

Database Technology

Exercise 10: Transactions and Concurrency

10.1 Transactions

- Suppose that there is a database system that never fails. Is a recovery manager required?
- List the ACID properties
- Why do database-system implementers pay much more attention to ACID properties than file-system implementers?
- In which situations are concurrent executions of transactions more important?

10.2. Precedence graph

For the following situations draw the corresponding precedence graph and check if the corresponding schedule is conflict serializable. If so, give a possible schedule.

a.

T1	T2	T3
R(A)		
	R(B)	
	W(A)	
W(A)		
		W(A)

- $r1(x) r3(y) w1(x) w2(y) r3(x) w2(x)$
- $T1 \rightarrow T2, T1 \rightarrow T3, T1 \rightarrow T4, T2 \rightarrow T3, T2 \rightarrow T4, T3 \rightarrow T5, T4 \rightarrow T5$

10.3. Lost Update

The lost update anomaly is said to occur if a transaction T_j reads a data item, then another transaction T_k writes the data item (possibly based on a previous read), after which T_j writes the data item. The update performed by T_k has been lost, since the update done by T_j ignored the value written by T_k .

- Give an example of a schedule showing the lost update anomaly.
- Give an example schedule to show that the lost update anomaly is possible with the read committed isolation level.
- Explain why the lost update anomaly is not possible with the repeatable read isolation level.

10.4. Deadlock

Consider the following two transactions:

T1: read(A) read(B) if A=0 then B:=B+1 write(B)

T2: read(B) read(A) if B=0 then A:=A+1 write(A)

Add lock and unlock instructions to both transactions, so that they observe the two-phase locking protocol. Can the execution of these transactions result in a deadlock?

10.5. Two-phase locking protocol

Show that the two-phase locking protocol ensures conflict serializability, and that transactions can be serialized according to their lock points.

10.6. Rollback

Use the university database from exercise two to execute a rollback after the following update:

```
UPDATE student SET tot_cred=0 WHERE ID=00128
```

Hint: Find out how to start a transaction.