January 11, 2018
Arnab Chakrabarti, Rihan Hai

Implementation of Databases (WS 17/18)

Exercise 7

Due until January 23, 2018, 10am.
Please submit your solution *in a single PDF file* before the deadline to the L^{2}P system!
Please submit solutions in groups of three students.

Exercise 7.1 (Schedules, Serializability, and Locking) (10 pts)

Consider the schedule given below for two transactions T1 and T2:

$$s_0 = r_0(A)w_0(A)r_1(A)r_1(B)c_1r_0(B)w_0(B)c_0$$

1. Is this schedule conflict-serializable? Explain why or why not.
2. Show how 2PL can ensure a conflict-serializable schedule for the same transactions above. Use the notation $wl_i(A)$ to indicate that transaction i acquires the lock on element A and $wu_i(A)$ to indicate that transaction i releases its lock on A.
3. Show how the use of locks without 2PL can lead to a schedule that is NOT conflict serializable.

Now consider the given schedules below and determine, if the schedules are in the classes **CSR, OCSR, CO**.:

$$\begin{aligned} s_1 &= r_1(z)r_2(x)w_1(x)r_3(y)w_3(y)r_2(z)w_2(y)w_1(z)c_1c_2c_3 \\ s_2 &= r_3(y)w_3(y)r_2(x)r_2(z)w_2(y)r_1(z)w_1(x)w_1(z)c_3c_2c_1 \\ s_3 &= r_1(z)r_3(z)w_3(x)w_2(z)c_3r_4(x)w_4(z)c_2r_5(z)c_4w_5(y)w_1(y)c_1c_5 \\ s_4 &= r_1(z)r_3(z)w_3(x)w_2(z)r_4(x)c_2w_4(z)c_4r_5(z)c_3w_5(y)c_5w_1(y)c_1 \end{aligned}$$

Exercise 7.2 (Recovery) (10 pts)

Consider the recovery scenario described in the following. There are three transactions $T1$, $T2$, and $T3$. $T1$ updates page C, $T2$ updates pages B and C, and $T3$ updates page A. At the time of crash, we have the following contents for the log:

LSN	LAST_LSN	TRAN_ID	TYPE	PAGE_ID
1	0	T1	update	C
2	0	T2	udpate	B
3	1	T1	commit	
4	begin CKPT			
5	end CKPT			
6	0	T3	update	A
7	2	T2	update	C
8	7	T2	commit	

The transaction table and dirty page table for the checkpoint are:

TRANSACTION_ID	LAST_LSN	STATUS
T1	3	commit
T2	2	active

PAGE_ID	LSN
C	1
B	2

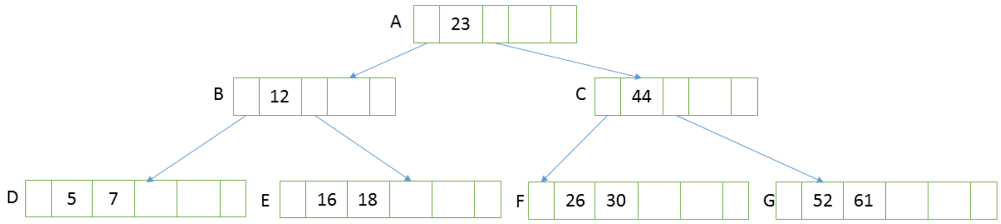
1. What is done during Analysis after the restart? Be precise about the points at which Analysis begins and ends and describe the contents of any tables constructed in this phase. **(4 pts)**

2. What is done during Redo? Be precise about the modifications to any tables due to processing of log records. **(3 pts)**

3. What is done during Undo? Be precise about the modifications to the log and any tables due to processing of log records. **(3 pts)**

Exercise 7.3 (B+-tree Locking) (10 pts)

Consider the B+-tree below, describe the steps involved in executing each of the following operations according to the simple tree locking algorithm. Redraw the result B+-tree if it is updated. Be specific about the kind of lock obtained and answer each question independently of the others, always starting with the tree above. Note that we would like to unlock a node as early as possible to maximize concurrency. We also would like to maximize throughput; i.e., releasing a higher-level node has priority over releasing a lower level node. Use the notation *rl(node), wl(node), ru(node), wu(node), r(node), w(node)* to indicate shared locking, exclusive locking, shared unlocking, exclusive unlocking, reading and writing a node respectively. Use *create(node)* to indicate creation of a new node (includes a lock on the new node) and *delete(node)* for the deletion. Use *redistribute(parent, child1,child2)* to indicate redistribution of entries between siblings and their parent. List the actions in the order they occur, and add short explanations if necessary.



1. Search 52 **(2 pts)**
2. Insert 19 **(3 pts)**
3. Delete 30 **(5 pts)**

(*Note:* Please make sure that you redraw the new tree in your solution after the insertion and deletion operation.)