

Q12. Groupware and its types

- Distributed system is a collection of autonomous computer systems that are physically separated, but are connected by a centralized computer network that is equipped with distributed system software.
- One of the advantages of a distributed system is information sharing.
A distributed system can be used efficiently when information is generated by one user and is shared by the users working at other nodes of the system.
- The use of distributed computing systems by ^{a group of} users to work cooperatively is known as computer-supported cooperative working (CSCW) or groupware.
- Groupware applications are primarily based on the sharing of data objects between programs running on different nodes of a distributed computer system.
- Groupware is an emerging technology that holds major promise for software developers.
- People work on a common project but are separated geographically. They share the interface for communication and this is done with the help of groupware.

- Alternative name for groupware is collaboration software.

Classification →

Groupware can be classified based on when and where the participants are collaborating, and by functionalities they need for their work.

1. By time-space matrix :

"when" classification : when the participants are working, at the same time or not.

Common axes for time based classification :

- synchronous
- asynchronous

synchronous groupware allow individuals ~~to~~ located at different locations ~~to~~ in real time. Coordination and sharing the resources is a main feature.

eg. a telephonic conversation where they share audio channels.

If the participants are co located, it will be like a face-to-face conversation in real time.

asynchronous collaboration happens when there is a replica of the data system and with reading and writing data access.

eg. email, file sharing systems etc.

Common axes for place classification are:

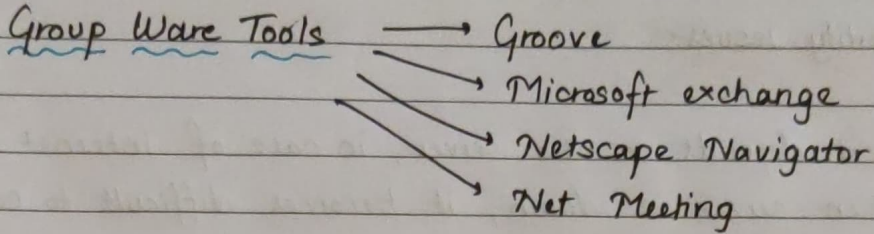
- Co-located
- Remotely located.

2. By functionality:

Cooperative work involves \rightarrow Participators who are work
 \rightarrow Artifacts which they work.

- There will be direct communication between participators, mediated by computer systems.
- Examples of computer media include emails, bulletin boards, text messages etc.
- A common understanding is captured by meeting and decision support systems.
- This may have a meeting room, shared drawing interface etc.
- Applications and artifacts of the work are shared among them
- Control and feedback can be done by ~~the~~ participants with the shared objects.

Group Ware Tools



Advantages →

- Enhances user creativity -

Different groups of people come up with ideas to foster creativity and enhance the project that they collaborate on

- Facilitates communication -

Users can discuss and come to joint conclusions via messages and chat.

- Visualize vision and goals -

Enables users to come up with a goal.

- Any time any place ~~collaboration~~ interaction

- Open network and open client standards make it very efficient to use.

Disadvantages →

- Cost factor -

The purchase and maintenance of groupware is a challenging task.

- Reliability issues-

Since it depends on the server, in case of internet issue or when server is down, it becomes difficult to carry on the work.

- Dependence on a single vendor

- Network security with slow speed, reliability and access problems is a major disadvantage.

Unit - 5

classmate

Date _____

Page _____

Q8. Transaction Processing Systems

What is a transaction in distributed DBMS?

- A program that includes a collection of database operations which are executed as a logical unit of processing data is called a transaction.
- In a transaction, one or more data operations such as insert, update, delete are performed
- All this process is automated, and if it is performed, it is performed to completion or is not performed at all.
- Transactions that do not involve any updating of data, but only involve retrieving data are called read-only transactions.
- A high level operation is divided into many low level operations, for example the update operation is divided into the following three lower level operations →
 - read-item()
 - modify-item()
 - write-item()
- The access of the database is restricted to only the read-item() and write-item() operations.
- Transaction operations →

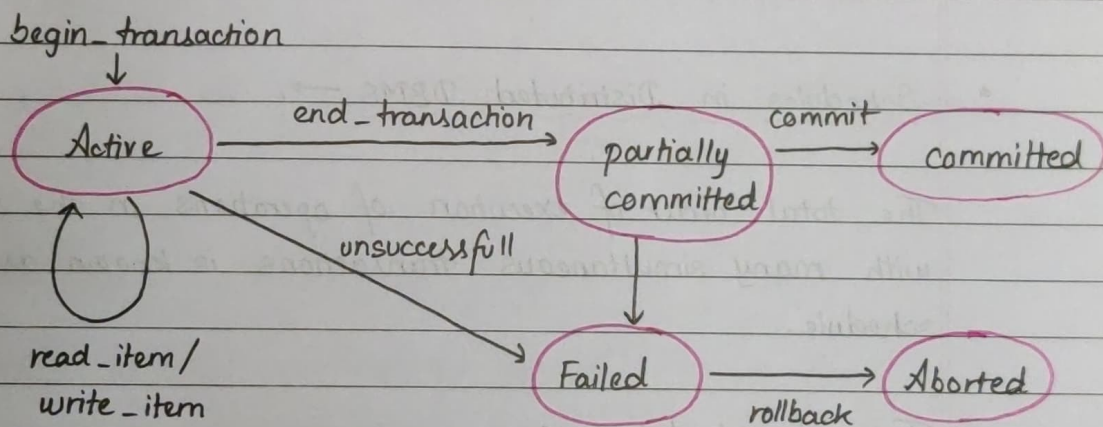
The operations that can be performed on a transaction at low level are:

- begin_transaction: The start of transaction execution is specified by this marker.
- read_item / write_item:
- end_transaction: The end of the transaction is specified by this marker.
- commit: The successful completion of the transaction in its entirety is specified by it.
- rollback: The failure of the transaction and the status that the temporary changes have been undone are specified by it.
Once a transaction is committed, it cannot be rolled back.

• Different States of a Transaction →

- There are a set of 5 states that a transaction must go through.
They are as follows →
- Active - The transaction is in an active state and operations like read, write etc. are performed by the transaction by remaining in this state.
- Partially Committed → Once the last statement of the transaction is completed, the transaction enters this state.
- Committed → After the transaction is successfully completed, and the commit signal is issued, the transaction enters this state.

- Failed → When the transaction cannot proceed with the normal execution, the transaction enters this state either from active or partially committed state.
- Aborted → The state when the transaction has been completely rolled back after restoring the failure database.



- What are the desirable properties of Transactions?

ACID properties need to be maintained by transactions -

- Atomicity: The transaction is either performed completely or is not performed at all.
There exists no partial update of the transaction.
- Consistency: The database needs to be taken from one consistent state to another by a transaction without affecting the data item in the database.

→ Isolation: There should not be any ^{interference} interface to a transaction from any other transactions running simultaneously.

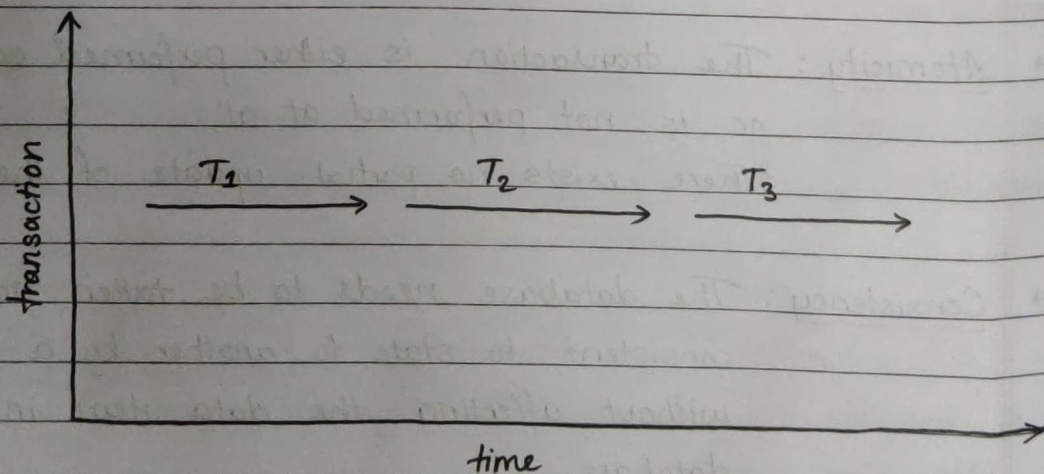
→ Durability: The change brought by the committed transaction should be durable in the database and should not go away or get lost in case of failure transactions.

• Schedules in Distributed DBMS →

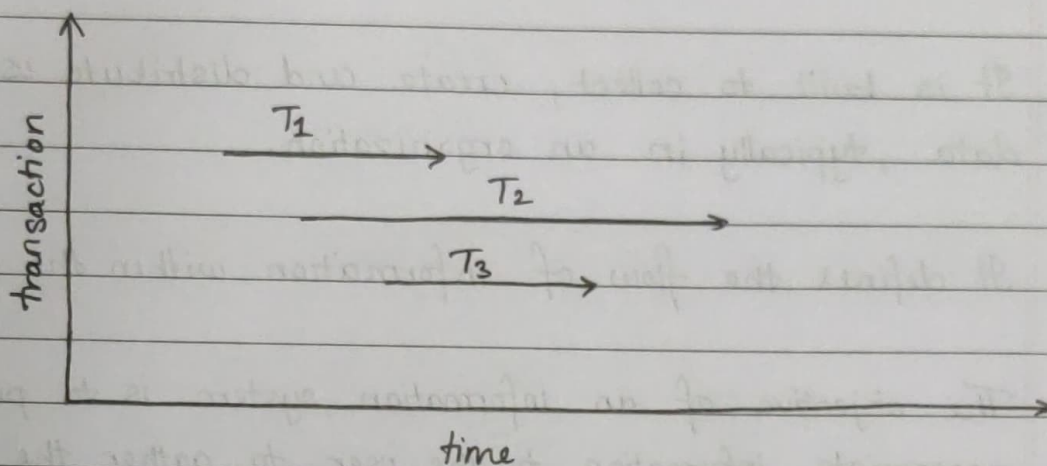
The total order of execution of operations in the system with many simultaneous transactions is known as a schedule.

Types of schedules →

→ Serial schedules - At any given time, if only one transaction is active, it is known as serial schedule. No transactions overlap in serial schedule.



→ Parallel schedules - When there are more than one active transactions at any given point of time it is called parallel schedule. Transactions overlap in parallel schedule.



- Conflicts in Schedules →

- When multiple ^{transactions} are included in a schedule, when non-compatible operations are performed by two active transactions this leads to occurrence of a conflict.

- The operation can be proved to be in conflict only on the existence of the following three conditions simultaneously →

- When ~~to~~ two operations belong to different transactions

- The same data item is accessed by both the operations.

- There should be atleast one write-item() operation, which modifies the data item.