Database Management System (DBMS)

Lecture-38

Dharmendra Kumar

December 15, 2020

Transaction

- A transaction is a unit of program execution that accesses and possibly updates various data items.
- A transaction is an action or sequence of actions. It is performed by a single user to perform operations for accessing the contents of the database.

Transaction

Example: Suppose an employee of bank transfers Rs. 800 from X's account to Y's account. This small transaction contains several low-level tasks:

X's Account:

Open_Account(X)

Old_Balance = X.balance

New_Balance = Old_Balance - 800

X.balance = New_Balance

Close_Account(X)

Y's Account:

Open_Account(Y)

Old_Balance = Y.balance

New_Balance = Old_Balance + 800

Y.balance = New_Balance

Close_Account(Y)

Transaction

Operations of Transaction:

Following are the main operations of a transaction:

Read(X): Read operation is used to read the value of X from the database and stores it in a buffer in main memory.

Write(X): Write operation is used to write the value back to the database from the buffer.

Transaction

Example: Let's take an example to debit transaction from an account which consists of following operations:

1. Read(X);
2. X = X - 500;
3. Write(X);

Let's assume the value of X before starting of the transaction is 4000.

- The first operation reads X's value from database and stores it in a buffer.
- The second operation will decrease the value of X by 500. So buffer will contain 3500.
- The third operation will write the buffer's value to the database. So X's final value will be 3500.

Transaction

But it may be possible that because of the failure of hardware, software or power, etc. that transaction may fail before finished all the operations in the set.

For example: If in the above transaction, the debit transaction fails after executing operation 2 then X's value will remain 4000 in the database which is not acceptable by the bank.
To solve this problem, we have two important operations:
Commit: It is used to save the work done permanently.
Rollback: It is used to undo the work done.

Properties of Transaction

To ensure integrity of the data, we require that the database system maintain the following properties of the transactions:

1. **Atomicity:** Either all operations of the transaction are reflected properly in the database, or none are.
2. **Consistency:**
   - This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database.
   - Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the database.

Transaction

3. Isolation:

- In isolation, if the transaction T1 is being executed and using the data item X, then that data item can't be accessed by any other transaction T2 until the transaction T1 ends.
- It shows that the data which is used at the time of execution of a transaction cannot be used by the second transaction until the first one is completed.

4. Durability: After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

These properties are often called the ACID properties.

Transaction

Example: Let Ti be a transaction that transfers $50 from account A to account B. This transaction can be defined as

Ti: read(A);

A := A - 50;

write(A);

read(B);

B := B + 50;

write(B).

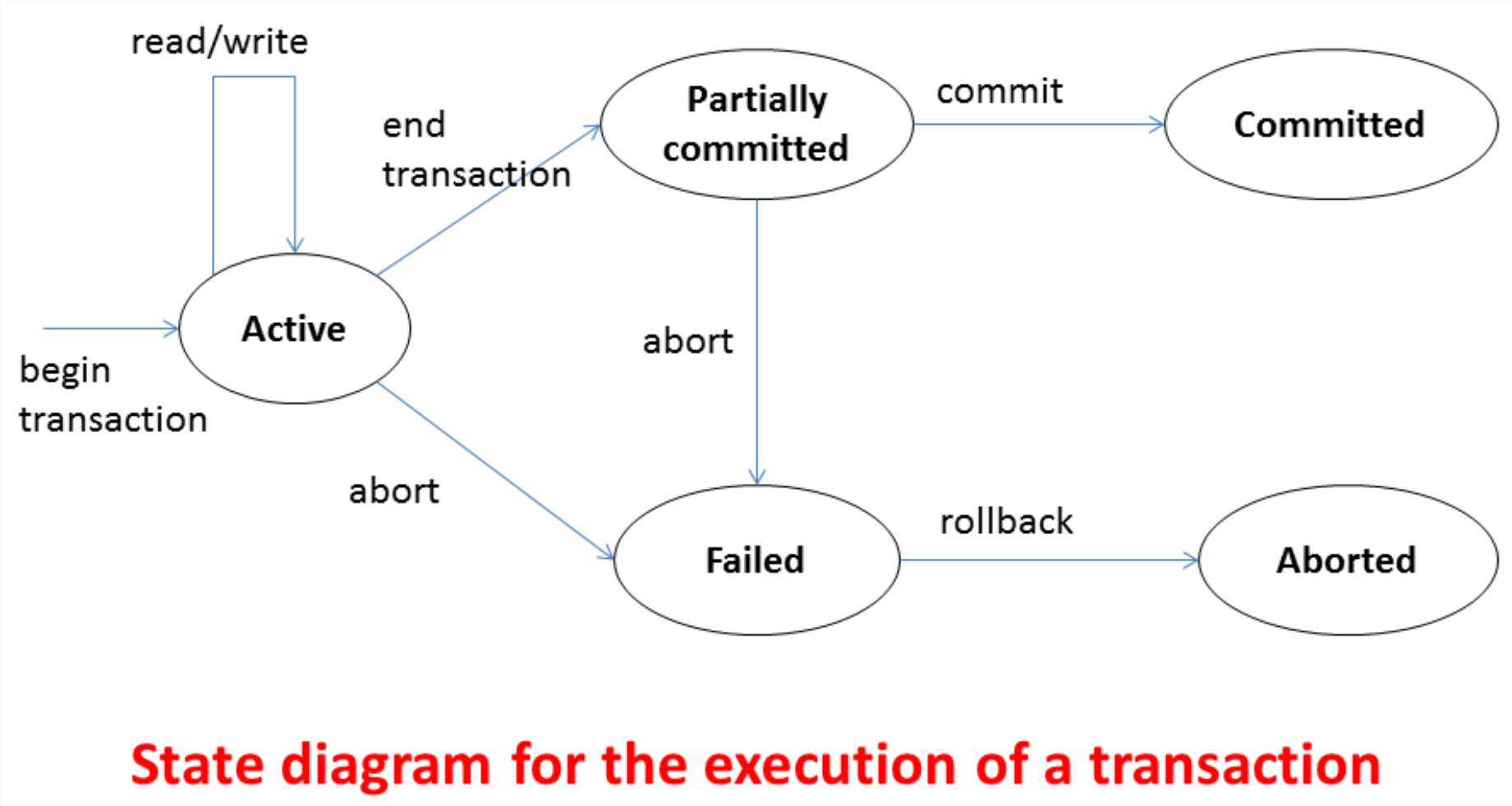
Transaction State

A transaction must be in one of the following states:

- **Active:** This is the initial state. The transaction stays in this state while it is executing.
- **Partially committed:** Transaction enter into this state after the final statement has been executed.
- **Failed:** Transaction enter into this state after the discovery that normal execution can no longer proceed.
- **Aborted:** Transaction enter into this state after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction.
- **Committed:** Transaction enter into this state after successful completion.

Transaction

The state diagram corresponding to a transaction is the following:-



State diagram for the execution of a transaction

Frame Title

When transaction enters the aborted state, at this point, the system has two options:

- It can restart the transaction, but only if the transaction was aborted as a result of some hardware or software error that was not created through the internal logic of the transaction. A restarted transaction is considered to be a new transaction.
- It can kill the transaction. It usually does so because of some internal logical error that can be corrected only by rewriting the application program, or because the input was bad, or because the desired data were not found in the database.