

10.03.2025: Week 4 exercises: Transactions

1) For each of the following schedules, state if they are conflict serializable or not.

Schedule S1	
T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	
C	
	R(B)
	W(B)
	C

Schedule S2	
T1	T2
R(A)	
	R(A)
	W(A)
	C
W(A)	
C	

Schedule S3		
T1	T2	T3
R(A)		
	W(A)	
	C	
W(A)		
C		
		W(A)
		C

Schedule S4		
T1	T2	T3
R(A)		
	W(A)	
	C	
W(A)		
C		
		R(A)
		C

2) Given T1, T2, T3, state which of the following schedules S1, S2, S3 are serializable and which are conflict serializable. For all the schedules that are serializable, write their equivalent serial schedules.

T1: R(X) W(X) C

T2: W(X) W(Y) C

T3: R(X) R(Y) C

Schedule S1		
T1	T2	T3
R(X)		
		R(X)
	W(X)	
W(X)		
C		
		R(Y)
	W(Y)	
	C	
		C

Schedule S2		
T1	T2	T3
	W(X)	
R(X)		
W(X)		
		R(X)
	W(Y)	
		R(Y)
C		
	C	
		C

Schedule S3		
T1	T2	T3
R(X)		
	W(X)	
		R(X)
W(X)		
		R(Y)
	W(Y)	
	C	
C		
		C

3) Is the following schedule allowed by:

- a) 2PL?
- b) Strict 2PL?

T1:	R(A)	W(A)		
T2:		R(A)	W(A) R(B) W(B)	Commit

Abort

4) Consider the following sequence of actions, listed in the order they are submitted to the DBMS:

T1: R(X), T2: W(X), T2: W(Y), T3: W(Y), T1: W(Y), T1: Commit, T2: Commit, T3: Commit

Describe how the sequence is handled by:

- a) Strict 2PL with timestamps used for deadlock prevention and Wait-Die policy;
- b) Strict 2PL with deadlock detection.

5) Are the following schedules allowed by Optimistic Concurrency Control?

Schedule S1	
T1	T2
W(B)	
R(A)	
	W(A)
	C
C	

Schedule S2	
T1	T2
	W(A)
W(A)	
C	
	C

Schedule S3	
T1	T2
R(A)	
	W(A)
R(A)	
C	
	C

6) State if the following schedule is allowed by a) Timestamp-based CC, b) MVTO.

T1	T2
R(A)	
	W(A)
	C
R(A)	
C	

7) Is the following schedule allowed by a) 2PL, b) Strict 2PL, c) Timestamp-based CC, d) OCC, e) MVTO?

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

8) Answer each of the following questions briefly. The questions are based on the following relational schema:

Emp(eid: integer, ename: string, age: integer, salary: real, did: integer)
 Dep(did: integer, dname: string, floor: integer)

and on the following update command:

replace (salary = 1.1 * EMP.salary) where EMP.ename = "Santa"

1. Give an example of a query that would conflict with this command (in a concurrency control sense) if both were run at the same time. Explain what could go wrong, and how locking tuples would solve the problem.
2. Give an example of a query or a command that would conflict with the update command such that the conflict could not be resolved by just locking individual tuples of pages but requires index locking.
3. Explain what index locking is and how it resolves the preceding conflict.