

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

ANS: C

- (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

ANS: C

- (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

ANS: D

- (4) When a program is abnormally terminated, the equivalent of a _____ command occurs.
- A. COMMIT
 - B. ROLLBACK
 - C. QUIT
 - D. EXIT

ANS: B

- (5) The deadlock state can be changed back to stable state by using _____ statement.
- A. COMMIT
 - B. ROLLBACK
 - C. SAVEPOINT
 - D. DEADLOCK

ANS: B

- (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

ANS: A

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

ANS:

Transaction contains single or multiple actions interacting between user or application with database that can be operated by a single unit. It acts as a unit of work while retrieving and updating the data. Database transaction must be atomic, consistent, isolated, and durable. Transaction is important to maintain the consistency in the database.

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

ANS:

- i. Lost problem occurs when one user successfully completed update operation can be overridden by another user.
- ii. Inconsistent analysis problems occurs when one transaction reads the multiple values from the database but at the same time another transaction update the some of the values of them during the execution of the first transaction.
- iii. Then uncommitted dependency or dirty read problem occurs when one transaction sees the intermediate results of another transaction before the second transaction has committed.

- (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.

ANS: When multiple transactions are running simultaneously there are chances of conflict in the database which may result inconsistent state. To handle this, we have concurrency control mechanisms which manages simultaneous operation without conflicting with each other. There is multiple way to maintain the concurrency like lock-based protocol, timestamp protocol & two-phase locking protocol.

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

ANS:

- i) Serial: It executes in a serial manner one after another where one transaction cycle completes before the start of another transaction.
- ii) Non-serial: In this schedule multiple transaction execute in a concurrent way without interfering with one another.
- iii) Serializable: It is a way to find the non-serial schedule such way that allow the transaction to execute in a concurrent way without interfering with one another.

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

ANS: It is the unique identifier used in DBMS to identify a transaction. Timestamp based protocol basically assigns a time stamp to every transaction and schedule them to execute based on the assigned timestamp. If deadlock is happened during the transaction, the transaction simply rolled back to previous state to maintain consistency. Lock based protocols basically handle the concurrency during the runtime by providing locks to the resources to the transaction before the execution of the resources. Second transaction must wait for the lock resources before the transaction finished by the first for lock to release.

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

ANS: Thomas's write rule is the enhancement of time-stamp protocol used to handle the concurrency. This rule states that if the most recent transaction has written a value to the database, then the older transaction should not write their value or change the data as it will be eventually overwritten by the newest transaction. It is slightly o than the old principle of timestamp ordering protocol.