

Homework 8 – Transaction and Concurrency

Due Date: Monday April 24, 2017

In the A. Silberschatz textbook *Database System Concepts*, sixth edition, answer the following questions:

1. Page 657, 14.3: ***“Database-system implementers have paid much more attention to the ACID properties than have file-system implementers. Why might this be the case?”***

It may be due to the use of log disks that may help file systems. Alternatively, due to the possibility of needing to create software specifically designed for file systems to be on similar ground to a DBMS ACID system. Or that ACID is much more crucial for a DBMS as loss of data or work is devastating and transactions occur regularly within that system, so an ACID guideline is needed, whereas a file system may have less transactions if any, so ACID is not as crucial.

2. Page 658, 14.12: ***“List the ACID properties. Explain the usefulness of each.”***

Atomicity: Ensures that a transaction is completed in full or aborted, because having a partially executed transaction is a failure worse than not completing the transaction as it can lead to logical errors in other transactions.

Consistency: Ensures that the transaction starts in a valid state and ends in a valid state, because having the transactions follow all of the rules is important for planning as well as not throwing back any errors.

Isolation: A transaction in process and not yet committed is isolated from any other transactions. This helps prevent phenomena's such as dirty reads or phantom reads.

Durability: Committed data is saved, and in such an event like a power outage or restart, the data is still available in the correct state. Rather important, as unforeseen events may occur and any prevention of loss of work is preferable.

3. Page 659, 14.19: ***“Explain why the read-committed isolation level ensures that schedules are cascade-free, i.e., transactions will not end up in cascade rollback.”***

Cascading rollbacks occur when one transaction (T1) changes something but is not yet committed, allows another transaction (T2) to change something based upon the changed information from the previous transaction, and the first transaction (T1) aborts, so the information that the other transaction (T2) read is now inaccurate and must be rolled back as well as the previous transaction (T1).

By only reading from committed data there is no doubt in the authenticity of the data, as there is no fear of a rollback due to a possible abort in a transaction, as it has been already committed and is checked and cleared.