November 26, 2020
Arnab Chakrabarti, Rihan Hai, Stefan Braun

Implementation of Databases (WS 20/21)

Exercise 2

Due until December 8, 2020, 10am.
Please submit your solution *in a single PDF file* before the deadline to the RWTHmoodle system!
Please submit solutions in groups.

Exercise 2.1 (Schedules, Serializability, and Locking) (10 pts)

Consider the following two transactions given in the schedule below (time goes from top to bottom).

Transaction T_0	Transaction T_1
$r_0[A]$	
$w_0[A]$	
	$r_1[A]$
	$r_1[B]$
	c_1
$r_0[B]$	
$w_0[B]$	
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.

Exercise 2.2 (Recovery) (7 pts)

$$s_1 = r_1(x)w_2(x)c_2w_1(x)c_1r_3(x)c_3$$
$$s_2 = w_1(x)w_2(x)w_2(y)c_2w_1(y)c_1r_3(y)w_3(x)w_3(y)c_3$$
$$s_3 = r_1(x)r_2(x)w_2(y)w_2(z)r_1(z)c_2c_1$$

For each of the following recoverability class, decide whether each of the three schedules is a member or not. Please explain briefly.

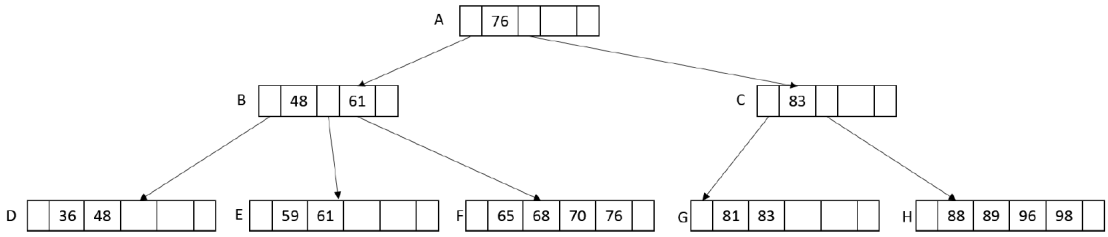
1. Strict (ST)
2. Avoid Cascading Abort (ACA)
3. Recoverable (RC)

Exercise 2.3 (ARIES Recovery Methods) **(5 pts)**

Recollect the ARIES method as discussed in the lecture. Now briefly answer the following questions. Please note that your answers must be short and to the point.

1. Describe briefly the 3 phases of ARIES recovery method.
2. What are log sequence numbers (LSNs) in ARIES? How are they used. What information does the Dirty Page Table and the Transaction Table contain?

Exercise 2.4 (B+-tree Locking) **(8 pts)**



Given the above B+-tree of degrees k=1 and k*=2, 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 *delete(node)* to indicate the deletion of a node. Use *merge(child1,child2,resultNode)* (includes writing the resultNode) to indicate merging of two siblings. List the actions in the order they occur, and add short explanations if necessary.

1. Search key = 76 **(1 pts)**
2. Insert 82 **(3 pts)**
3. Delete 61 **(4 pts)**