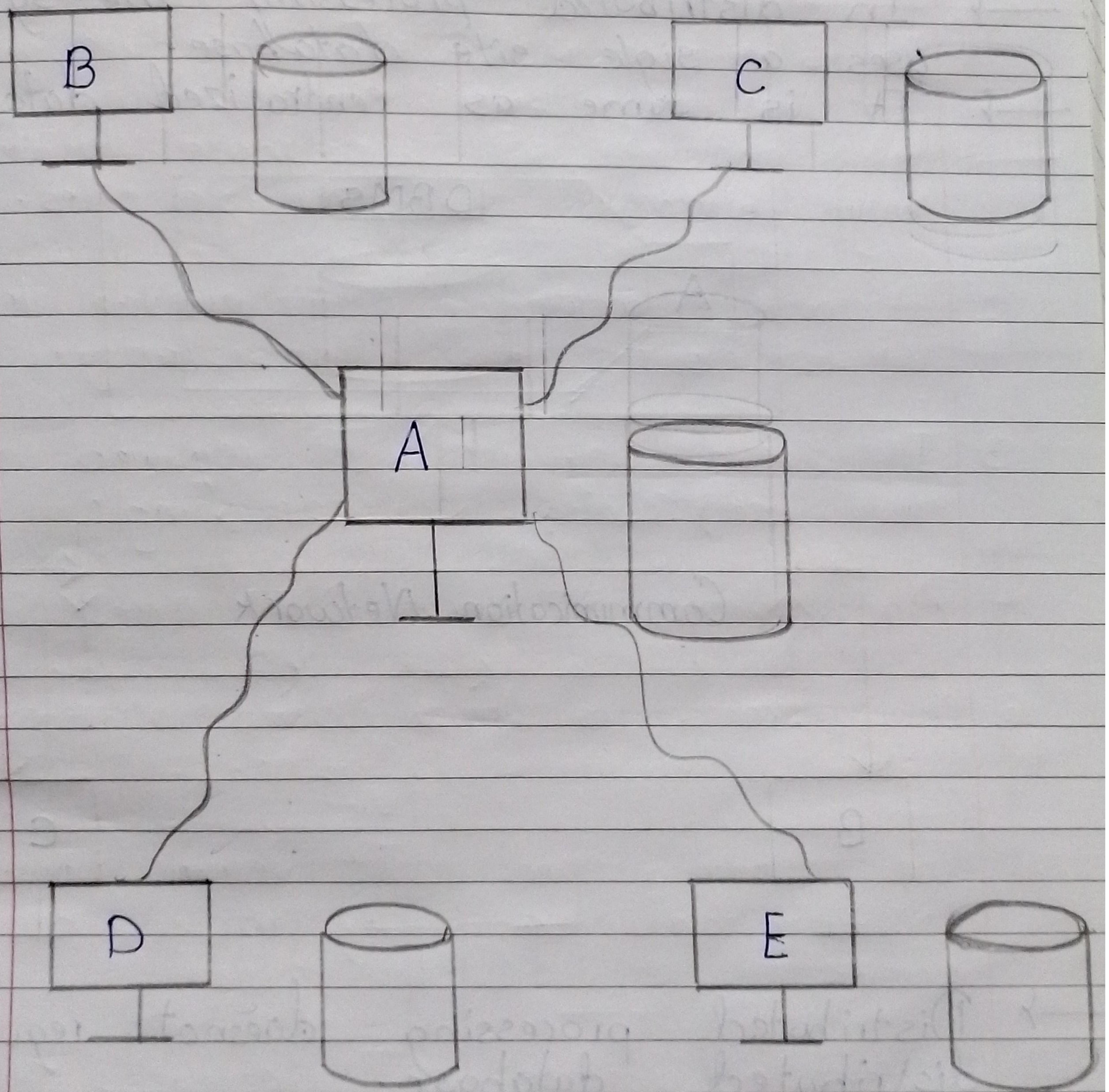


Unit - 3 Distributed Database Management System

10/3

PAGE NO.:

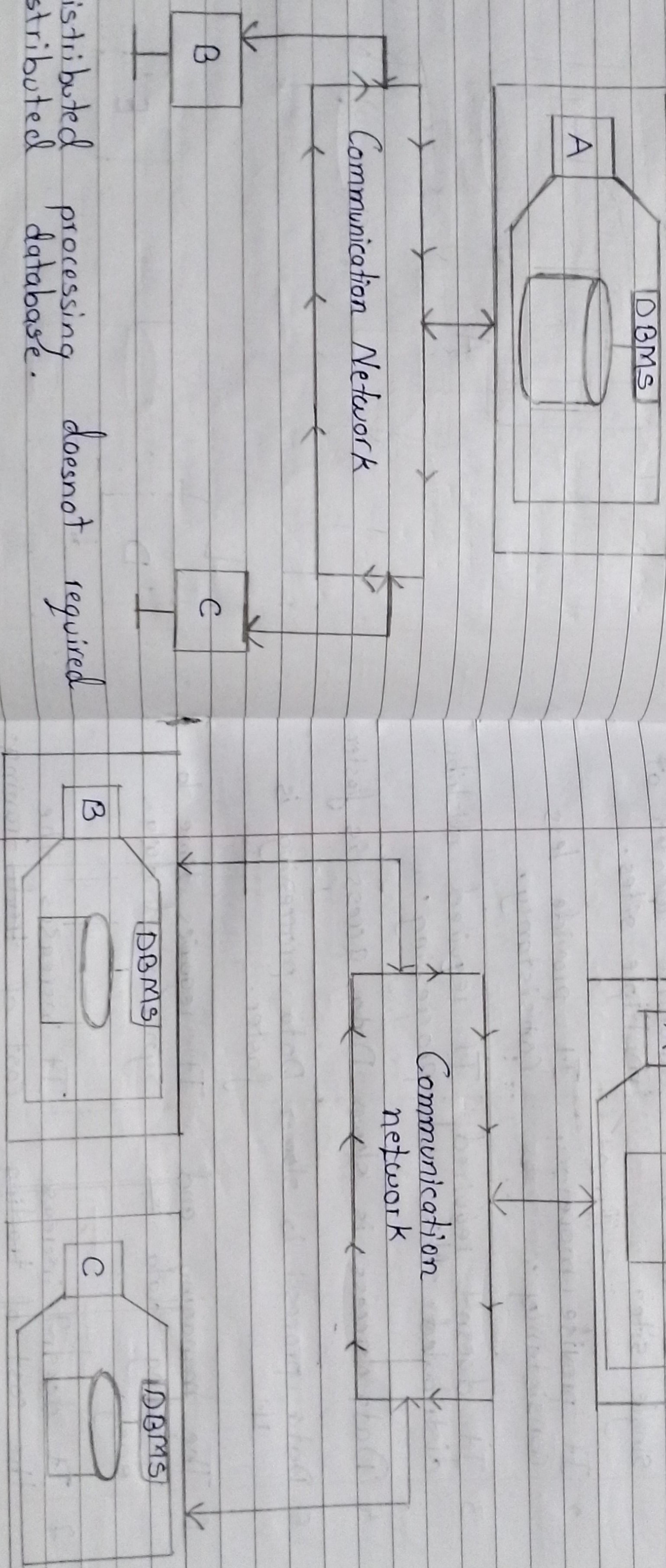
- D-DBMS Stores a logically related data over interconnected computer system In which data & processing are distributed among several sites.
- When multiple copies of single database is stored at different sites then it is called D-DBMS.



* Distributed processing & Distributed database

Distributed database.

- 1) Distributed processing
 - In distributed processing in database logical processing is shared among 2 or more physically independent sites.
 - In distributed processing through a network.
 - It uses a single site database.
 - It is same as centralized database.



→ Distributed processing doesn't required distributed database.

→ Distributed database required distribute processing.

► Difference between Distributed database & Distributed processing

- | Distributed Processing | Distributed Database |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 1 It is a type of database that contains a single database located at single site. | It is a type of database that contains copies of database located at multiple sites. |
| 2 It provide more consistancy. | It provide less consistency. |
| 3 It doesn't required processing. | It required distributed processing. |
| 4 Data access is slower. | Data access is faster. |
| 5 Data processi is slower | Data processing is faster. |
| 6 The managing and updating data is easy. | It requires time to synchronize data. |
| 7 It doesn't increase the cost of training. | It increases the cost of training. |

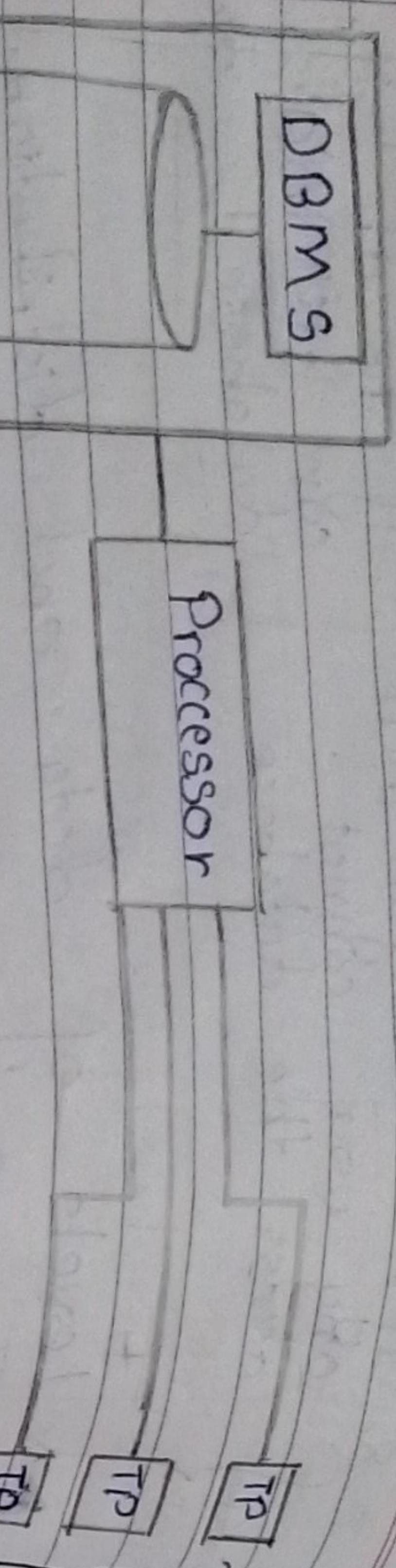
► Levels of data and distribution.

- | There are 3 levels of data & process distribution. |
|---------------------------------------------------------|
| 1 Single Site processing , single site data (SSP SD) |
| 2 Multiple site processing , single site data (MPP SD) |
| 3 Multiple site processing , multiple site data (MPNND) |
- ↳ Single site processing , single site data (SSP SD)
- In a single site processing single site data scenario all process is done on a single host computer and all data are stored on the host computer local disk.
- Processing cannot be done on the end user site system.
- All data storage and data processing are handled by a single host computer.

8 If the db fail then any user cannot access the database.

If the db fail, user can still access the database.

3) Multi site processing , multi site data (MPPD)



Dumb Terminal

2) Multiple site processing and Single Site data (MSSD)

- In the multiple site processing scenario describes a fully distributed dbms with support for multiple data processor and transaction processor at multiple site.
- The D-DBMS in this level is classified as either homogeneous or heterogeneous.

The multiple site processing multiple site data scenario describes a fully distributed dbms with support for multiple data processor and transaction processor at multiple site.

The Homogeneous D-DBMS integrate multiple types of the same dbms over a network.

Heterogeneous D-DBMS integrate different types of dbms over a different network.

→ It is same as client server architecture where multiple clients request to the server for the data processing.

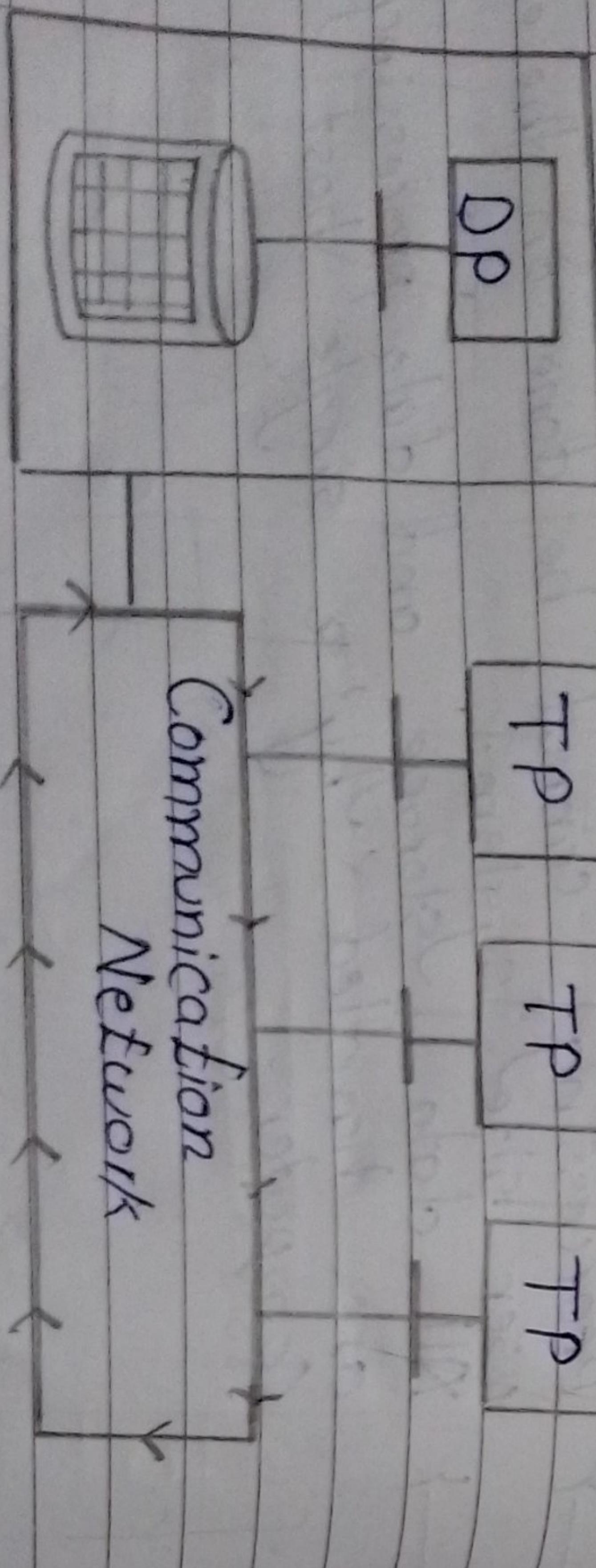
File Server

A B C

TP TP TP

DP

Communication Network



*

Distributed database transparency features

- The distributed transparency features are
 - It allows a distributed database to be treated as a single logical database with distributed transparency. The user doesn't need to know that
 - that the data are replicated among multiple sites.
 - that the data are geographically distributed.
- Transaction transparency
- It allows a transaction to update data at more than one network sites. It ensures that the transaction will be either completed or aborted which maintain data integrity.
- Failure Transparency.
- It ensures that the system will continue to operate in the event of a node or network failure.
- Performance

Performance Transparency

It allows the system to perform as if it was a centralized dbms. The system will not suffer any performance degradation due to its use on a network. It ensure that the system will find the most effective path to access remote data.

Heterogeneity transparency

It allows the integration of several local dbms under a common global schema.

Distributed transparency

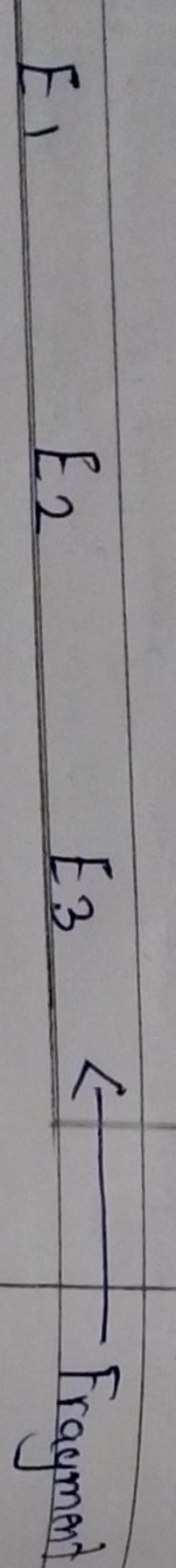
- It allows physically distributed database to be managed as a centralized database.
- There are 3 levels of distribution transparency.

(1) Fragmentation transparency

- It is the highest level of transparency.
- The end user or programmer doesn't need to know that a database is partitioned.
- In this type of transparency the fragment name & location are not specified.

DBMS

EMPLOYEE TABLE



Newyork Atlanta Miami ← Location

DBMS

Select * from E1 where age > 30;

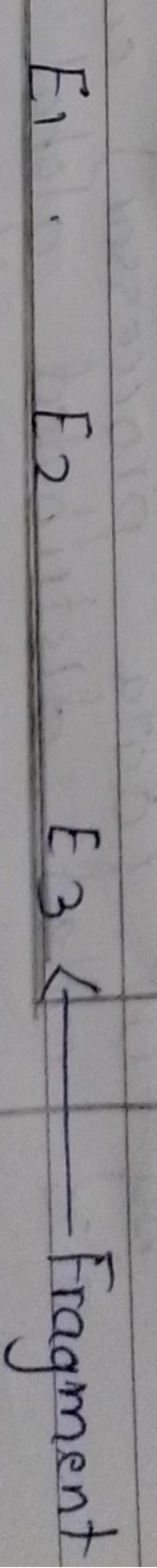
(2) Local

- The local mapping transparency exist when end user or programmer must specify both the fragment name & their location.

Newyork Atlanta Miami ← Location

DBMS

EMPLOYEE TABLE



Newyork Atlanta Miami ← Location

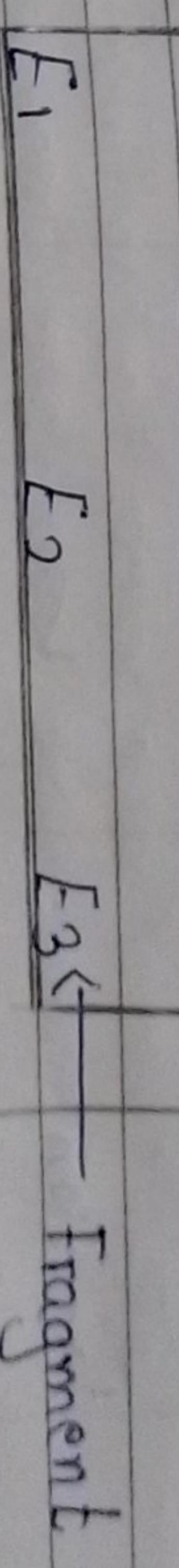
Level 2 Select * from Employee where age > 30;

Location

- The location transparency exist when end user or programmer must specify fragment name but doesn't specify where fragment is stored.

DBMS

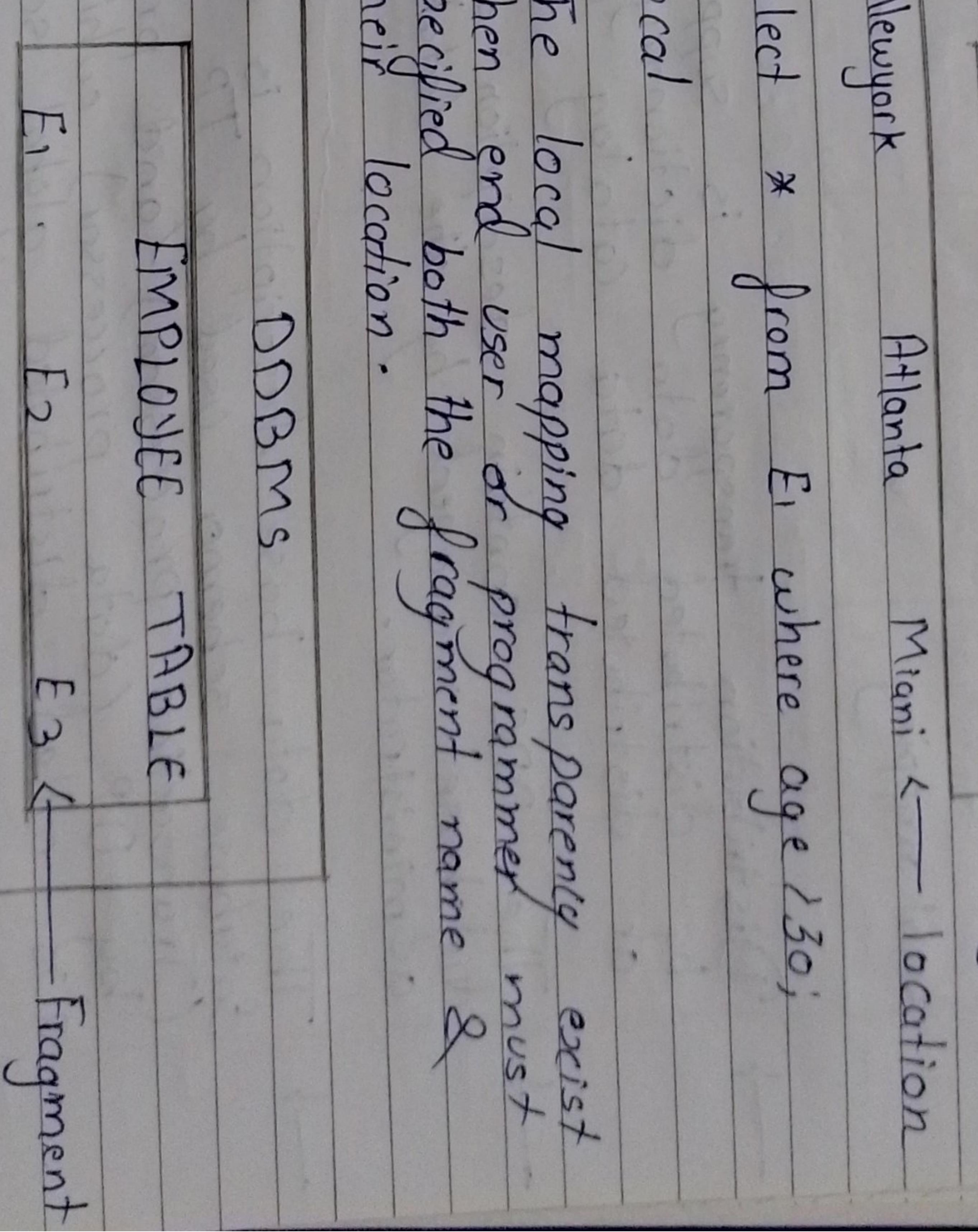
EMPLOYEE TABLE



Newyork Atlanta Miami ← Location

DBMS

EMPLOYEE TABLE



Newyork Atlanta Miami ← Location

DBMS

EMPLOYEE TABLE

Select * from E1 NODE Newyork

where age > 30;

→ The distribution transparency levels can be summarized as

Levels Fragment name Location name

Fragmentation X X

Location ✓ X

Local mapping ✓ ✓

→ Distribution transparency is supported by distributed data dictionary (DDC)

or distributed data catalog (DDC)

→ The DDC contain description of entire database managed by the database administrator.

→ The data base description is a common database schema used by TP (transaction processor) and processed by DP (data processor) which is also called distributed global schema.

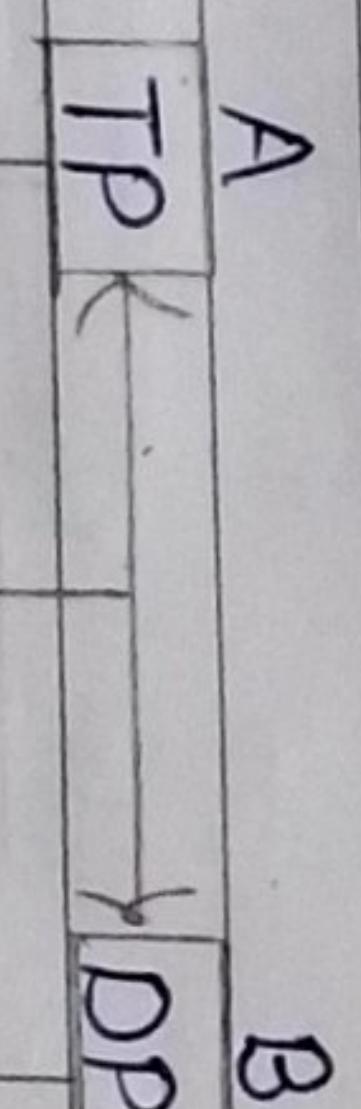
Transaction transparency

→ The transaction transparency ensure that the transaction will be completed only when all database sites involved in the transaction complete phase.

1 Distributed request & Distributed transaction

→ A distributed request allows single SQL statement reference data located at several different local AP sites.

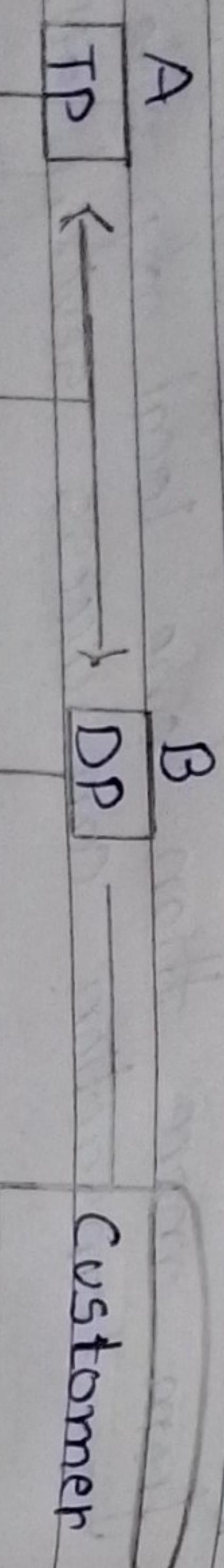
→ Because each request can access data from more than one local AP site, a transaction can access several sites.



→ Select * from customer, invoice where customer.cno = invoice.cno;

Distributed Concurrency Control

- In the given example transaction using a single statement to reference to using a table which are located at different sites.
- * Distributed Transaction
 - A distributed transaction reference different local site.
 - It single request can reference on one dp site, the transaction as a whole can reference multiple dp site because each request can reference a different site.



Such a scenario can result in inconsistent data database because the committed data cannot be uncommitted.

DP	Lock(x)		
Site A	write(x)	Commit	
			Cannot
			rollback

Site A & B

DP	Lock(y)		
Site B	write(y)	Commit	
			Lock(x)
			rollback

→ Update product set qty = qty - 1
where pid = 1;

→ Update customer set balance + 100
where Cno = 1;

Site C	Lock(x)		Rollback
	- - -		at Site C.

→ The solution to this problem is two phase commit protocol.

3 Two phase Commit

→ The two phase commit protocol (2PC)

guarantee set if a portion of a transaction cannot be committed then all the changes will be undone to maintain

a consistent database site state.

The 2 phase commit protocol require

that the transaction log entry for each dp will be return before the database is actually updated.

→ It requires DO-UNDO-REDO protocol and write-ahead protocol.

[A] Do-UNDO-REDO

→ It is used by a dp to rollback the transaction.

→ It defines 3 types of operations

DO: Performs the transaction (operation)

UNDO: return recursive, the operation.

REDO: redost the operation.

[B] Write-ahead

→ It forces the log entry to be written to permanent storage before an actual operation take place.

→ The two phase commit protocol defines the operation between two type of nodes Coordination and Subordinates

→ The protocol is implemented in two phases

Phase 1 : Preparation

① The coordinator sends PREPARE to commit message to order sub ordinate.

② The sub coordinate receive the message write the transaction log and send the acknowledge message (Yes / PREPARE or NO / NOTPREPARED) to the coordinator.

The coordinator make sure the all the nodes are ready to commit, otherwise abort the action.

If all nodes are PREPARE to commit then the transaction goes to phase - 2

Phase 2 : The final Commit

① The coordinator broadcast COMMIT message to all the sub ordinates and wait for the reply .

② Each subordinates receive the COMMIT message and update the database. The subordinates reply with COMMITTED

or Not COMMITTED Message to the Coordinator.

If one or more Subordinates do not commit, the coordinator sends abort message to UNDO all the changes

(1) Data distribution

In dbms the every transaction is more complicated because dbms must decide which fragment access.

Performance & Failure Transparency

The performance transparency allow DBMS to perform vs if it was:
The objective of query optimization is to minimize the total cost associated with the execution of a request.

(2) Data replication
→ The data may be replicated at several different site.
→ The database must ensure that all copies of the data are consistency

(3) Network and node availability

→ The response time associated with remote site cannot be predetermined based on the network traffic.

* Access time (TO) cost involve in accessing the data from multiple remote sites.

* Communication cost associated with data transmission among notes in dbms.
→ Associated with the processing overhead of managing the distributed transaction.

→ The resolving data request is distributed data environment must consider following points.

Questions

- 1 What is ddbms ? Explain distributed processing and distributed database.
- 2 List and explain levels of data and process distribution.
(or)
Explain SDSN, MPSD, MPMD
- 3 List and explain levels distributed database transparency features.
- 4 List and explain levels of distributed transparency.
- 5 Explain two phase commit protocol.
- 6 Explain the problem with distributed concurrency control.
- 7 Differentiate distributed request and distributed transaction.
- 8 Explain factors associate with performance transparency.