Assignment 2: Database Management Systems

Q1: Consider the following transactions and a relation R(a) that, before either transaction executes, contains only the tuple (1).

T1:
INSERT INTO R VALUES(2);
INSERT INTO R VALUES(3);
DELETE FROM R WHERE a=1;

T2:
SELECT SUM(a) FROM R;

Answer the following questions
a) If T2 runs with isolation level SERIALIZABLE while T1 runs with isolation level READ UNCOMMITTED, then what are the possible results returned by T2.
b) If T2 runs with isolation level READ UNCOMMITTED while T1 runs with isolation level SERIALIZABLE, then what are the possible results returned by T2.

Q2: Consider the following schedule for transactions T1, T2 and T3:

T1	**T2**	**T3**
R(X)		
	R(Y)	
		R(Y)
	W(Y)	
W(X)		W(X)
	R(X)	
	W(X)	

Write a correct serialization schedule and also draw the corresponding dependency graph.

Q3: Consider the following two phase locking protocol. Suppose a transaction T accesses (for read or write operations), a certain set of objects $\{O_1,...,O_k\}$. This is done in the following manner:
Step 1. T acquires exclusive locks to $O_1, \ldots, O_k$ in increasing order of their addresses.

Step 2. The required operations are performed.
Step 3. All locks are released.
Comment on the following
(i) Is the transaction be serializable?
(ii) Is the transaction deadlock free?
(iii) Is the transaction guaranteed to be serializable and deadlock free?

Q4: What is the equivalent serial schedule for the following transactions? Also show the corresponding precedence graph.

T1	T2	T3
		R(Y)
		R(Z)
R(X)		
W(X)		
		W(Y)
		W(Z)
	W(Z)	
R(Y)		
W(Y)		
	R(Y)	
	W(Y)	
	R(X)	
	W(X)	

Q5: A bank offers its account holders 7% interest if their balance is above Rs. 10000, else 6%. To achieve this, each of the following SQL statements is executed as separate transactions T1 and T2.
T1: update account set balance = balance * 1.6 where balance <= 10000
T2: update account set balance = balance * 1.7 where balance > 10000

What should be sequence of execution for T1 and T2?

Q6: What is the serialization order of transactions when (a) two phase locking is used for concurrency control, (b) timestamp-based is used for concurrency control?

Q.7: Suppose we log only “after images” of an updated page. Describe (a) what should happen at commit time (b) what should not happen during transaction execution time?

Q8: For better performance, a database is replicated on multiple sites. Suppose a transaction executing from a mobile laptop visits different replicas and reads and writes (parts of) the database. Design a protocol whereby a transaction can read globally-consistent data from the database replica “nearest” to it. In particular, what should a reading transaction lock? what should a writing transaction lock? Will your transaction lead to deadlocks? If so, how can you avoid it?

Q9: One way to reduce transaction conflicts is to “chop” a transaction into many components: together the chopped components produce the same (consistent) updates and outputs as the original version, but each uses less locks. Argue about the correctness of the following chopping heuristic: “Suppose transaction T accesses X and Y, but any other transaction S accesses at most one of X or Y and nothing else. Then, T can be chopped into two components, one of which accesses X and the other accesses Y.”

Q10: What is query optimization. Consider the following relations
instructor(ID, name, dept_name, salary)
teaches(ID, course_id, sec_id, semester, year)
course(course_id, title, dept_name, credits)

Write optimized relational queries for the following
i. Find the names of all instructors in the Music department, along with the titles of the courses that they teach.
ii. Find the names of all instructors in the Music department who have taught a course in 2009, along with the titles of the courses that they taught.