

Assignment 7 – Week 10 & 11

This assignment is based on lecture 9 (chapter 22).

- Submit your *own work* on time. No credit will be given if the assignment is submitted after the due date.
 - Note that the completed assignment should be submitted in .doc, .docx, .rtf or .pdf format only.
 - In MCQs, if you think that your answer needs more explanation to get credit then please write it down.
 - You are encouraged to discuss these questions in the Sakai forum.
-

- (1) _____ ensures that once transaction changes are done, they cannot be undone or lost, even in the event of a system failure.
- A. Atomicity
 - B. Consistency
 - C. Durability**
 - D. Isolation
- (2) Deadlocks are possible only when one of the transactions wants to obtain a(n) _____ lock on a data item.
- A. Binary
 - B. Shared
 - C. Exclusive**
 - D. Complete
- (3) If several concurrent transactions are executed over the same data set and the second transaction updates the database before the first transaction is finished, the _____ property is violated and the database is no longer consistent.
- A. Atomicity
 - B. Consistency
 - C. Durability
 - D. Isolation**
- (4) When a program is abnormally terminated, the equivalent of a ____ command occurs.
- A. COMMIT
 - B. ROLLBACK**
 - C. QUIT
 - D. EXIT
- (5) The deadlock state can be changed back to stable state by using _____ statement.
- A. COMMIT
 - B. ROLLBACK**
 - C. SAVEPOINT
 - D. DEADLOCK

ANS:

- (6) When transaction T_i requests a data item currently held by T_j , T_i is allowed to wait only if it has a timestamp smaller than that of T_j (that is, T_i is older than T_j). Otherwise, T_i is rolled back (dies). This is
- A. **Wait-die**
 - B. Wait-wound
 - C. Wound-wait
 - D. Wait

- (7) Explain what is meant by a transaction. Why are transactions important units of operation in a DBMS?

Transaction is action or serials of actions treated as a unit. The concept of transaction is everything or nothing is executed.

- (8) Describe, with examples, the types of problem that can occur in a multi-user environment when concurrent access to the database is allowed.

Inconsistent analysis problem, uncommitted dependency problem and lost update problem. Lost update is a problem that happens when a multiple user tries to access concurrently and as a result the changes made by one user will not be updated correctly and appropriately which will lead to the loss of update.

- (9) Give full details of a mechanism for concurrency control that can be used to ensure the types of problems discussed in the above question cannot occur. Show how the mechanism prevents the problems illustrated from occurring. Discuss how the concurrency control mechanism interacts with the transaction mechanism.

When more than one transaction is running simultaneously there are chances of a conflict to occur which can leave database to an inconsistent state. To handle these conflicts, we need concurrency control in DBMS, which allows transactions to run simultaneously but handles them in such a way so that the integrity of data remains intact.

- (10) Explain the concepts of serial, non-serial, and serializable schedules. State the rules for equivalence of schedules.

Schedule: A sequence of the operations by a set of concurrent transactions that preserves the order of the operations in each of the individual transactions.

Serial schedule: A schedule where the operations of each transaction are executed consecutively without any interleaved operations from other transactions.

Nonserial schedule: A schedule where the operations from a set of concurrent transactions are interleaved.

Serializable Schedule: A schedule which produces the same result as some serial schedule of the same transactions.

- (11) What is a timestamp? How do timestamp-based protocols for concurrency control differ from locking based protocols?

Timestamp: A unique identifier created by the DBMS that indicates the relative starting time of a transaction.

Locks may be shared (read) or exclusive (write). In two phase locking, a transaction acquires all its locks before releasing any. With timestamping transactions are ordered in such a way that older transactions get priority in the event of conflict.

- (12) What is Thomas's write rule and how does this affect the basic timestamp ordering protocol?

Thomas's write rule provides greater concurrency by rejecting obsolete write operations (i.e. Transaction T asks to write an item whose value has already been written by a younger transaction)