Extra exercises 10: Concurrency Control II

Question 1: 2PL is enough to satisfy conflict serializability, but has a disadvantage over Strict 2PL due to cascading aborts.

Question 2: With strict 2PL, if transaction A writes to row X and transaction B reads from X; and transaction A obtains the lock on X first; then transaction B has to wait until transaction A finishes writing X in order to start reading.

Question 3: For each of the following lock/unlock sequences, determine whether they conform to 2PL or Strict 2PL (or neither).

1.

LOCK_S(A)	
	LOCK_X(B)
LOCK_X(C)	
UNLOCK(A)	
	LOCK_X(A)
	UNLOCK(B)
LOCK_S(B)	
UNLOCK(C)	
UNLOCK(B)	
	UNLOCK(A)
COMMIT	
	COMMIT

2.

LOCK_S(A)	
	LOCK_S(B)
LOCK_X(C)	
LOCK_S(B)	
UNLOCK(A)	
	LOCK_X(A)
	UNLOCK(A)
	UNLOCK(B)
	COMMIT
UNLOCK(C)	
UNLOCK(B)	
COMMIT	

3.

LOCK_S(A)	
	LOCK_S(B)
LOCK_X(C)	
LOCK_S(B)	
COMMIT	
UNLOCK(A)	
UNLOCK(C)	LOCK_X(A)
UNLOCK(B)	COMMIT
	UNLOCK(A)
	UNLOCK(B)

Question 4: The list of transactions holding a single lock has a size of at most one, because each lock has an exclusive owner when locked.

Question 5: It is possible for a transaction to hold an S lock of a certain object, and then hold an X lock of the same object at a later point in the same transaction.

Question 6: A deadlock is not possible if none of the transactions try to acquire an X lock.

Question 7: In a deadlock detection graph, if there is a cycle, then we detected a deadlock and all transactions in the graph should be aborted.

Question 8:
Consider the following scenario with deadlock prevention and the following actions
Transaction A: timestamp = 5
Transaction B: timestamp = 8

Choice [[1]]	Answer	A should wait
Choice [[2]]	Answer	B should wait
Choice [[3]]	Answer	A should abort
Choice [[4]]	Answer	B should abort

Then,

1. with Wait-Die strategy…
    a. when A wants a lock that B holds
    b. when B wants a lock that A holds
2. with Wound-Wait strategy…
    a. when A wants a lock that B holds:
    b. when B wants a lock that A holds

All questions below assume that we have Multiple-Granularity Lock Protocol.

Question 9: Multiple transactions cannot hold SIX locks on the same object, since it assumes that both can obtain an X lock on the same object at the same time.

Question 10: Choose the locks acquired by the following queries, in the order of execution:

1) UPDATE stats SET value = (SELECT MAX(score) FROM students) WHERE name = “max_score”

2) UPDATE students SET latest = true WHERE id = (SELECT max(id) FROM students)

Choice [[1]]	Answer	S on table students
Choice [[2]]	Answer	IX on table stats
Choice [[3]]	Answer	X on record of table stats
Choice [[4]]	Answer	SIX on table students
Choice [[5]]	Answer	X on record of table students
Choice [[6]]	Answer	X on table students
Choice [[7]]	Answer	SIX on record of table students
Choice [[8]]	Answer	IX on record of table stats

Extra exercises 10: Concurrency Control II
Solutions

Answer 1: true
Answer 2: false (the second transaction has to wait until the first transaction either commits or aborts).
Answer 3:
1 Not 2PL
2 2PL, but not Strict
3 Strict 2PL
Answer 4: false (the lock can be owned by multiple owners as an S lock)
Answer 5: true (this can be done through upgrading)
Answer 6: true
Answer 7: false (only the transactions creating the cycle needs to be aborted)
Answer 8:

1 a [[1]]
1 b [[4]]

2 a [[4]]
2 b [[2]]

Answer 9: false (it is true that multiple transactions cannot hold SIX locks, but for another reason: one cannot hold an X lock on an object if another transaction holds an S lock – contained in a SIX lock – on one of its ancestors).
Answer 10:
1) [[1]] ➜ [[2]] ➜ [[3]]
2) [[4]] ➜ [[5]]