Database Management System – cs422 DE

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

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: B

(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 Ti requests a data item currently held by Tj , Ti is allowed to wait only if it has a timestamp smaller than that of Tj (that is, Ti is older than Tj). Otherwise, Ti 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 is the process of that takes the database from one state to another. Transaction may be an action or series of action that carried out by the single user or application that change the data in database. Transaction can be success or unsuccess denoted by commit or abort.

(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: Following are the problems that can occur in multi-user environment.

1. The Lost Update Problems: When two transaction T1 and T2 is updating the same data on database there might a chance of losing the data of one transaction.

2. Dirty Read/ Uncommit Dependency Problems: When T1 and T2 executed there might be chance of getting dirty(uncommitted) data that is rollbacked by T1.

3. The Inconsistent analysis Problems: There are lots of data reads happens on database concurrently but if T2(Second Transaction) updates some datas during the execution of T1(First transaction).

(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: Following are the mechanism for concurrency control that we must implement.

1. Locking: If we have multiple Transaction, this locks the whole database for one transaction only at time. Second Transaction execute only after completion of first transaction.

2. Time Stamping: In this case, system will generate unique identifier that indicate the starting time of particular transaction. Read/Write proceeds only if last update on that data item was carried out by an older transaction.

3. Multi-Version: In this each write operation creates new version of data item while reading old version. When T1 transaction attempt to read data item, system select only one version to ensure serializability.

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

ANS:

1. Serial Schedule: Schedule where operation of each transaction is executed consecutively without any interleaved operations from other transaction.
2. Non-Serial Schedule: Schedule where operations from set op concurrent transaction are interleaved. This is the main objective of serializability that allow transaction to execute concurrently without interfering with one another.
3. Serializable Schedule: In this case read/write is most important, if two transactions (T1, T2) only read the data and they do not conflict. If two transaction (T1, T2) read/write separate data and last T1 writes the data and T2 reads or write the same data then order is important.

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

ANS: Timestamp is a unique identifier that is created by DBMS system that indicate the relative starting time of a transaction.

While locking-based protocols use locking for simultaneous transactions, timestamp-based protocols serialize the execution of concurrent transactions. In the event of a conflict, the lower timestamp is given priority in the timestamp protocol, whereas in the locking-based protocol priority is given according to the kind of lock.

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

ANS: Thomos write rule states that a modification to the basic timestamp ordering protocol that relaxes conflict serializability can be used to provide greater concurrency by rejecting obsolete write operations.

- ⇨ Transaction (T) asks to write and item whose value is already read by younger transaction that should be rollback and restart the timestamps.
- ⇨ Transaction (T) asks to write and item whose value is already written by younger transaction that means that a later transaction has already updated the value of the item and the value that the older transaction is writing must be based on an obsolete value of the item. In this case thw write operation can safely be ignored.
- ⇨ At last, as before the write operation can proceed by setting timestamp equal to time of T.