Unit 5: (Transaction processing concepts , concurrency control techniques)

Section-A: (2 Marks)

IMPORTANT QUESTIONS

1. What is concurrency control?

When transactions are executed concurrently, there should not be any interference. This can be ensured by serializiability of schedules using protocols which are a set of rules. Concurrency control protocols use Time stamps.

2. Define Lock.

A lock is a variable associated with a data item that describes the status of the item with respect to possible operations that can be applied to it. Generally, there is one lock for each data item in the database. Locks are used as a means of synchronizing the access by concurrent transactions to the database items.

3. What are the ACID properties of transaction?

The execution of a transaction, it must satisfy some properties that are referred to as "Acid". The acronym “ACID” stands for "Atomicity", "Consistency", "Isolation" and "Durability".

4. Define transaction with Examples.

A Transaction is a Logical unit of database processing that includes one or more access operations (read -retrieval, write - insert or update, delete). A transaction can be defined as a group of tasks. A single task is the minimum processing unit of work, which cannot be divided further. An example of transaction can be bank accounts of two users, say A & B. When a bank employee transfers amount of Rs. 500 from A's account to B's account, a number of tasks are executed behind the screen. This very simple and small transaction includes several steps: decrease A's bank account from 500.

```
Open_Account(A)
Old_Balance = A.balance
New_Balance = Old_Balance - 500
A.balance = New_Balance
Close_Account(A)
```

In simple words, the transaction involves many tasks, such as opening the account of A, reading the old balance, decreasing the 500 from it, saving new balance to account of A and finally closing it. To add amount 500 in B's account same sort of tasks need to be done:

```
Open_Account(B)
Old_Balance = B.balance
New_Balance = Old_Balance + 500
```

B.balance = New_Balance
Close_Account(B)

A simple transaction of moving an amount of 500 from A to B involves many low level tasks.

5. Define two phase locking (2 PL).

The two-phase locking protocol is used to ensure the serializability in Database. This protocol is implemented in two phase

Growing Phase□
Shrinking Phase

6. What is timestamp?

A timestamp is a unique identifier created by the DBMS to identify a transaction.

□

7. What is system log?

The system maintains a log sometimes called DBMS journal to keep track of all transaction operations that affect the values of database items.

8. What are the DBMS failures?

Failures are generally classified as transaction, system, and media failures.

Section-B: (5 Marks)

1. Explain various transaction operations.

The following are the operations that the recovery manager keeps of.

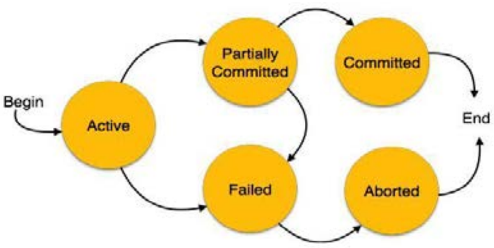
- BEGIN _ TRANSACTION
  This marks the beginning of transaction execution.
- READ OR WRITE
  These specify read or write operations on the database items that are executed as part of a transaction.
- END _ TRANSACTION
  This marks the end of transaction execution. At this point it should be checked whether the changes introduced by the transaction could be permanently applied to the database or whether the transaction has to be aborted.
- COMMIT _ TRANSACTION
  It signifies successful end of transaction so that any changes executed by the transaction can be safely committed to the database and will not be undone.
- ROLBACK (ABORT)
  This signals the transaction has ended successfully, so that any changes or effects that the transaction may have applied to the database must be undone.

2. Explain different states of transactions with Diagram.

- A transaction in a database can be in one of the following state:

Active: In this state the transaction is being executed. This is the initial state of every transaction.

Partially Committed: When a transaction executes its final operation, it is said to be in this state. After execution of all operations, the database system performs some checks e.g. the consistency state of database after applying output of transaction onto the database.



Failed: If any checks made by database recovery system fails, the transaction is said to be in failed state, from where it can no longer proceed further.

Aborted: If any of checks fails and transaction reached in Failed state, the recovery manager rolls back all its write operation on the database to make database in the state where it was prior to start of execution of transaction. Transactions in this state are called aborted. Database recovery module can select one of the two operations after a transaction aborts:

- Re-start the transaction
- Kill the transaction

Committed: If transaction executes all its operations successfully it is said to be committed. All its effects are now permanently made on database system.

3. Explain ACID properties of transaction in Detail.

The execution of a transaction Ti must satisfy some properties that are referred to as "Acid" . The acronym “ACID” stands for "Atomicity", "Consistency", "Isolation" and "Durability", which are explained below:-

Atomicity

Though a transaction involves several low level operations but this property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in database where the transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.

Consistency

This property states that after the transaction is finished, its database must remain in a consistent state. There must not be any possibility that some data is incorrectly affected by the execution of transaction..

Isolation

In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system. No transaction will affect the existence of any other transaction.

□

Durability

This property states that in any case all updates made on the database will persist even if the system fails and restarts. If a transaction writes or updates some data in database and commits that data will always be there in the database. If the transaction commits but data is not written on the disk and the system fails, that data will be updated once the system comes up.

4. Explain various types of Locks in Detail.

Several types of locks are used in concurrency control.

Binary Locks : A binary lock can have two states or values: locked and unlocked (or 1 and 0, for simplicity). A distinct lock is associated with each database item X. If the value of the lock on X is 1, item 'X cannot be accessed by a database operation that requests the item. If the value of the lock on X is 0, the item can be accessed when requested. We refer to the current value (or state) of the lock associated with item X as LOCK(X).

Two operations, lock_item and unlock_item, are used with binary locking. A transaction requests access to an item X by first issuing a lock_item(X) operation. If LOCK(X) = 1, the transaction is forced to wait. If LOCK(X) = 0, it is set to 1 (the transaction locks the item) and the transaction is allowed to access item X.

Shared/Exclusive (or Read/Write) Locks: The preceding binary locking scheme is too restrictive for database items, because at most one transaction can hold a lock on a given item. We should allow several transactions to access the same item X if they all access X for reading purposes only. However, if a transaction is to write an item X, it must have exclusive access to X. For this purpose, a different type of lock called a multiple mode lock is used. In this scheme-called shared/exclusive or read/write locks there are three locking operations: read_lock(X), write_lock(X), and unlock(X).

A lock associated with an item X, LOCK(X), now has three possible states: "read-locked," "writelocked," or "unlocked." A readlocked item is also called share-locked, because other transactions are allowed to read the item, whereas a write-locked item is called exclusive-locked, because a single transaction exclusively holds the lock on the item.