

Assignment 7 – Week 10 & 11

- (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:

A transaction is a basic logical unit of work.

Transactions are important units of operation in DBMS because of its four ACID properties; Atomicity, Consistency, Isolation and Durability.

- (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:

The types of problem that can occur in a multi-user environment when concurrent access to the database is allowed:

1. Lost Update Problem: if a record is accessed by two or more users simultaneously, it can lose one's update because it may be overwritten by other users.
e.g., if an account has \$1000 balance, user1 update it by depositing (\$500) and user2 update by withdrawing (\$300) at the same time, the balance \$700, resulting in lost update.
2. Uncommitted dependency problem: if a record is accessed by two or more users simultaneously, the result may result wrong one if one is aborts.
e.g., if an account has \$1000 balance, user1 deposited \$1000 and aborts. But when user2 withdraws \$100 before user1's transaction aborts, it could result \$1900 when the correct result should be \$900.
3. Inconsistent analyst problem: if a record is accessed by two or more users, it could make data inconsistent.
e.g., if an account has \$1000 balance, user1 reads that balance and got \$1000. At that time, user2 update that balance \$800. User1 will not know if it's changed to \$800 if it didn't read again.

- (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:

According to the above answers, if two or more users are using same data with different transactions simultaneously, the data cannot be consistent. Concurrency control mechanisms can help to avoid conflicts in that multiuser database environment.

- 2PL which consists of two phases: growing and shrinking phases – transactions will not interfere by locking the shared data item.
- Timestamp protocol – transactions will be ordered to execute based on their timestamps.

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

ANS:

Serial schedule: Transaction are executed one by one in a serial order.

Non-serial schedule: Transactions are executed concurrently so that there is no order to execute.

Serializable schedule: The non-serial schedule which ensures that the result of its concurrent execution is the same with a serial execution of the same transactions.

The rules for equivalence of schedules:

1. Read-Write equivalence
2. Write-Write equivalence
3. View serializability
4. Conflict serializability

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

ANS:

A timestamp is a unique identifier representing the time when the transaction is executed.

Timestamp-based protocols works with timestamps to decide which order the transaction should be executed while locking based protocols use locks to restrict access to shared data items.

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

ANS:

Thomas's write rule is also called as strict 2PL protocol where once a transaction has acquired an exclusive lock on the data item, it will not release that lock until that transaction completes.

With the basic timestamp ordering protocol, transactions are assigned with timestamps where they must execute, which can cause a situation where a granted transaction can be rolled back because of conflicts with other transactions.

Thomas's write rule affects the basic timestamp ordering protocol in the way where a transaction must hold all its exclusive locks until the transaction is committed or rolled back so that it can prevent conflicts with other transactions which can reduce the likelihood of transaction rollbacks and ensure the consistency.