Fundamentals of Database Systems

Assignment: 7

Due Date: 15th September, 2017

Instructions

This question paper contains 10 questions in 3 pages.

Q1: Choose the correct option about the following schedule.
S: R_2(A); W_3(A); (commit T_3); W_1(A); (commit T_1); W_2(B); R_2(C); (commit T_2); R_4(A); R_4(B); (commit T_4)

A. S is both recoverable and conflict serializable

B. S is neither recoverable nor conflict serializable

C. S is recoverable but not conflict serializable

D. S is not recoverable but conflict serializable

Explanation: A is correct. Only commited data is being read in whole schedule by all transactions

Q2: Which of the following statement is correct?

A. 2 phase locking protocol does not suffer with problem of cascading rollback

B. Strict 2 phase protocol never results in deadlock

C. In strict 2 phase protocol, restriction of releasing *exclusive* locks until commit ensures no cascading rollbacks

D. Time-stamp ordering protocol can generate non-recoverable schedules

Explanation: Directly from text.

Q3: Which of the following transaction(s) follow 2-phase locking protocol?

1. $lock_1(a)read_1(a)lock_2(b)write_2(b)lock_1(c)unlock_2(b)unclock_1(c)lock_2(c)unlock_1(a)$
2. $lock_1(a)lock_1(b)unlock_1(a)lock_2(a)write_2(a)unlock_2(a)unlock_1(b)$
3. $lock_1(a)lock_2(b)lock_1(c)lock_3(d)unlock_2(b)lock_3(b)unlock_1(a)unlock_3(d)unlock_1(c)lock_3(c)$

A. Only 1

B. Only 2

C. Only 1 and 3

D. None of them

Explanation: violations
1: $unlock_2(b) \rightarrow lock_2(c)$ 3: $unlock_3(d) \rightarrow lock_3(c)$

Q4: Which of the following statement(s) is/are correct?

1. Strict 2 phase locking protocol is deadlock free
2. A schedule that follows rigorous 2 phase protocol can be serialized by the commit order of transactions
3. All strict 2 phase schedules are recoverable
4. Every rigorous 2 phase schedule is strict 2 phase schedule

A. Only 1
B. Only 3 and 4
C. 2, 3 and 4
D. All of them

Explanation: From text

Q5: Two-phase locking ensures view serialization.

A. True
B. False

Explanation: 2 phase ensures conflict serialization which in turn is view serialize; strict 2 phase schedules are cascade-less

Q6: All strict schedules are _____ and all cascadeless schedules are _____.

A. Cascadeless, Recoverable
B. Serial, Recoverable
C. Recoverable, Conflict serializable
D. None of the choices

Explanation: A is correct. All strict schedules are Cascadeless and all cascadeless schedules are recoverable

Q7: Consider the following statements:
(I) All recoverable schedules are conflict-serializable
(II) Any view serializable schedule is conflict serializable.
(III) Every cascadeless schedule is strict.
(IV) All cascadeless schedules are recoverable.

Which statement(s) is/are correct?

A. I and IV only
B. I, III and IV only
C. IV only
D. I, II and III only

Explanation: Only statement IV is correct as per the definitions.

Q8: Which of the following prevent(s) deadlock?

A. Rigorous 2-phase locking protocol

B. Conservative 2-phase locking protocol

C. Both rigorous and conservative 2-phase locking protocols

D. None of rigorous and conservative 2-phase locking protocols

Explanation: Deadlock can only be prevented by Conservative 2-Phase Locking Protocol

Q9: In wait-die scheme, transactions T_1 and T_2 have timestamps 10 and 15 respectively. If T_2 requests a data item held by T_1 then

A. T_2 will be rolled back

B. T_2 will wait

C. T_1 will be rolled back

D. T_1 will wait

Explanation: In wait-die scheme, when transaction T_i requests a data item currently held by T_j and time stamp of T_j is smaller than that of T_i, then T_i is rolled back. Thus when T_2 requests a data item held by T_1, then T_2 will be rolled back.

Q10: In a wound-wait scheme, transactions T_1 and T_2 have time stamp 2 and 5 respectively. If T_1 requests a data item held by T_2, then T_1 will be rolled back.

A. True

B. False

Explanation: In wound-wait scheme, when transaction T_i requests a data item currently held by T_j and time stamp of T_j is greater than that of T_i, then T_j is rolled back. Thus when T_1 requests a data item held by T_2, then T_2 will be rolled back.