both schedules and T3 is first to read B in both schedules.
2. T3 reads A from T2 in both schedules.
3. T3 is last to write both A and B in both schedules.

Quiz Score: **5** out of 5