Fundamentals of Database Systems

Assignment: 6

Due Date: 5th September, 2017

Instructions

This question paper contains 10 questions in 4 pages.

Q1: Consider the following two transactions T_1 and T_2 involving data items A and B. The values of A and B are initially 100 and 200 respectively.
T_1: read(A); read(B); $B = A + B$; write(B);
T_2: read(B); read(A); $A = A - B$; write(A);

Any non-serial interleaving of transactions T_1 and T_2 allowing concurrent execution will lead to a serial that is

A. Conflict equivalent to only T_2, T_1

B. Conflict equivalent to only T_1, T_2

C. Conflict equivalent to both the serial schedules

D. Conflict equivalent to none of the serial schedules

Explanation: D is correct. Both serial schedules $T_1 \rightarrow T_2$ and $T_2 \rightarrow T_1$, In the very first step one transaction reads the value written by other transaction so any non-serial interleaving will never lead to conflict serializable.

Q2: If a schedule is serializable, which of the following is guaranteed to give a serial schedule in the precedence graph?

A. Depth First order

B. Topological order

C. Breadth first order

D. None

Explanation: B is correct because DFS and BFS will lead to an order even if there exists a cycle in the precedence graph but toplogical sorting is only possible in cycle-free graph. Cycle in precedence graph indicates non conflict serializability.

Q3: Shadow paging recovery scheme is a _____ type of scheme.

A. redo / undo scheme

B. no-redo / undo scheme

C. redo / no-undo scheme

D. no-redo / no-undo scheme

Explanation: D is correct. Directly from slides.

Q4: Consider the following sequence of log records in the log file before the system crashed:

(start T_1), W_1(A, 3, 4), W_1(B, 1, 2), (commit T_1), (start T_2), W_2(B, 2, 7), W_2(A, 4, 8), system crash

Which of the following would be the recovery sequence in the immediate database modification scheme?

A. Undo T_2{A:=4, B:=2}, Redo T_1**{A:=4, B:=2}**

B. Redo T_1{A:=4, B:=2}, Undo T_2{A:=4, B:=2}

C. Redo T_1{A:=4, B:=2}, Redo T_2{B:=7, A:=8}

D. Undo T_2{A:=4, B:=2}, Undo T_1{B:=1, A:=3}

Explanation: A is correct. In immediate database modification scheme, when system crashes first UNDO operation is performed in reverse order of log records for the transactions having no "commit" entry and then REDO operation is done in the forward order of transactions which have "commit" entry.

Q5: Consider the following log file, created in a basic checkpointing recovery protocol environment:

(start T_1); (W_1,A,2,3); (start T_2); (W_2,B,4,5); (W_1,B,5,6); (start T_3); (commit T_1); (W_3,A,3,6); (Checkpoint, T_3,T_2); (start T_4); (W_4,A,6,7); (W_3,A,7,9); (W_4,B,6,7); (commit T_4); (start T_5); (W_5,A,9,4);

If the system crashes now, what is the correct order of recovery operations using undo-list and redo-list?

A. Redo:{(T_4,A:=7);(T_4,B:=7)}; Undo:{(T_5,A:=9);(T_3,A:=7);(T_2,no op)}

B. Redo:{(T_4,A:=7);(T_3,A:=9);(T_4,B:=7);(T_5,A:=4)}

C. Undo:{(T_5,A:=9);(T_3,A:=7);(T_2,no op)}; Redo:{(T_4,A:=7);(T_4,B:=7)}

D. Undo:{(T_5,A:=9);(T_4,B:=6);(T_3,A:=7);(T_4,A:=6)}

Explanation: C is correct. First all the Undo operations in reverse order of their log entry and then redo operations in forward order of their log entry, Nothing to be done for transactions committed before checkpoint. Only log entries upto checkpoint are read from the last entry

Q6: Which of the following is true?

A. Blind writes appear in any schedule that is view serializable but not conflict serializable

B. Blind writes appear in all view serializable schedules that are also conflict serializable

C. Blind writes appear only in conflict serializable schedules that are not view serializable

D. Blind writes appear in non conflict serializable schedules

Explanation: It is from the video, blind writes appear only in view serializable schedules and not in conflict serializable schedules.

Q7: If several concurrent transactions are executed over the same data set and the second transaction updates the database before the first transaction is finished, then which property is violated and the database is no longer consistent?

A. Consistency

B. Durability

C. Isolation

D. Atomicity

Explanation: Isolation is the property which allows multiple transactions to occur at the same time without impacting each other's execution.

Q8: Consider the following schedules involving three transactions:
S1: $r_2(x)$, $r_3(z)$, $w_3(x)$, $w_2(x)$, $w_2(y)$, $w_3(z)$, $r_1(x)$, $r_1(y)$, $w_1(x)$, $w_1(y)$
S2: $r_1(x)$, $r_2(x)$, $r_3(y)$, $w_1(x)$, $r_2(z)$, $r_2(y)$, $w_2(y)$, $w_1(z)$

Which one of the following statements is true?

A. S1 is conflict serializable and S2 is not conflict serializable

B. S1 is not conflict serializable and S2 is conflict serializable

C. Both S1 and S2 are conflict serializable

D. Both S1 and S2 are not conflict serializable

Explanation: When we make the precedence graphs of both the schedule, we can see that the precedence graph of S1 has cycle while that of S2 does not have cycle. So, only S2 is conflict serializable schedule.

Q9: Consider the following schedule:
S: $r_1(x)$, $r_2(x)$, $w_2(x)$, $w_3(x)$, $w_1(x)$

Which one of the following statements is true?

A. The schedule is conflict serializable

B. The schedule is view serializable but not conflict serializable

C. The schedule does not have blind writes

D. The schedule is neither view nor conflict serializable

Explanation: It is not conflict serializable since the precedence graph contain cycle also it does not fulfill the conditions of view serializability hence it is not view serializable.

Q10: Assume a basic checkpointing recovery protocol. Suppose the following schedule is being run:
(start, T_1); (W_1, A, 1200, 1000); (commit, T_1); (checkpoint); (start, T_2); (W_2, B, 1500, 1800); (start, T_3); (W_3, A, 1000, 500); (start, T_4); (W_4, C, 3000, 4000); (W_3, D, 3000, 2000); (commit, T_3); (W_2, A, 500, 1500);

Suppose the schedule crashes at this point. What are the undo and redo lists in the correct order?

A. Undo List: T_4, T_2; Redo List: T_1

B. Undo List: T_2, T_4; Redo List: T_3

C. Undo List: T_4, T_2; Redo List: T_1, T_3

D. Undo List: T_4, T_2; Redo List: T_3

Explanation: From the schedule we can find that transactions $T2$ and $T4$ have uncommitted write operations, so they must be undone (in reverse order i.e first $T4$ and then $T2$). And even though $T3$ has committed after writing, but it is committed after checkpoint. So, it needs to be redone. So answer is D