Ceng352 - Database Management Systems Written Assignment 3

Spring 2020

Q1 Consider the following two transactions:
T1 = R(A) W(A) R(B) W(B)
T2 = R(B) R(A) W(A) W(B)

Assume that exclusive lock (X) and unlock (U) actions are inserted by the scheduler, resulting in the following annotated transactions:
T1 = X(A) R(A) W(A) X(B) R(B) W(B), AFTER COMMIT: U(A) U(B)
T2 = X(B) R(B) X(A) R(A) W(A) W(B), AFTER COMMIT: U(A) U(B)

Consider concurrent execution of these two transactions and answer the following questions:

(a) Is conflict serializability guaranteed? Why or why not?

(b) Is deadlock possible? If so, then assuming that T1 starts first, which transaction(s) would be rolled back (aborted) under the wait-die deadlock prevention scheme?

(c) Is cascading rollback possible? If not, explain why not. If so, show a scenario that results in cascading rollback.

Now, say that lock and unlock actions are inserted in the following way instead:
T1 = X(A) R(A) W(A) X(B) U(A) R(B) W(B) U(B)
T2 = X(A) X(B) R(B) R(A) W(A) U(A) W(B) U(B)

(d) Is conflict serializability guaranteed? Why or why not?

(e) Is deadlock possible? If so, then assuming that T1 starts first, which transaction(s) would be rolled back (aborted) under the wait-die deadlock prevention scheme?

(f) Is cascading rollback possible? If not, explain why not. If so, show a scenario that results in cascading rollback.

Q2 **Timestamp-based Scheduling**

Consider the schedule below. The symbol $r_i(x)$ stands for a read by transaction T_i to item x, $w_i(x)$ stands for a write by T_i to item x and c_i stand for the commit of T_i. Suppose **timestamp-based scheduler** is used as the concurrency control protocol.

$r_1(A), r_2(B), r_3(A), w_1(A), r_2(C), w_3(B), w_2(C), c_1, r_2(A), w_3(C), c_3, w_2(B), c_2$

Operation	A			B			C		
	RTS	WTS	C	RTS	WTS	C	RTS	WTS	C

a Use the above table to give describe what happens at each operation for the given timestamps:

i. TS(T1)=1, TS(T2)=2, TS(T3)=3.

ii. TS(T1)=2, TS(T2)=3, TS(T3)=1.

Justify whether the operation is accepted or rejected and the RTS, WTS and C (commit bit) of the data items after the operation is executed (or rejected).

b Explain the importance of the commit bit, what could have changed if we don't use the commit bit.

Q3 Assume a database contains two pages P1 and P2, and 4 data items A, B, C, D. The data items A and B are stored in page P1, data items C and D are stored in page P2. Consider the following operations occurring in the given order:

- T1 writes A.
- The system flushes log to disk and flushes page P1 to disk
- T2 writes B.
- T2 writes C.
- T2 aborts.
- The systems starts checkpointing and writes the necessary log records for fuzzy checkpointing.
- The system writes all necessary log records related to rolling back T2. It writes an END record for T2.
- T3 writes D.
- T1 commits.
- The system performs all necessary operations with committing T1. It writes an END record for T1.
- T4 writes A.
- T3 writes B.
- T4 commits.
- The system crashes.

Assume that you are using **Aries recovery manager**.

(a) Write down the contents of the write-ahead-log (WAL), the transaction table and the dirty page table just before the crash. In addition, write down all pages that are in memory, showing their contents.

(b) Write down the contents of the transaction table and dirty page table right after the Analysis phase.

(c) Indicate clearly at which LSN the Redo phase starts. Explain which changes are redone during the Redo phase. Write down the contents of pages in memory at the end of the Redo phase.

(d) Write down the contents of the log at the end of the Undo phase. Also state clearly the contents of the pages in the memory, together with their pageLSNs.

SUBMISSION

You should send a pdf file, named ‘eXXXXXXX.pdf’ (your seven-digit ID number) contains your answers. You can prepare the pdf using both ‘DOCS’ or ‘Latex’, doesn’t matter.

For each question, please show the steps that you’ve followed to find the answer. **You will not get any credits from direct answers without any explanation**.