CIS560 – Database System Concepts **Name:_____**

Homework Assignment 6 [15 points] – due October 25 (in class)

1. [5 points] After a system's crash, the undo-log using nonquiescent checkpointing contains the following data.

<START T1>
<T1, X1, 1>
<START CKPT ????>
<START T2>
<T2, X2, 2>
<T1, X1, 3>
<START T3>
<COMMIT T1>
<END CKPT>
<START CKPT ???>
<T2, X2, 4>
<T3, X3, 5>
<START T4>
<COMMIT T2>
<T4, X4, 6>
<COMMIT T3>
<END CKPT>
<START T5>
<T5, X5, 7>
<START CKPT ????>
<T4, X4, 8>
CRASH !!!

   i. What are the correct values of the three <START CKPT ????> records? You have to provide three correct values for the three "????"s.

   ii. Assuming that the three <START CKPT ????> records are correctly stored in the log, according to your answer at i., show which elements are recovered by the undo recovery manager and compute their values after recovery.

   iii. Indicate what fragment of the log the recovery manager needs to read.

2. [5 points] After a system's crash, the redo-log using nonquiescent checkpointing contains the following data.

<START T1>
<T1, A, 10>
<START T2>
<T2, B, 5>
<T1, C, 7>
<START T3>
<T3, D, 12>
<COMMIT T1>
<START CKPT ????>
<START T4>
<T2, E, 5>
<COMMIT T2>
<T3, F, 1>
<T4, G, 15>
<END CKPT>
<COMMIT T3>
<START T5>
<T5, H, 3>
<START CKPT ????>
<COMMIT T5>
CRASH !!!

i. What are the correct values of the two <START CKPT ????> records? You have to provide two correct values for the two ????s.

ii. Indicate and explain what fragment of the log the recovery manager needs to read.

iii. Assuming that the two < START CKPT ??? > records are correctly stored in the log, according to your answer above, show which elements are recovered by the redo recovery manager and compute their values after recovery.

3. [5 points] The SuperSQL database system stores its undo log file in a table, with the following schema:

Log(N, T, E, V)

where **N** is the entry number (0, 1, 2, 3, …), **T** is the transaction id, **E** is the element id, and **V** is the old value. A log entry of the form <T, E, V> is simply represented by the tuple (N, T, E, V), where N is the entry number and E>0 for an ordinary element id. The log entries <START T>, <COMMIT T>, and <ABORT T> are represented by a tuple (N, T, E, null), where E=-1 for START, E=-2 for COMMIT, and E=-3 for ABORT. For example, the log:

<START T1>
<T1, X1, 55>
<START T2>
<T2, X2, 99>
<COMMIT T1>
. . . .

is represented by the table:

N	T	E	V
0	T1	-1	
1	T1	X1	55
2	T2	-1	
3	T2	X2	99
4	T1	-2	
. . .			

Recall that each transaction starts and ends at most once; for example, a sequence <START T> … <COMMIT T> … <START T> … will not occur in the log. Moreover, any update by the transaction T will occur between its <START T> and <COMMIT T>, or between <START T> and <ABORT T> respectively. Finally, once a transaction has ended in COMMIT or ABORT and the corresponding log record is on disk, the transaction has completed and does not need to be undone.

Write a SQL query that can be run during database recovery, after a system crash. The answer to your query should return a table with two attributes, **E**, and **V**, indicating which elements have to be updated with what values. You should include each element **E** at most once in your answer: otherwise it would be ambiguous how to update it.