Course code : **CSE2007**
Course title : **Database Management System**
Module : **5**
Topic : **1**

Introduction to Transaction Processing Concepts

Objectives

This session will give the knowledge about

- Introduction to Transactions
- Operations on transaction processing
- Concurrent execution
- Problems in concurrent execution

Introduction

1. Begin Transaction
2. read balance(a)
3. compute balance(a)-100
4. write balance(a)
5. read balance(b)
6. compute balance(b)+100
7. write balance(b)
8. commit

Account A
(Balance:500)

Account B
(Balance:100)

Introduction

Transaction

Logical unit of database processing that includes one or more access operations (read -retrieval, write - insert or update, delete).

A transaction (set of operations) may be stand-alone specified in a high level language like SQL submitted interactively, or may be embedded within a program.

Example

Fund Transfer, Booking Systems etc

Introduction

Transaction processing systems

Systems with large databases and hundreds of concurrent users require high availability and fast response time.

Transaction boundaries

- Begin and End transaction.

An application program may contain several transactions separated by the Begin and End transaction boundaries.

Transaction Processing

Single-user DBMS

- At most one user at a time can use the system
- Example: home computer

Multiuser DBMS

- Many users can access the system (database) concurrently
- Example: airline reservations system

Transaction Processing

Multiprogramming

- Allows operating system to execute multiple processes concurrently
- Executes commands from one process, then suspends that process and executes commands from another process, etc.

Interleaved processing:

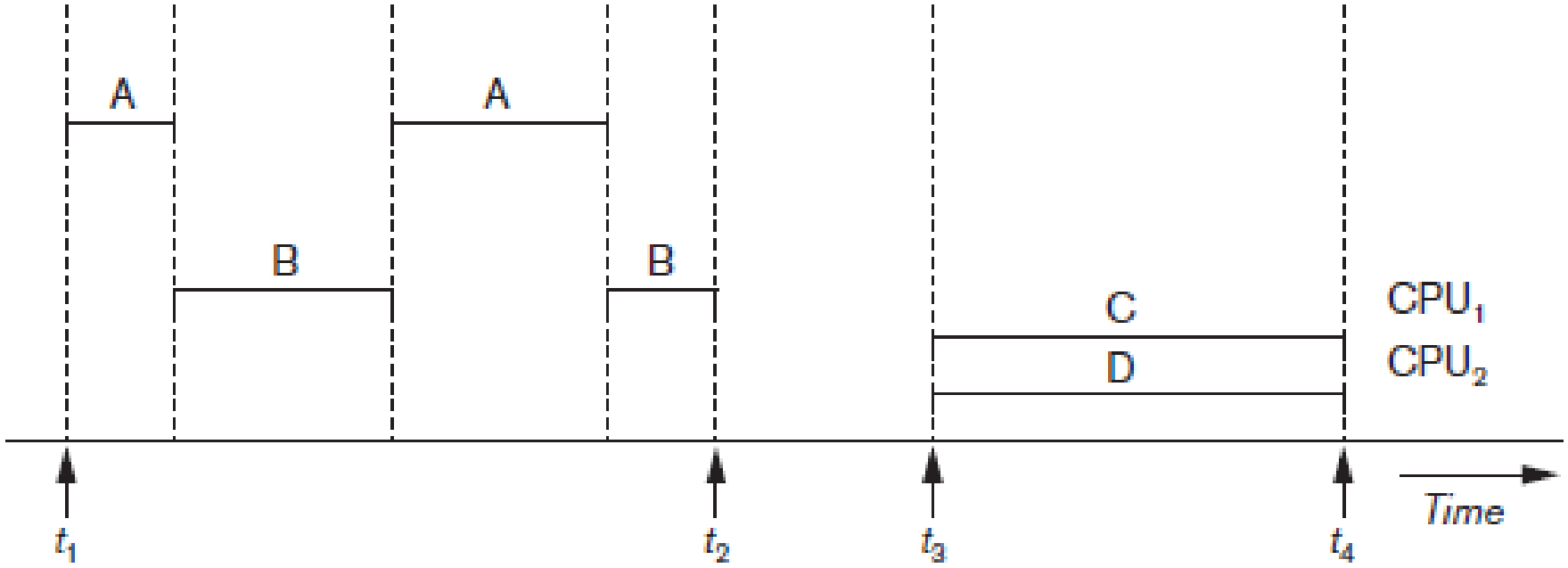
- Concurrent execution of processes is interleaved in a single CPU

Parallel processing:

- Processes are concurrently executed in multiple CPUs.

Transaction Processing

Interleaved processing versus parallel processing of concurrent transactions



Database Items

Database represented as collection of named data items

Size of a data item called its granularity

- Data item
- Record
- Disk block
- Attribute value of a record

Transaction processing concepts independent of item granularity

Read and Write Operations

Read and Write Operations are the basic unit of data transfer from the disk to the computer main memory is one block.

read_item(X)

Reads a database item named X into a program variable named X.

- Find the address of the disk block that contains item X.
- Copy that disk block into a buffer in main memory (if that disk block is not already in some main memory buffer).
- Copy item X from the buffer to the program variable named X.

Read and Write Operations

write_item(X)

Writes the value of program variable X into the database item named X

- Find the address of the disk block that contains item X.
- Copy that disk block into a buffer in main memory (if that disk block is not already in some main memory buffer).
- Copy item X from the program variable named X into its correct location in the buffer.
- Store the updated block from the buffer back to disk (either immediately or at some later point in time).

Read and Write Operations

Two sample transactions (a) Transaction T1 (b) Transaction T2

(a)

T_1

read_item(X);
$X := X - N$;
write_item(X);
read_item(Y);
$Y := Y + N$;
write_item(Y);

(b)

T_2

read_item(X);
$X := X + M$;
write_item(X);

DBMS Buffers

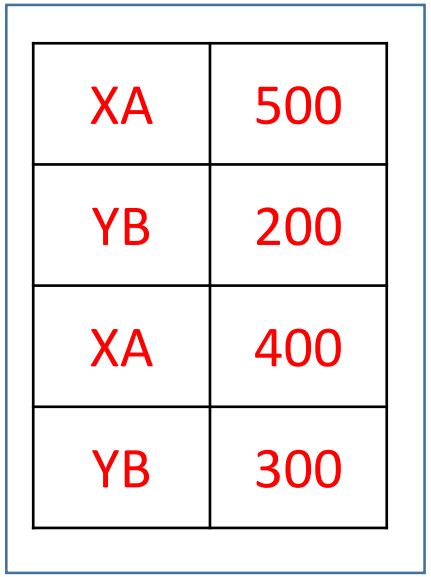
DBMS will maintain several main memory data buffers in the database cache

When buffers are occupied, a buffer replacement policy is used to choose which buffer will be replaced

Example policy: least recently used

DBMS Buffers

- Variable XA
- Read value X into XA
- XA=XA-100
- Write XA
- Variable YB
- Read value Y into YB
- YB=YB+100
- Write YB
- Commit



Main Memory

Secondary Memory

A	500
B	200

Concurrency Control

Transactions submitted by various users may execute concurrently.

Access and update the same database items.

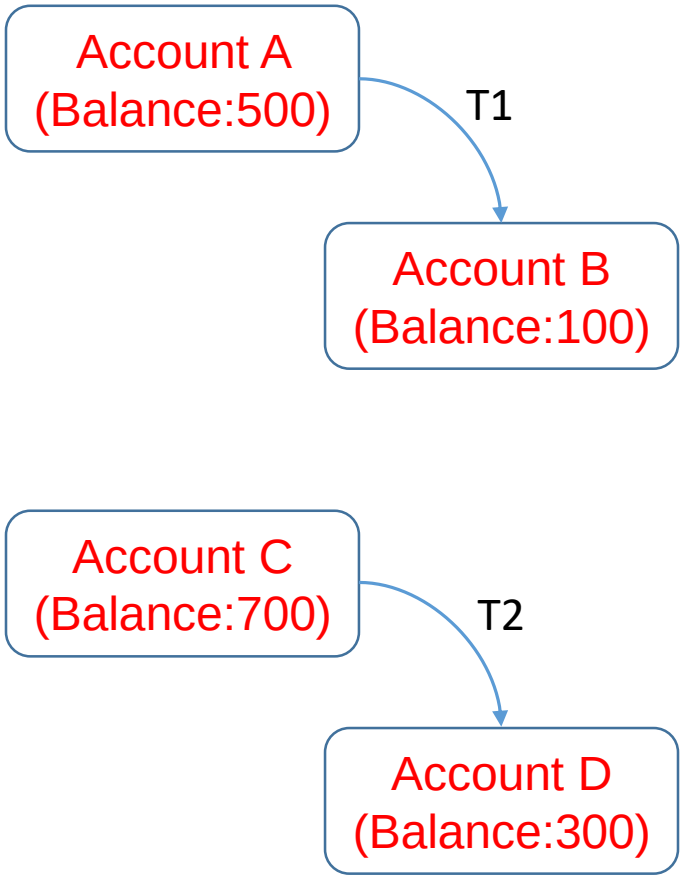
Some form of concurrency control is needed.

Problems in concurrency control:

- The Lost Update Problem
- The Temporary Update (or Dirty Read) Problem
- The Incorrect Summary Problem

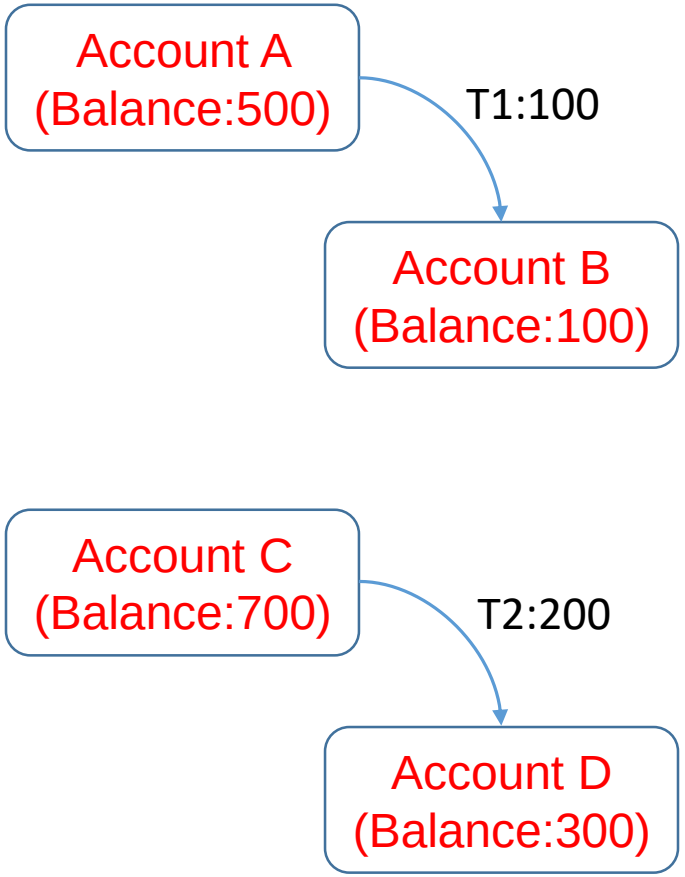
Transaction Schedule: Serial execution

Time	T1	T2
TS1	Begin Transaction	
TS2	read balance(a)	
TS3	compute balance(a)-100	
TS4	write balance(a)	
TS5	read balance(b)	
TS6	compute balance(b)+100	
TS7	write balance(b)	
TS8	commit	
TS9		Begin Transaction
TS10		read balance(c)
TS11		compute balance(c)-200
TS12		write balance(c)
TS13		read balance(d)
TS14		compute balance(d)+200
TS15		write balance(d)
TS16		commit



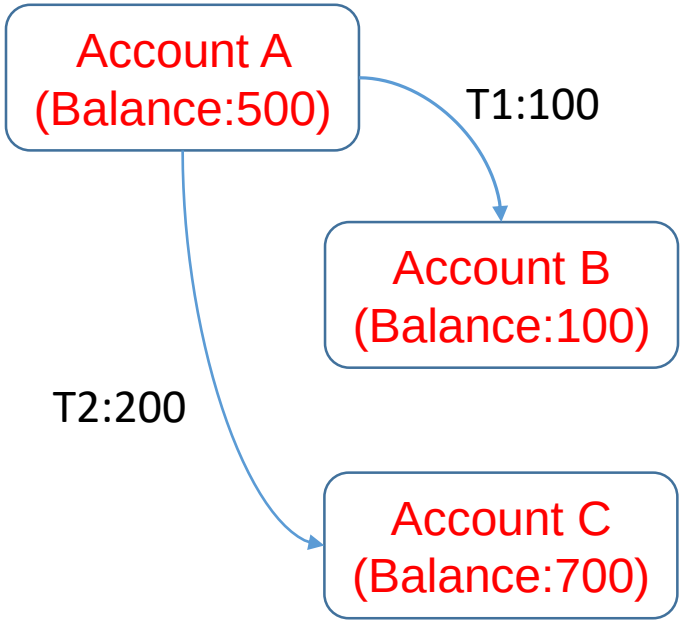
Transaction Schedule: Parallel execution

Time	T1	T2
TS1	Begin Transaction	
TS2	read balance(a)	Begin Transaction
TS3		
TS4	compute balance(a)-100	read balance(c)
TS5		
TS6	write balance(a)	compute balance(c)-200
TS7	read balance(b)	
TS8		write balance(c)
TS9		read balance(d)
TS10	compute balance(b)+100	
TS11		compute balance(d)+200
TS12	write balance(b)	write balance(d)
TS13	commit	commit



The Lost Update Problem

Time	T1	T2
TS1	Begin Transaction	
TS2	read balance(a)	Begin Transaction
TS3	compute balance(a)-100	
TS4		read balance(a)
TS5	write balance(a)	
TS6		compute balance(a)-200
TS7	read balance(b)	
TS8		write balance(a)
TS9	compute balance(b)+100	
TS10		read balance(c)
TS11	write balance(b)	
TS12		compute balance(c)+200
TS13	commit	write balance(c)
TS14		Commit



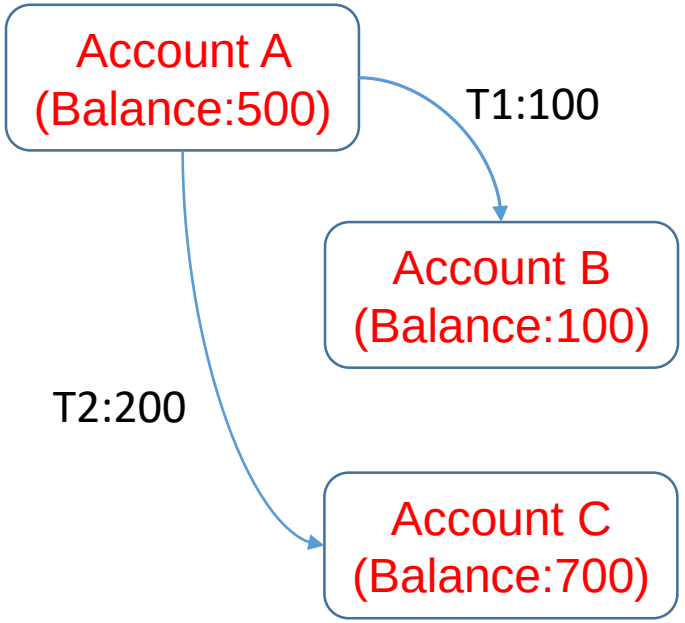
The Lost Update Problem

This occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database item incorrect.

T2 reads the value of A before T1 changes it in the database, and hence the updated value resulting from T1 is lost

The Temporary Update (or Dirty Read) Problem

Time	T1	T2
TS1	Begin Transaction	
TS2	read balance(a)	Begin Transaction
TS3	compute balance(a)-100	
TS4	write balance(a)	
TS5		read balance(a)
TS6		compute balance(a)-200
TS7	read balance(b)	
TS8	Fails (Abort)	
TS9		write balance(a)
TS10		read balance(c)
TS11		compute balance(c)+200
TS12		write balance(c)
TS13		Commit



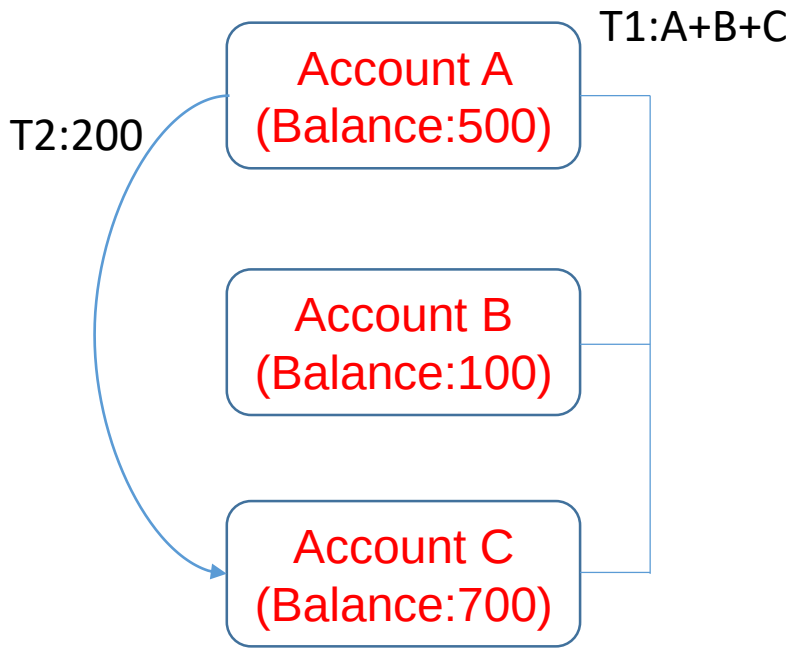
The Temporary Update (or Dirty Read) Problem

This occurs when one transaction updates a database item and then the transaction fails for some reason.

The updated item is accessed by another transaction before it is changed back to its original value.

The Incorrect Summary Problem

Time	T1	T2
TS1	Begin Transaction	
TS2	read balance(a)	Begin Transaction
TS3		read balance(a)
TS4	Sum = Sum + balance(a)	
TS5		compute balance(a)-200
TS6	read balance(b)	
TS7	Sum = Sum + balance(b)	
TS8		write balance(a)
TS9		read balance(c)
TS10		compute balance(c)+200
TS11		write balance(c)
TS12	read balance(c)	Commit
TS13	Sum = Sum + balance(c)	

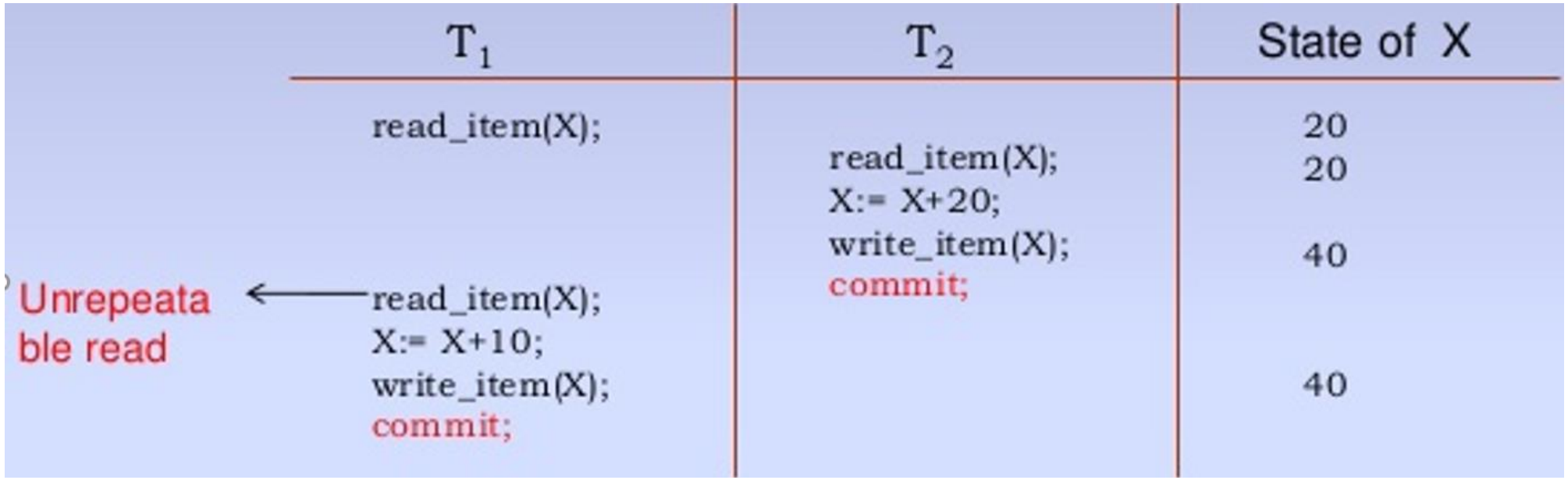


The Incorrect Summary Problem

If one transaction is calculating an aggregate summary function on a number of records while other transactions are updating some of these records, the aggregate function may calculate some values before they are updated and others after they are updated.

The Unrepeatable Read Problem

Where a transaction T reads the same item twice and the item is changed by another transaction T- between the two reads. Hence, T receives different values for its two reads of the same item.



Summary

This session will give the knowledge about

- Introduction to Transactions
- Operations on transaction processing
- Concurrent execution
- Problems in concurrent execution