Database Management System – 37 Transaction Processing Introduction

Ajay James
Asst. Prof in CSE
Government Engineering College Thrissur

Outline

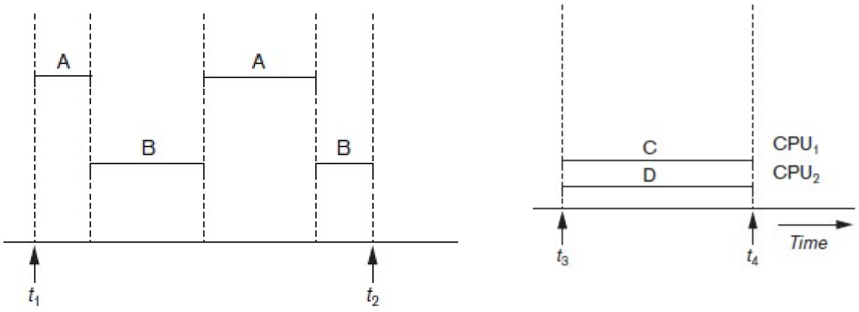
- Introduction
- Transactions
- Database model
- Why concurrency control is needed?
- Why recovery is needed?

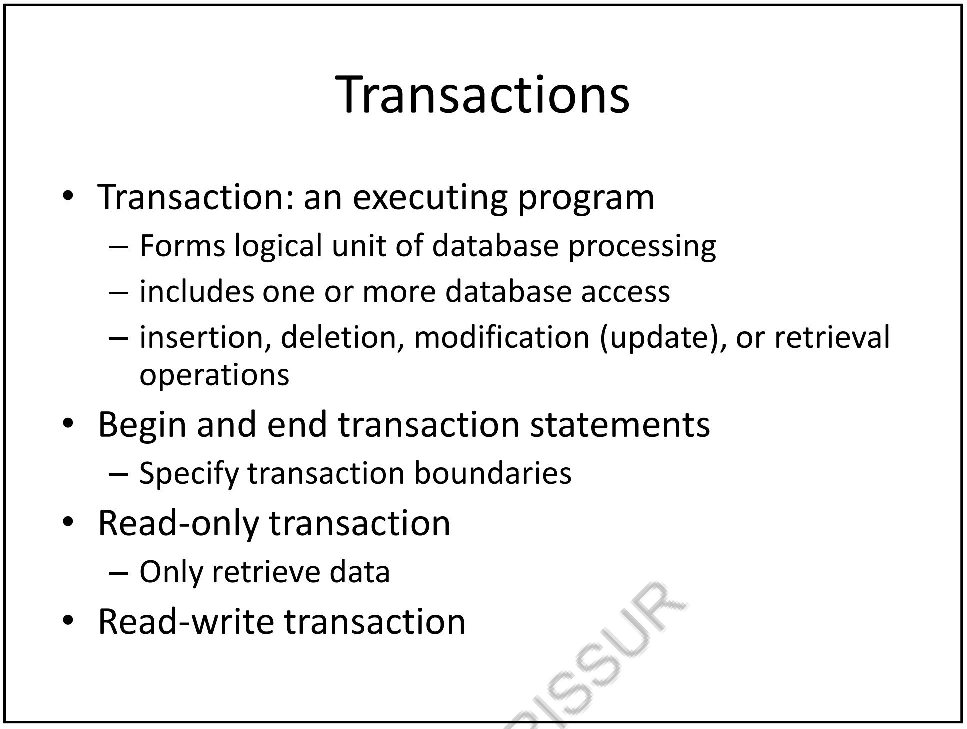
Introduction

- Transaction
  - Describes local unit of database processing
- Transaction processing systems
  - Systems with large databases and hundreds of concurrent users
  - Require high availability and fast response time
  - Examples airline reservation, banking etc.

Introduction to Transaction Processing

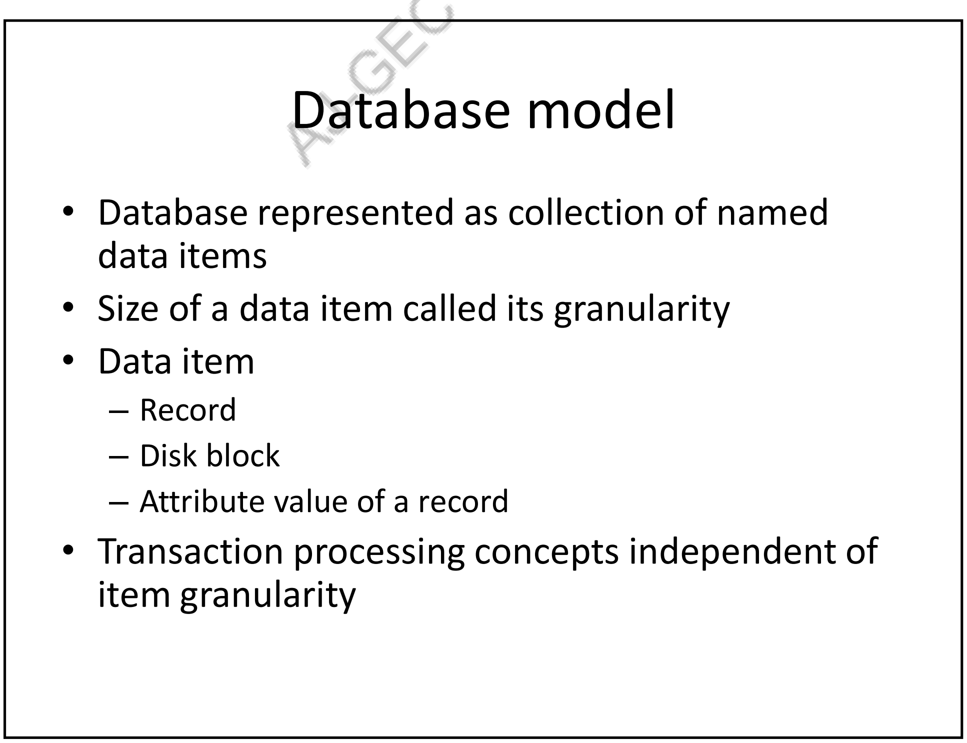
- Single-user DBMS
- Multiuser DBMS
- Multiprogramming





Transactions

- Transaction: an executing program
  - Forms logical unit of database processing
  - includes one or more database access
  - insertion, deletion, modification (update), or retrieval operations
- Begin and end transaction statements
  - Specify transaction boundaries
- Read-only transaction
  - Only retrieve data
- Read-write transaction



Database model

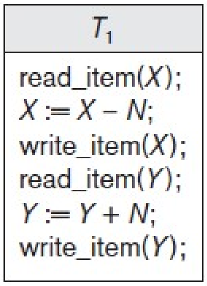
- Database represented as collection of named data items
- Size of a data item called its granularity
- Data item
  - Record
  - Disk block
  - Attribute value of a record
- Transaction processing concepts independent of item granularity

Read and Write Operations

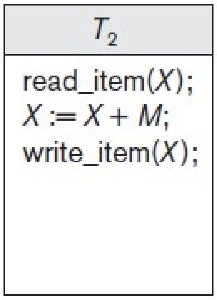
- ***read_item(X)***
  - Reads a database item named X into a program variable named X
  - Process includes finding the address of the disk block, and copying to and from a memory buffer
- ***write_item(X)***
  - Writes the value of program variable X into the database item named X
  - Process includes finding the address of the disk block, copying to and from a memory buffer, and storing the updated disk block back to disk

Read and Write Operations

- Read set of a transaction
  - Set of all items read ({X,Y} in T1)
- Write set of a transaction
  - Set of all items written ({X,Y} in T1)



T_1
read_item(X);
$X := X - N$;
write_item(X);
read_item(Y);
$Y := Y + N$;
write_item(Y);



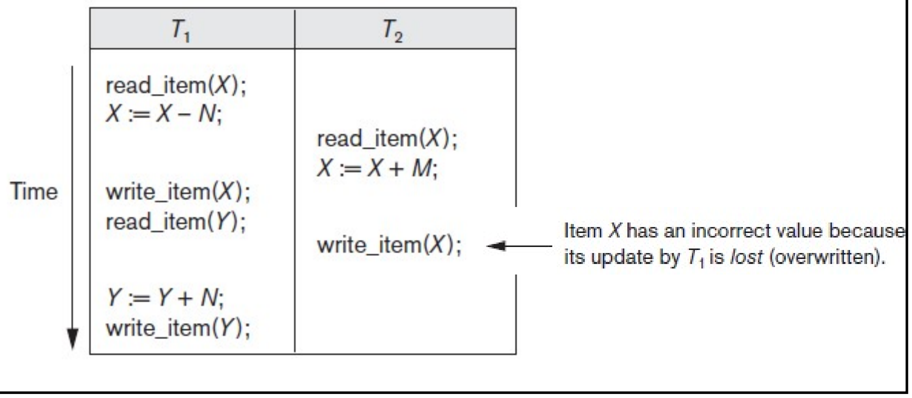
T_2
read_item(X);
$X := X + M$;
write_item(X);

Why Concurrency Control Is Needed

- Transactions submitted by various users may execute concurrently
  - Access and update the same database items
  - Some form of concurrency control is needed
- Lost update problem
- Temporary Update (or Dirty Read) Problem
- Incorrect Summary Problem
- Unrepeatable Read Problem

The Lost Update Problem

- Occurs when two transactions that access the same database items have operations interleaved
- Results in incorrect value of some database items



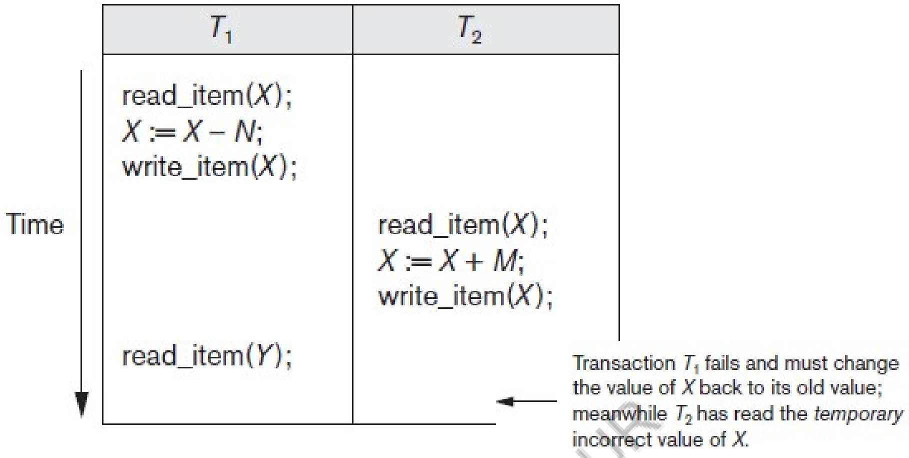
T_1	T_2
read_item(X); $X := X - N$;	
	read_item(X); $X := X + M$;

write_item(X);
read_item(Y);	

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

Item X has an incorrect value because its update by T_1 is *lost* (overwritten).

The Temporary update(dirty read) problem



T_1	T_2

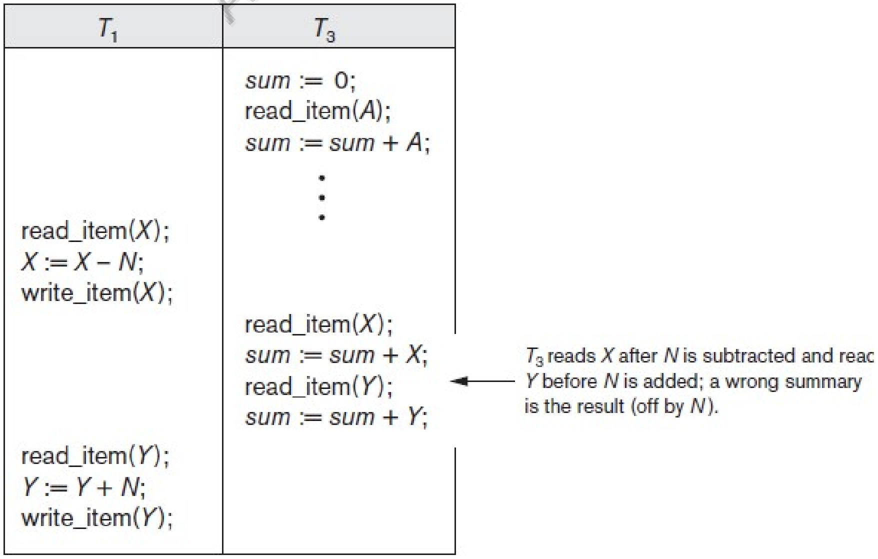
read_item(X);
$X := X - N$;
write_item(X);	

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

read_item(Y);	

Transaction T_1 fails and must change the value of X back to its old value; meanwhile T_2 has read the *temporary* incorrect value of X.

The Incorrect Summary Problem



T_1	T_3
	$sum := 0$; read_item(A); $sum := sum + A$; ⋮

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

	read_item(X);
	$sum := sum + X$;
	read_item(Y);
	$sum := sum + Y$;

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

T_3 reads X after N is subtracted and reac Y before N is added; a wrong summary is the result (off by N).

The Unrepeatable Read Problem

- Transaction T reads the same item twice
- Value is changed by another transaction T’ between the two reads
- T receives different values for the two reads of the same item

Why Recovery Is Needed

- Committed transaction
  - Effect recorded permanently in the database
- Aborted transaction
  - Does not affect the database
- Types of transaction failures
  - Computer failure (system crash)
  - Transaction or system error
  - Local errors or exception conditions detected by the transaction

Why Recovery Is Needed

- Types of transaction failures (cont'd.)
  - Concurrency control enforcement
  - Disk failure
  - Physical problems or catastrophes
- System must keep sufficient information to recover quickly from the failure
  - Disk failure or other catastrophes have long recovery times

Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6th edition and 7th edition

Thank you