Database Management Systems

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Sai University

Lecture 19, 3 November 2023

Transactions

Application form – Assign a new App ID to each applicant

?

DB has the current max ID

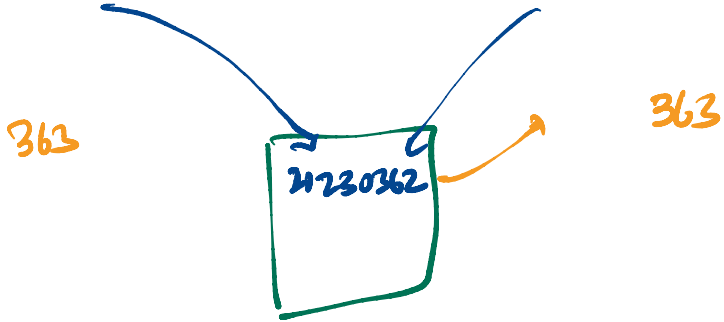
20230001
20230002
⋮
20230362

20230363

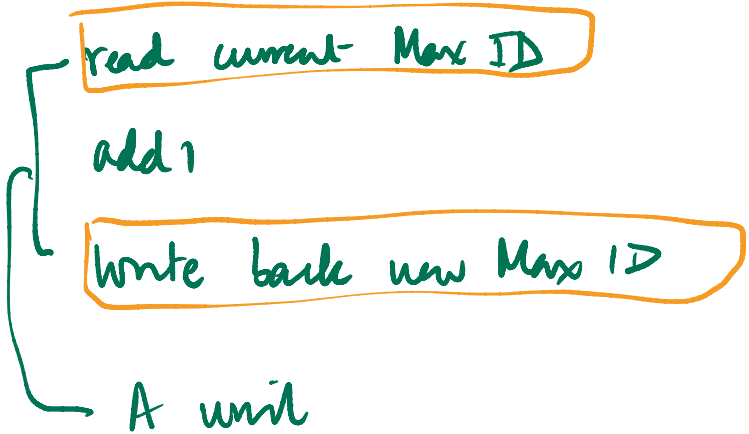
20230363

Transactions

Two applicants access in parallel

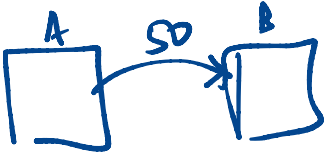


Transactions



Transactions

Bank transfer



Reduce A's balance by 50

Increase B's balance by 50

UNIT

read( ) transfer from disk to memory

write( ) transfer from memory to disk

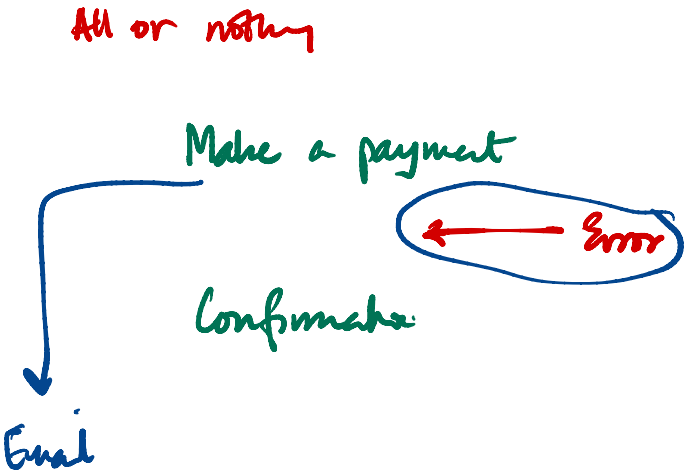
Transactions

Transaction is a unit

All or nothing ...

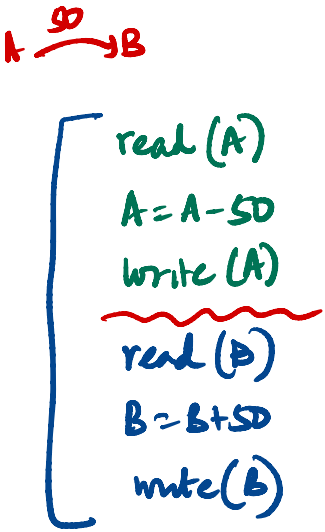
Desirable properties

- Atomicity



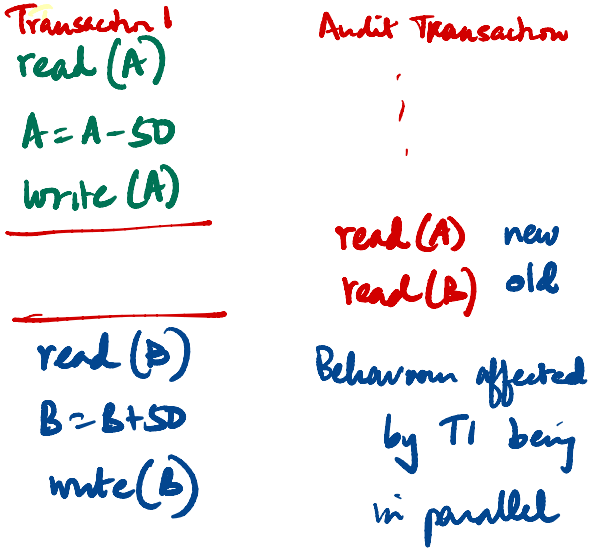
Desirable properties

- Atomicity
- Consistency



Desirable properties

- Atomicity
- Consistency
- Isolation



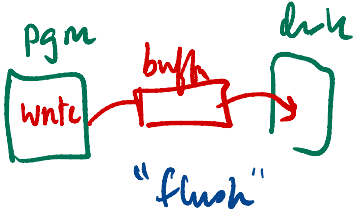
Desirable properties

- Atomicity
- Consistency
- Isolation
- Durability

Effect is permanent

Volatile & non volatile storage

↓ Memory ↓ Disk



"Committing" a transaction

RAID disks

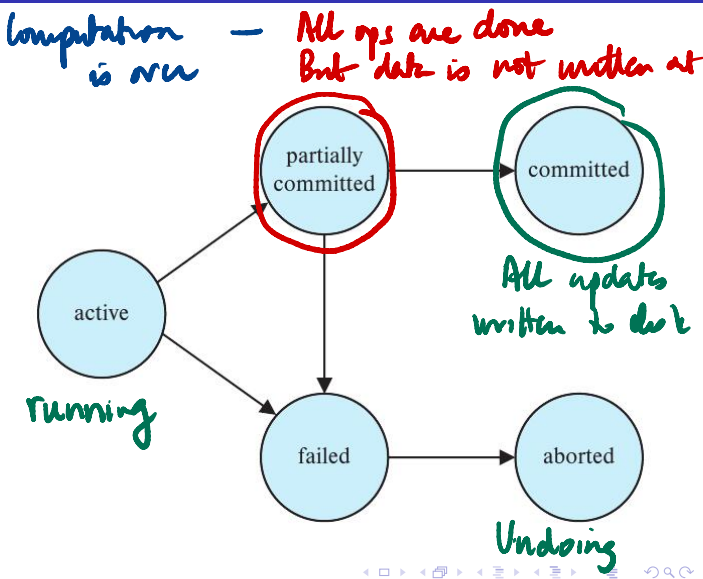
Desirable properties

- Atomicity
- Consistency
- Isolation
- Durability
- ACID properties

States of a transaction

Sequence of actions
reads & writes

Abort
Retry Abandon
decided by "user"



Transaction logs

How do you undo the effect of an aborted transaction?

Record each update

Where Old value New value → On disk

When to log?

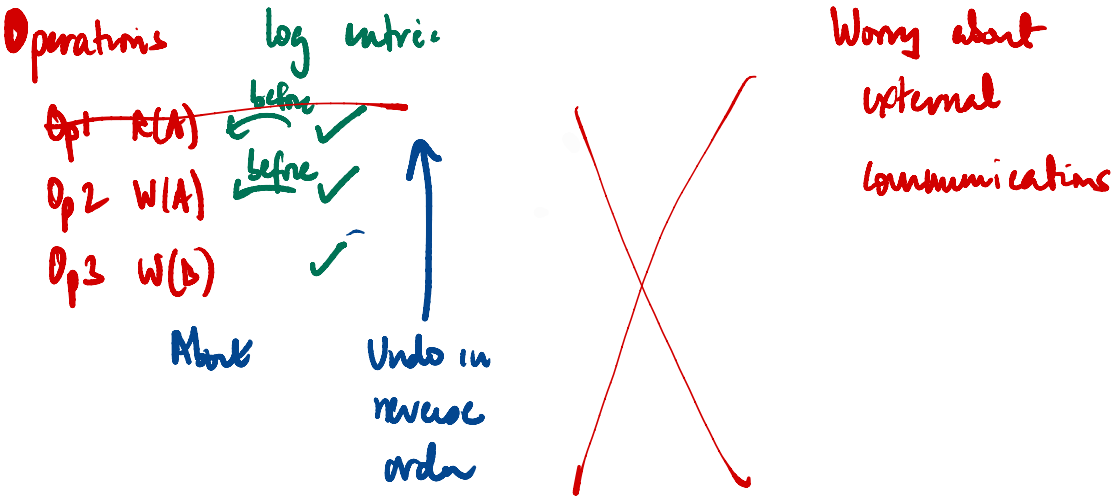
Crash!

① Update database || ② Write log entry

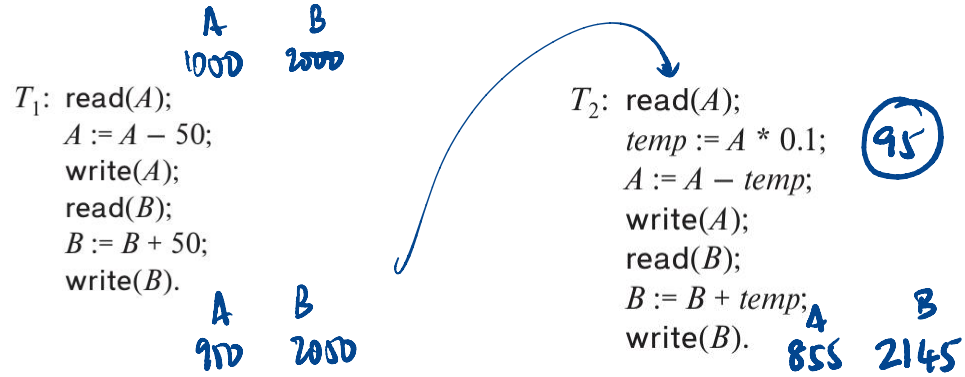
Log before update

log ——— crash
update

Transaction logs



Concurrent execution and schedules



Concurrent execution and schedules

Schedule

Order in which the operations are executed

All of T_1 before all of T_2

T_1	T_2
read(A)	
$A := A - 50$	
write(A)	
read(B)	
$B := B + 50$	
write(B)	
commit	
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
	read(B)
	$B := B + temp$
	write(B)
	commit

1000, 2000

950, 1050

855, 1145

Concurrent execution and schedules

All of T_2 before all of T_1

Two schedules have different outcomes

T_1	T_2
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
	read(B)
	$B := B + temp$
	write(B)
	commit
read(A)	
$A := A - 50$	
write(A)	
read(B)	
$B := B + 50$	
write(B)	
commit	

1000 2000

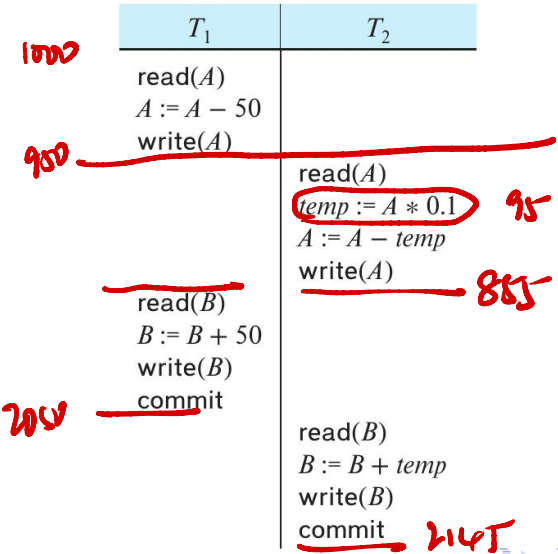
900 2100

850 2150

Concurrent execution and schedules

This concurrent schedules "behaves" like T_1 then T_2

When is a concurrent schedule good?



Concurrent execution and schedules

Like my applicant ID problem

Inconsistent (bad) concurrent schedule