Concurrency control exercises

Principles of Computer System Design, Block 2, Fall 2014

Exercise 1

T1		r(x)		w(y)			C		
T2			r(y)		r(x)			C	
T3	r(x)					w(x)			C

r – read, w – write, C – commit

- What is the precedence graph for this schedule?
- Is the schedule conflict serializable?
- Could the schedule have been generated by a scheduler using Strict 2PL?

Exercise 2

T1	r(a)				w(a)						w(c)	C
T2		r(a)				r(c)	r(b)	w(b)			C	
T3				r(a)								C

r – read, w – write, C – commit

- Are there conflicts between the following pairs of transactions: {T1,T2}, {T1,T3}, {T2,T3}?
- Is the schedule conflict serializable?
- Could the schedule have been generated by a scheduler using Strict 2PL?

Exercise 3

For each statement, say whether it is true or false. Explain why.

- The ACID properties are Availability, Consistency, Integrity, and Durability
- Strict 2PL is a concurrency control protocol that has a lock acquisition phase and a lock release phase, so strict 2PL may suffer from cascading aborts
- Strict 2PL may lead to deadlocks

Exercise 4

For each statement, say whether it is true or false. If false, say why.

- Locking is a conservative approach in which conflicts are prevented and performance is improved.
- There are concurrency control protocols based on locking that prevent deadlocks altogether, and therefore these protocols do not need to implement deadlock detection.
- In optimistic concurrency control, transactions are allowed to issue writes during the Read Phase, but these writes are NOT made in-place directly to the database.