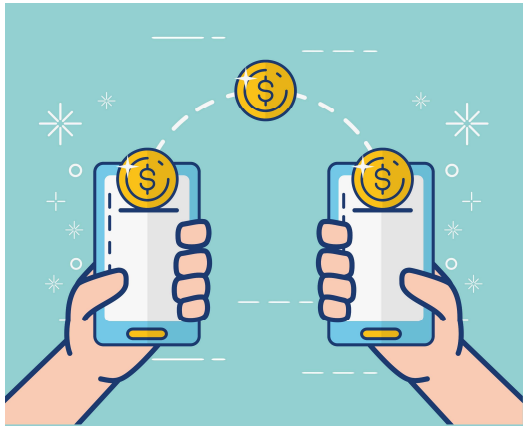


UNIT 4 TRANSACTIONS

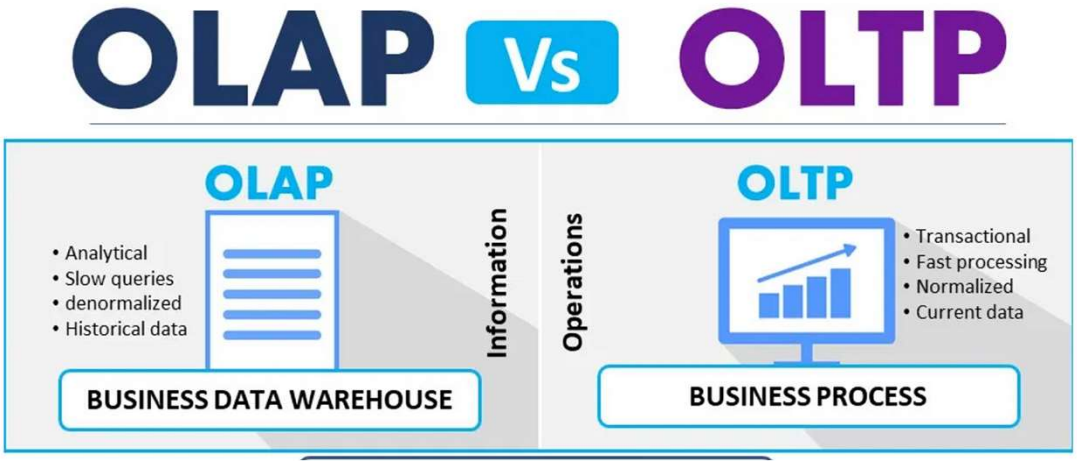
Transaction Concepts –ACID Properties – Schedules – Serializability – Concurrency Control – Need for Concurrency – Locking Protocols – Two Phase Locking – Deadlocks – Transaction Recovery – Save Points – Isolation Levels – SQL Facilities for Concurrency and Recovery

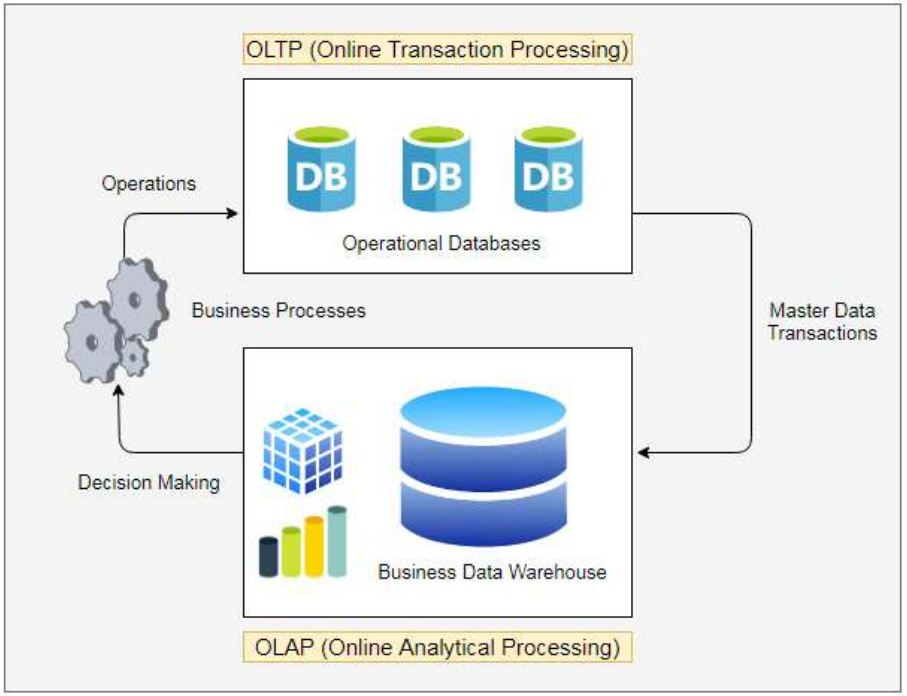
Recap

UNIT I – Introduction , Architecture, ER Diagram

UNIT II – Relational Algebra and Queries – SQL, Intermediate, Advanced, Embedded, Dynamic, etc.,

UNIT III – Dependencies and Normal Form


OLAP Vs OLTP
OLAP
• Analytical
• Slow queries
• denormalized
• Historical data
BUSINESS DATA WAREHOUSE
Information
Operations
OLTP
• Transactional
• Fast processing
• Normalized
• Current data
BUSINESS PROCESS


OLTP (Online Transaction Processing)
DB
DB
DB
Operational Databases
Operations
Business Processes
Master Data
Transactions
Decision Making
Business Data Warehouse
OLAP (Online Analytical Processing)

Dealing with the bank accounts of two users

Moving an amount of 5000 from Karlos to Ray

1. Open_Acc (Karlos)
2. OldBal = Karlos.
3. bal NewBal = OldBal - 5000
4. Ram.bal = NewBal
5. CloseAccount(Karlos)

1. OpenAccount(Ray)
2. Old_Bal = Ray.bal
3. NewBal = Old_Bal + 5000
4. Ahmed.bal = NewBal
5. CloseAccount(Ray)

Transaction Concepts

- Unit of program execution that accesses and possibly updates various data items.
- Example

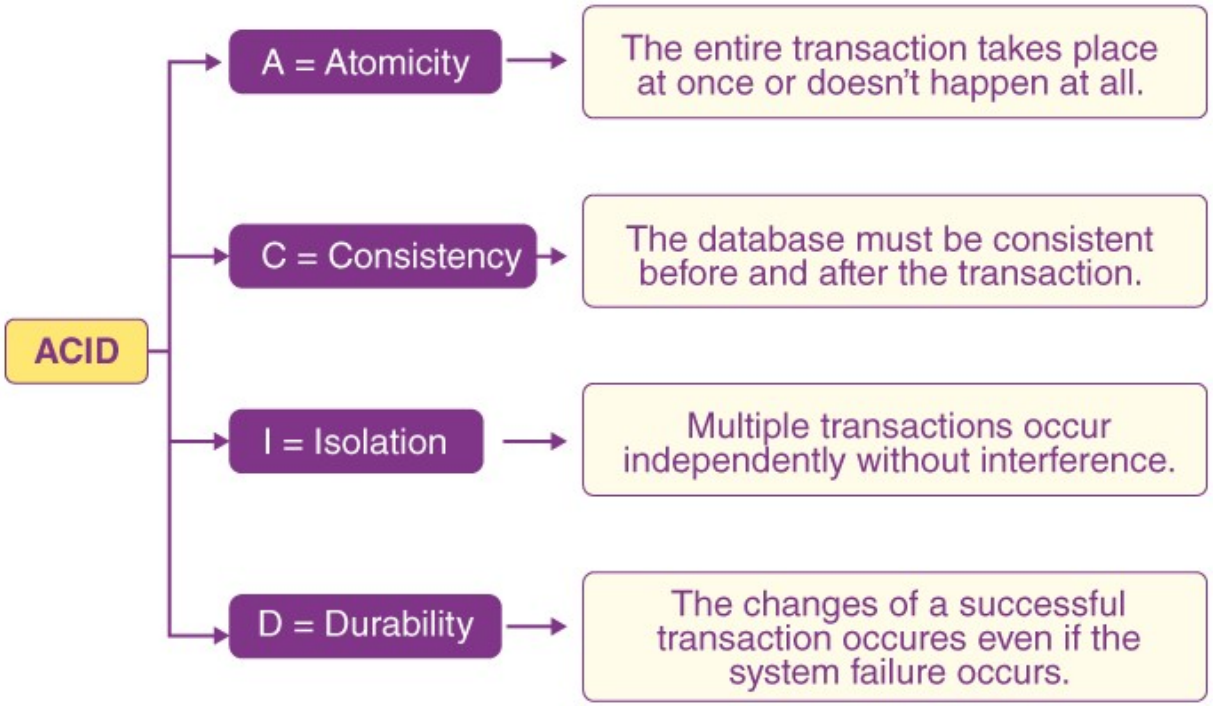
  Transaction to transfer $50 from account A to account B:

  1. **read**(A)
  2. $A := A - 50$
  3. **write**(A)
  4. **read**(B)
  5. $B := B + 50$
  6. **write**(B)

Transaction Concepts

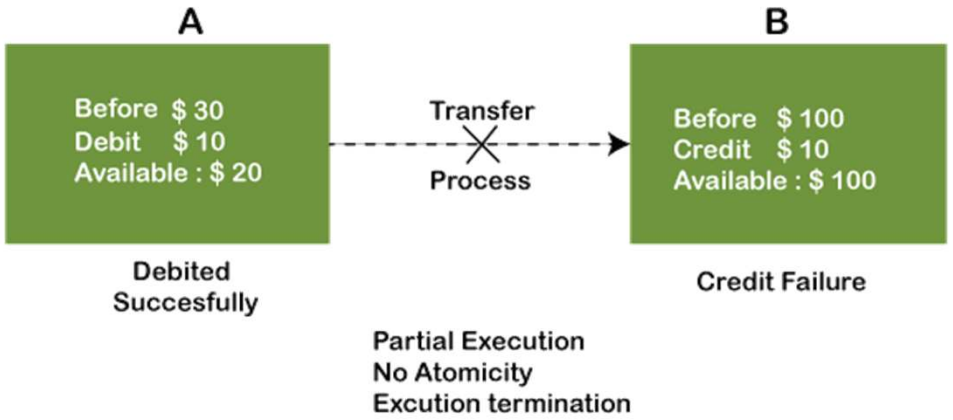
- Two main issues to deal with:
  - Failures of various kinds, such as hardware failures and system crashes
  - Concurrent execution of multiple transactions

ACID Properties in DBMS



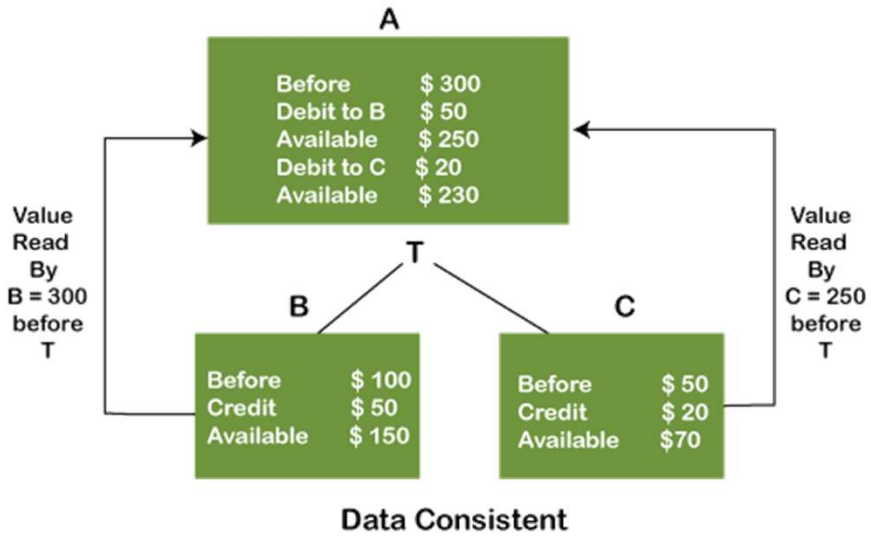
ACID Properties

- **Atomicity -** Either all operations of the transaction are properly reflected in the database or none are**. (ALL / NONE)**



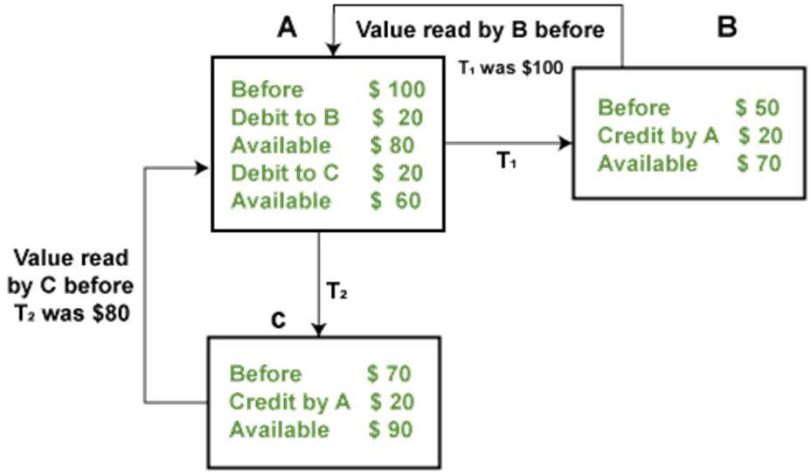
ACID Properties

- **Consistency -** Execution of a transaction in isolation preserves the consistency of the database.



ACID Properties

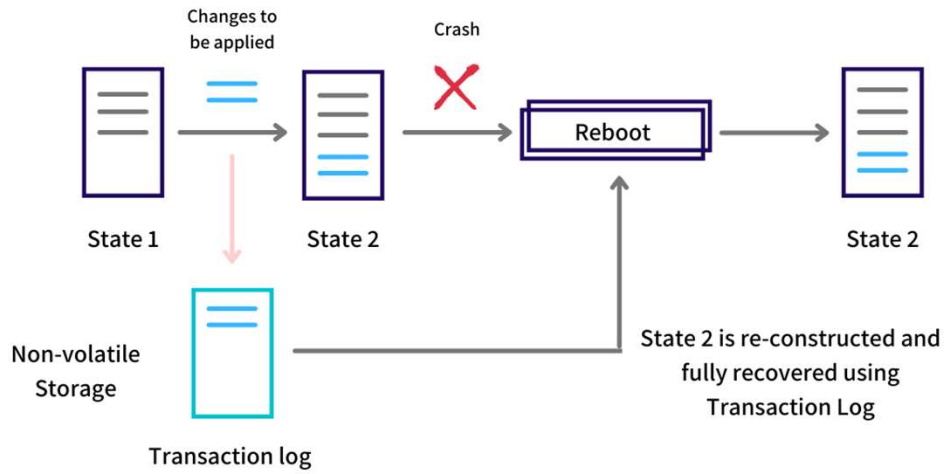
- **Isolation -** Multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions.



Isolation - Independent execution of T_1 & T_2 by A

ACID Properties

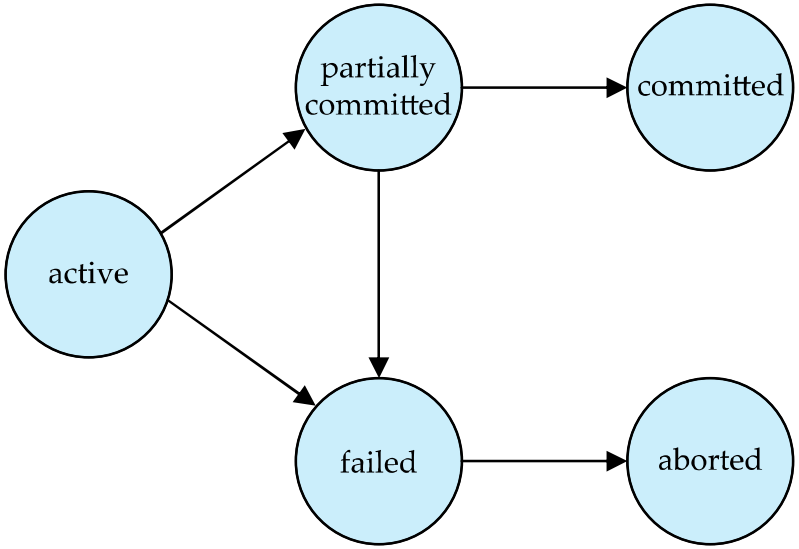
- **Durability -** After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

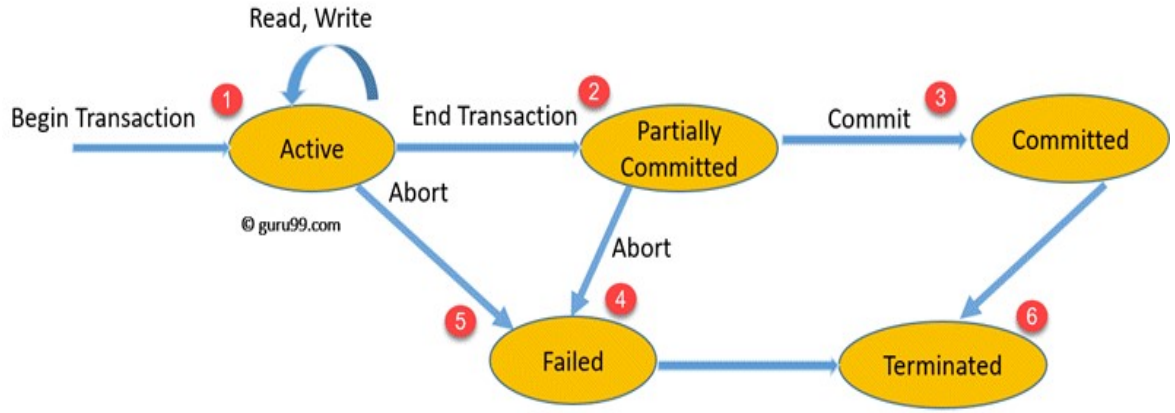


Durability

Icebreaker Activity

Transaction State





Concurrent Executions

Multiple transactions are allowed to run concurrently in the system.

Advantages

- **Increased processor and disk utilization –** Better throughput
- **Reduced average response**

Schedules

A sequences of instructions that specify the chronological order in which instructions of concurrent transactions are executed

- Successfully completes – **Execute Commit**
- Fails to Complete – **Execute Abort**

Schedule 1

- T_1 transfer \$50 from A to B, and
- T_2 transfer 10% of the balance from A to B.
- A serial schedule - T_1 is followed by 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

Schedule 2

A serial schedule where T_2 is followed by T_1

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	

Schedule 3

- T_1 and T_2 be the transactions defined previously not a serial schedule, but it is *equivalent* to Schedule 1

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

Schedule 4

- concurrent schedule does not preserve the value of $(A + B)$

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

Assessment

1. What is a transaction in DBMS?

A transaction refers to a set of operations that perform a single set of logical work.

2. What are ACID properties in DBMS?

ACID stands for Atomicity, Consistency, Isolation, and Durability.

3. Why are ACID properties important?

ACID properties help to keep the data within databases correct and consistent, and preserve it in case of unexpected events like power failures.

Summary

- Transaction
- ACID Properties
- Schedules

Upcoming Session - Serializability

- How different process operates the shared data?
- Each transaction preserves database consistency.
- Serial execution of a set of transactions preserves database consistency.
- Different forms of schedule equivalence give rise to the notions of:
  1. **Conflict Serializability**
  2. **View Serializability**


Thank you