Database Management Systems

BITS Pilani
Hyderabad Campus

Dr.R.Gururaj
CS&IS Dept.

Lecture Session-15

Transaction Processing

innovate achieve lead

Content

- *What is Transaction Model*
- *Significance of Transaction Model*
- *States of a transaction*
- *ACID Properties*

Introduction to the Transaction Model

Multi-user Database systems:
Multiple users access the database simultaneously (multi processing).

Concurrency There occurs concurrency in data access.

Transaction:

A transaction is a collection of operations that perform a single logical operation or function in a database application.

Each transaction is a unit of *atomicity*.

Storage Types

Volatile Storage: Ex- main memory, cache
Nonvolatile storage : Ex- disk, tapes, etc.

Storage Hierarchy

DB system resides on nonvolatile storage.

Database is partitioned into blocks of fixed length storage, which are units for storage allocation and transfer.

Transactions input and output data from disk to main memory, and main memory to the disk.

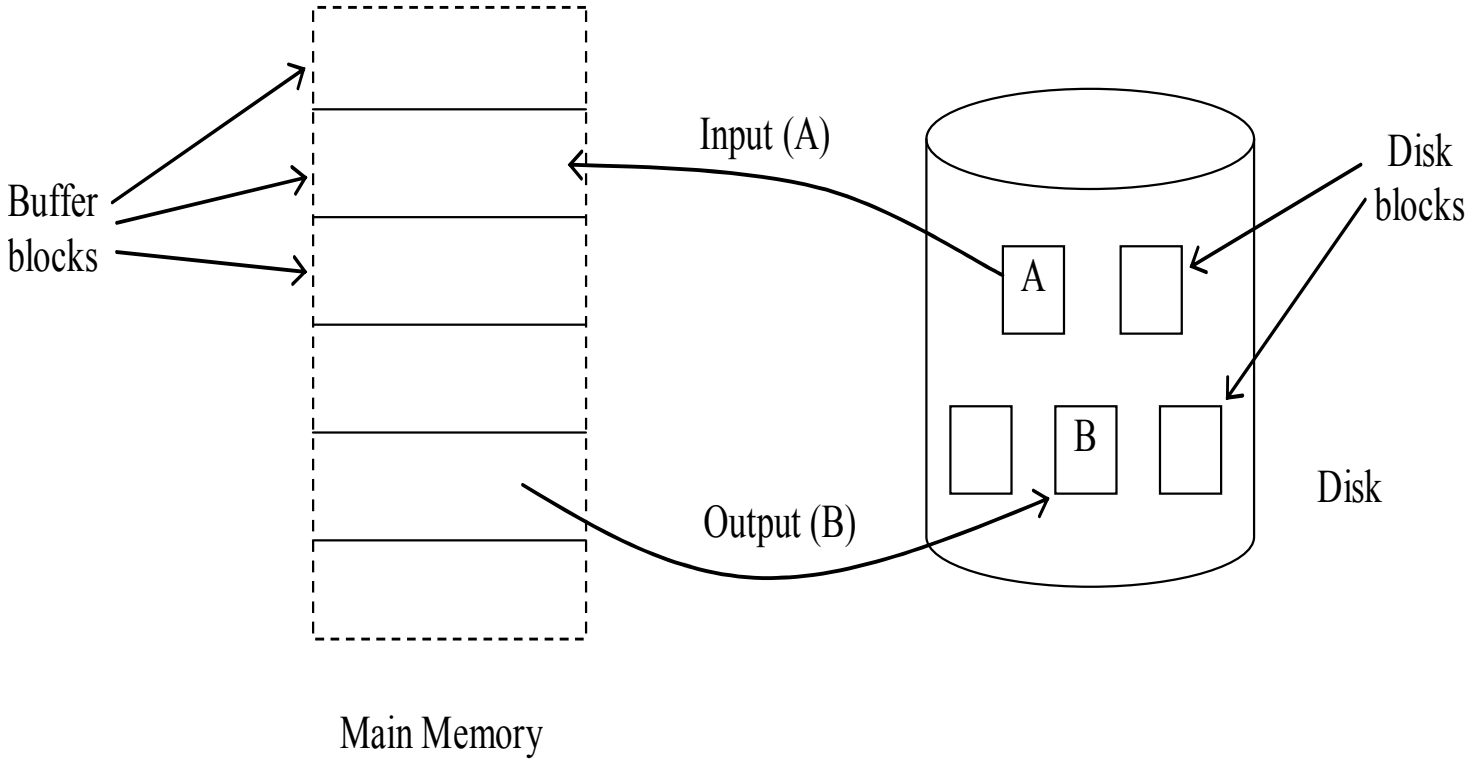
The data transfer is done in terms of blocks.

A buffer block is same as disk block but it is in main memory, but the size is same.

The block movement between disk and main memory is initiated by following operations.

Input (X) – The physical block with data item *X* is brought from disk into main memory.

Output (X) – Buffer block containing the data item *X* is sent to disk to replace the appropriate physical block.


Buffer
blocks
Input (A)
Disk
blocks
A
B
Output (B)
Disk
Main Memory

Transactions interact with DB by transferring data from program variables to the DB and DB to program variables.

This transfer of data is achieved through the following two operations.

I.read (X, x_i) -where x_i is local variable *X* is DB data item and represents the operation

$x_i \leftarrow X$

If the block with data item *X* is not in main memory issued ***Input (X)***
Assign x_i the value of *X* from buffer blocks

II. *write (X, x_i)* performs $x_i \rightarrow X$
If block with *X* is not there ***Input (X),*** assign x_i to *X* in the buffer block

Note: Reading is must but transaction need not write every item. The modified blocks can be written back onto the disk during page replacement in main memory.

Steps followed by Transactions while accessing data for processing

Read (X, x_i) → uses (***Input (X)***)

↓

Modify x_i

↓

Write (X, x_i) → uses (O***utput (X)***)

If the system crashed before the new value is written to the disk, then the new value is lost forever and never written to the disk.

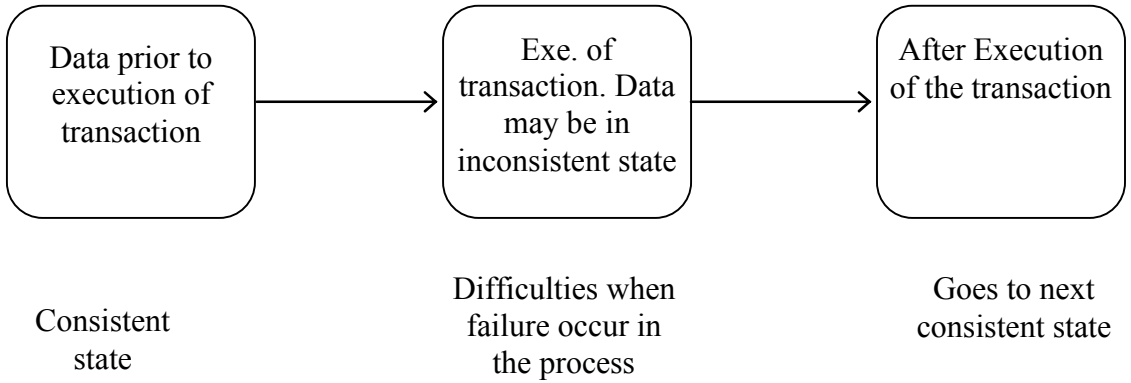
A *transaction* is an atomic unit of work that is either completed in its entirety or not done at all.

- For recovery purposes, the system needs to keep track of when the transaction starts, terminates, and commits or aborts.

Transaction Model

A *transaction* is a program unit that access and update several data items.

Read ( ) and *Write ( )* are the basic operations.



Hence, as a result of failure, state of the system will not reflect the state of the real world that the database is supposed to capture.

We call that state as *inconsistent state*.

It is important to define transactions such that they preserve consistency.

ACID Properties of a Transaction

Transaction should possess the following properties called as ***ACID*** properties.

***A**tomicity:* A transaction is an atomic unit of processing. It is either performed in its entirety or not performed at all.

***C**onsistency Preservation:* The successful execution of a transaction takes the database from one consistent state to another.

***I**solation:* A transaction should be executed as if it is not interfered by any other transaction.

***D**urability:* The changes applied to the data by a transaction must be permanent.

Transaction States

Active State: Initial state when a transaction starts.

Partially committed State: This state is reached when the last statement is executed and the outcome is not written to the DB.

Failed State: After discovering that the normal execution cannot be continued a transaction is aborted and reaches failed state.

Comitted State: Is reached after successful completion of the transaction.

Terminated State: Is reached after failure or success of the transaction.

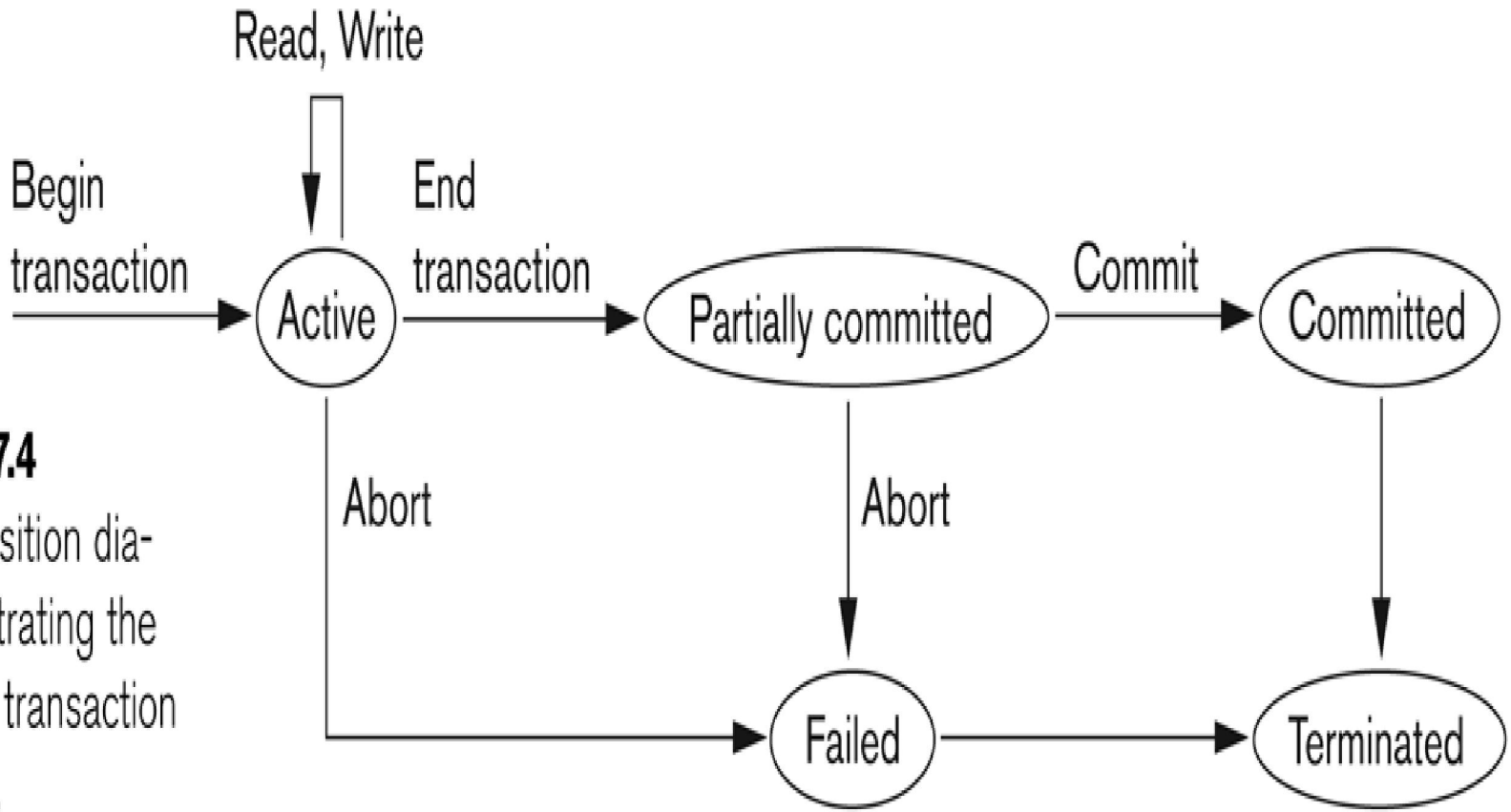


Figure 17.4
State transition diagram illustrating the states for transaction execution.

Those transactions which are not completely successful are called *failed transactions*.

In order to ensure atomicity property, failed transactions should have no effect on the database.

Hence the state of the DB must be restored to the state it was in just before the transaction started its execution.

We say such transaction is *rolled back*.

When the transaction rolls back the modifications done to the DB by the half compete transaction are removed so that the state reflects the state before the start of execution of the transaction.

A transaction which is completely successful is called *committed transaction*.

A committed transaction brings the DB to new consistent state. The effects of committed transactions cannot be rolled back.

Once a transaction is aborted, it must be terminated and new transaction must be started.

A transaction reaches committed state if it has partially committed and it is guaranteed that it will never be aborted.

Summary

- ✓ *What is a transaction*
- ✓ *Basic database operations performed by a transaction*
- ✓ *Properties of a transaction*
- ✓ *States of a transaction*
- ✓ *Transaction execution and the database consistency*