QUANTITATIVE VS QUALITATIVE INFORMATION REGARDING FRAGMENTATION

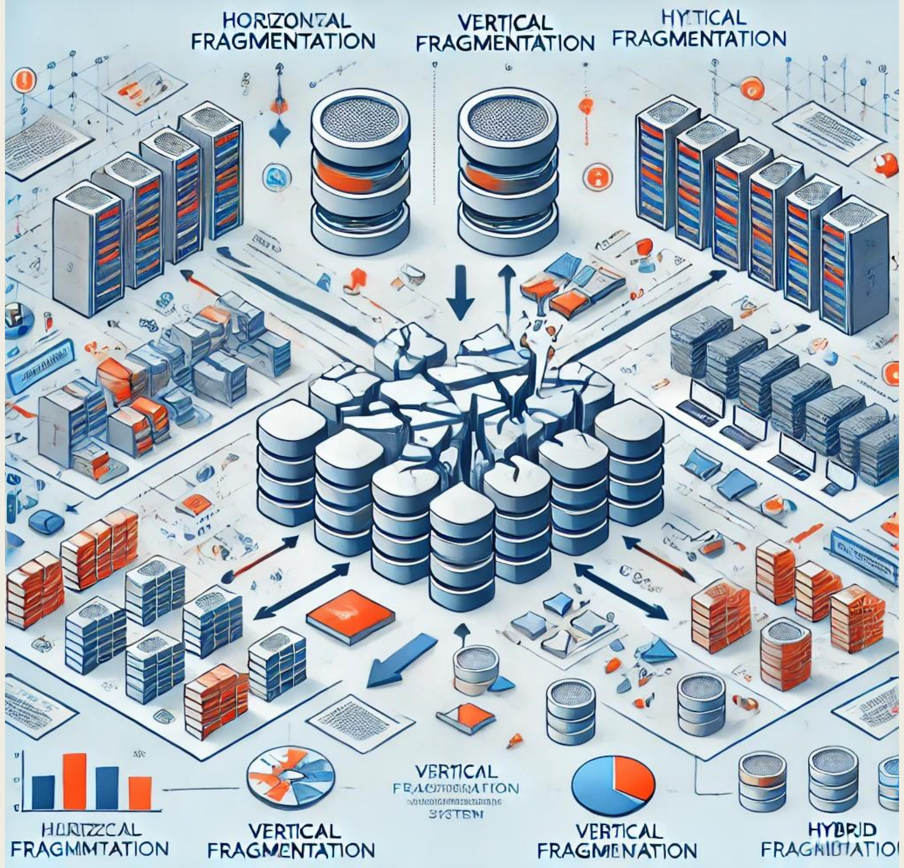
NAME – ABHIRUP BAG
ROLL – 13000122082
DEPARTMENT – CSE(B)
SEMESTER – 6
PAPER – DISTRIBUTED DATABASE MANAGEMENT SYSTEM(PEC-IT 601B)

CONTENT

❑ Introduction	3
❑ Quantitative vs Qualitative Information	4
❑ Quantitative information	5 - 7
❑ Qualitative information	8 - 10
❑ Comparison with Parameters	11
❑ Practical Application	12
❑ Example scenario	13-15
❑ Conclusion	16
❑ References	17

INTRODUCTION

- **What is Fragmentation in Distributed DBMS?**
  - ✓ Division of a database into smaller parts (fragments) stored across multiple nodes.
- **Why is Fragmentation Important?**
  - ✓ Improves performance, scalability, and resource utilization.
- **Objective of the Presentation:**
  - ✓ Understand the role of quantitative and qualitative information in fragmentation.
  - ✓ Explore key parameters: minterm selectivity, access frequency, completeness, and minimality.



QUANTITATIVE VS QUALITATIVE INFORMATION

1 Quantitative Information:

- Numerical, measurable data.
- Focus: Performance metrics, statistics.
- Examples: Minterm selectivity, access frequency.

2 Qualitative Information:

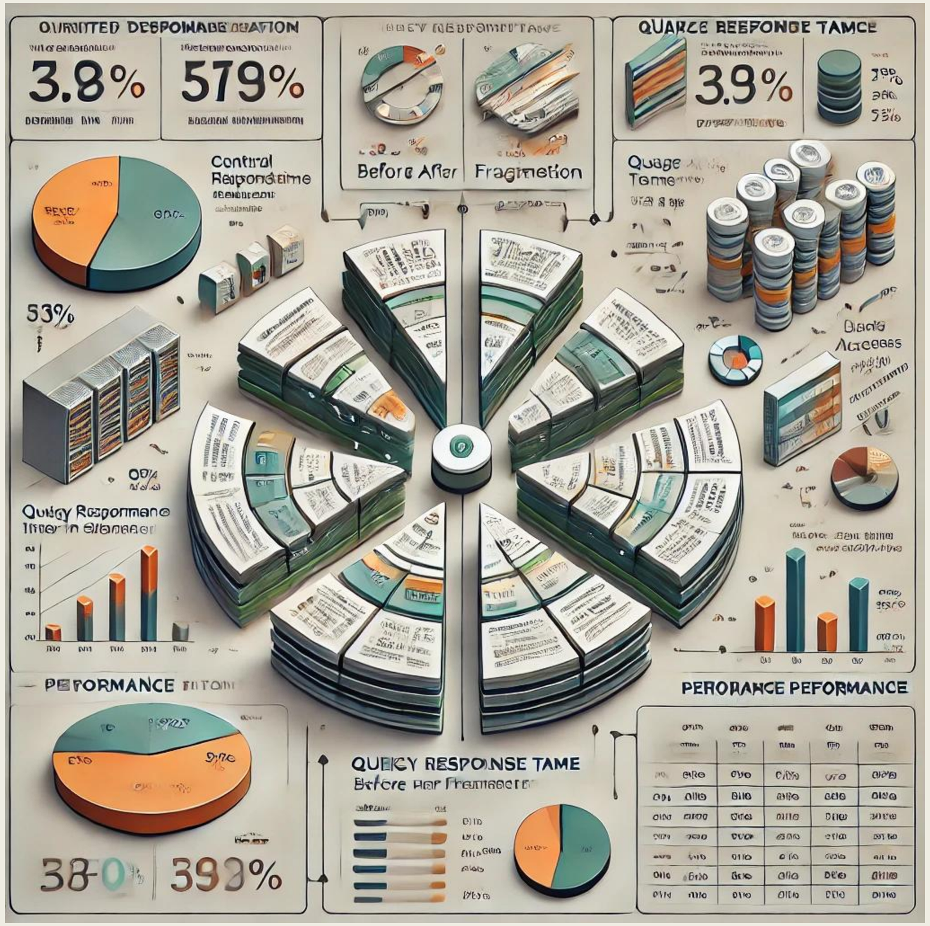
- Descriptive, conceptual data.
- Focus: Design principles, trade-offs.
- Examples: Completeness, minimality.

QUANTITATIVE INFORMATION

Quantitative analysis of fragmentation in a **Distributed DBMS** helps evaluate efficiency and performance. Key metrics include:

- **Fragmentation Granularity (FG):** Measures fragment size in terms of rows or columns.
- **Data Transmission Cost (DTC):** Cost of transferring fragments across network nodes.
- **Access Frequency (AF):** Number of times a fragment is accessed, influencing placement decisions.
- **Response Time (RT):** Time taken to execute queries involving fragmented data.
- **Redundancy Ratio (RR):** Measures data duplication in replicated fragments.
- **Fragmentation Completeness (FC):** Ensures all necessary data is included in fragments.
- **Fragmentation Disjointness (FD):** Ensures minimal data duplication (except for replication).

Efficient fragmentation minimizes **data transfer costs, improves query performance, and enhances system scalability**.



▪ MINTERM SELECTIVITY:

- **Definition:** Minterm selectivity refers to the probability of a record satisfying a given minterm predicate in a horizontal fragmentation scheme.
- **Measurement:** Represented as a fraction (0 to 1) or a percentage (0% to 100%).
- **Usage:** Helps in optimizing horizontal fragmentation by ensuring balanced partitioning based on frequently used predicates.
- **Example:** If a minterm predicate is Region = 'North', selectivity quantifies how many rows match this condition.

ACCESS FREQUENCY:

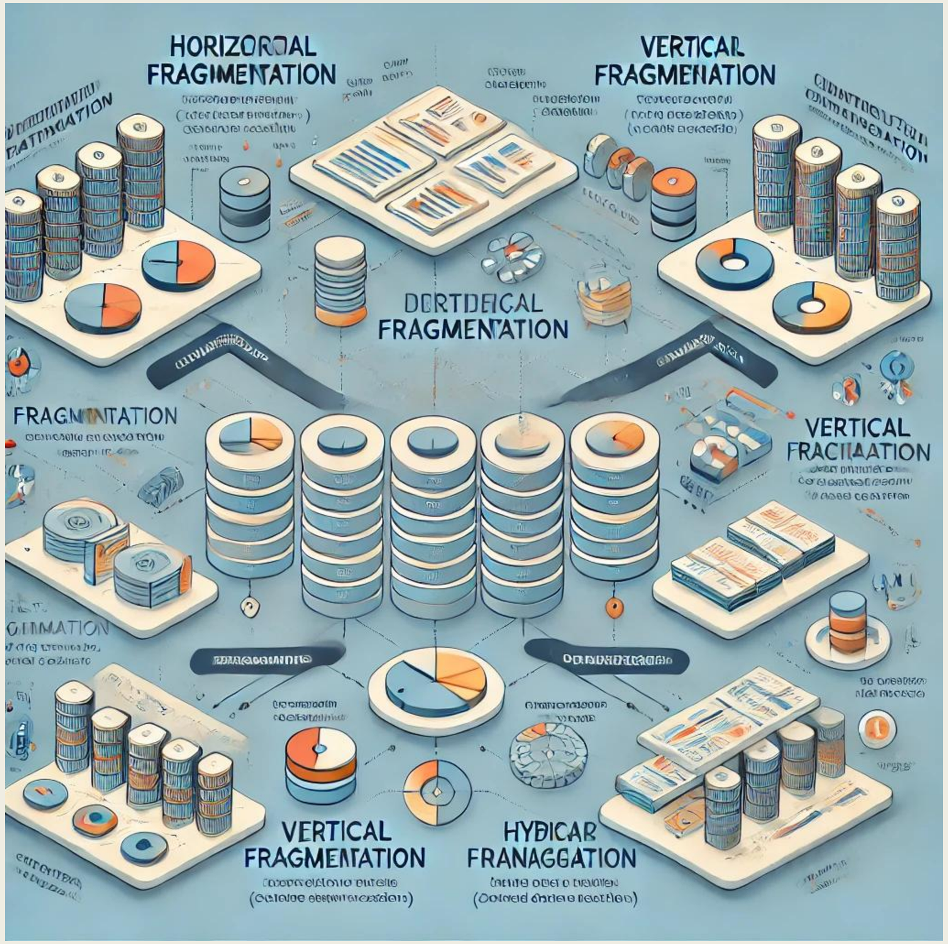
- **Definition:** The number of times a fragment is accessed over a given period.
- **Measurement:** Expressed as requests per second (RPS), queries per minute (QPM), or other time-based metrics.
- **Usage:** Determines optimal fragment placement to minimize network latency and data transfer costs.
- **Example:** A fragment containing customer data for a specific region may be accessed 1,000 times per day.

QUALITATIVE INFORMATION

Qualitative aspects of fragmentation in a **Distributed DBMS** focus on the logical and structural effectiveness of data distribution. Key considerations include:

- **Data Locality:** Ensures frequently accessed data is placed close to users to improve performance.
- **Scalability:** Supports system growth by enabling efficient distribution of data across multiple sites.
- **Reliability & Availability:** Proper fragmentation enhances fault tolerance and ensures data availability.
- **Consistency:** Maintains data integrity by minimizing redundancy and synchronization issues.
- **Query Optimization:** Well-designed fragmentation reduces query processing time by eliminating irrelevant data retrieval.

A well-planned fragmentation strategy **enhances system efficiency, reliability, and user experience** in distributed databases.



▪ COMPLETENESS PARAMETER

- **Definition:** Ensures that when a relation is fragmented, no data is lost—i.e., the union of all fragments reconstructs the original relation.

- **Qualitative value:** A design principle that ensures no data loss during fragmentation.

- **Usage:**
  - ✓ Guarantees that fragmentation does not exclude any tuples.
  - ✓ Affects query correctness and data integrity.

- **Example:** A fragmentation strategy must ensure that all customer records are preserved across fragments, even if they are distributed geographically.

▪ MINIMALITY PARAMETER

- **Definition:** Ensures that no redundant data exists across fragments unless explicitly required (e.g., replication).
- **Usage:**
  - Prevents unnecessary data duplication across sites.
  - Helps in storage efficiency and reducing synchronization costs.
- **Qualitative value:** A design principle that avoids over-fragmentation or redundancy.
- **Example:** A fragmentation strategy should avoid creating too many small fragments, as this increases management overhead.

COMPARISON WITH PARAMETERS

Parameter	Quantitative Information	Qualitative Information
Minterm Selectivity	Proportion of tuples satisfying a predicate (e.g., 30%).	N/A (Quantitative only).
Access Frequency	Number of accesses per fragment (e.g., 1,000/day).	N/A (Quantitative only).
Completeness	N/A (Qualitative only).	Ensures no data loss during fragmentation.
Minimality	N/A (Qualitative only).	Avoids unnecessary complexity or redundancy.

PRACTICAL APPLICATION

1 Quantitative Analysis:

- Use minterm selectivity to identify frequently accessed data subsets and optimize query performance.
- Use access frequency to allocate resources (e.g., placing high-traffic fragments on high-performance nodes).

2 Qualitative Analysis:

- Use completeness to ensure data integrity and compliance with business or regulatory requirements.
- Use minimality to design a fragmentation strategy that is efficient and scalable.

EXAMPLE SCENARIO

Consider a distributed DBMS for a organisation with database schema:

- PAY(TITLE, SAL)
- EMP(ENO, ENAME, TITLE)

- **<u>PAY(TITLE, SAL) TABLE :</u>**

TITLE	SAL
MANAGER	65K
ELECTRICAL ENGINEER	52K
PREGRAMMER	45K
ACCOUNTANT	42K

PAY-1

PAY-2

Simple Predicate

P_1	TITLE = MANAGER
P_2	TITLE = ELECTRICAL ENGINEER
P_3	TITLE = PREGRAMMER
P_4	TITLE = ACCOUNTANT
P_5	SAL $\leq$ 50K
P_6	SAL $>$ 50K

Some Minterm Predicates And Minterm Selectivity [Sel(M_i)]

M_i	Minterm Predicate	Sel(M_i)
M_1	P_1 ^ P_2	0
M_2	P_1 ^ P_6	1
M_3	P_3 ^ P_6	1
M_4	P_4 ^ P_6	0

- **Access frequency:**

The Sal > 50K is accessed 50 times per day.

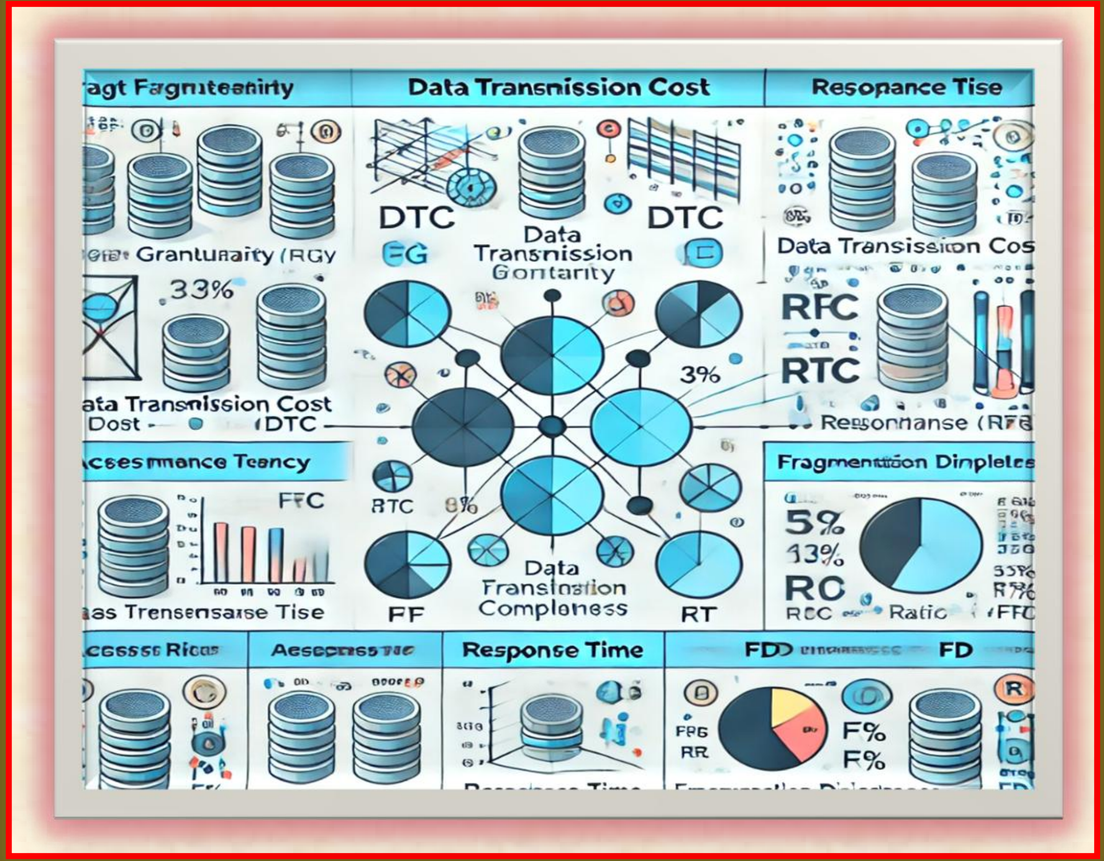
- **Qualitative Information:**
  - **Completeness:** All order records are preserved across fragments, ensuring no data loss.
  - **Minimality:** The fragmentation strategy avoids creating too many small fragments, reducing management overhead.

CONCLUSION

Effective fragmentation in a Distributed DBMS requires both quantitative and qualitative considerations. Minterm selectivity and access frequency help optimize performance, while completeness and minimality ensure data integrity and efficiency. Balancing these factors leads to better query performance, reduced redundancy, and improved scalability in distributed databases.

REFERENCES

- **"Distributed Databases: Principles & Systems"** – Stefano Ceri, Giuseppe Pelagatti
- **"Principles of Distributed Database Systems"** – M. Tamer Özsu, Patrick Valduriez
- **"Fundamentals of Database Systems"** – Ramez Elmasri, Shamkant B. Navathe
- **"Distributed Database Management Systems: A Practical Approach"** – Saeed K. Rahimi, Frank S. Haug
- **"Database System Concepts"** – Abraham Silberschatz, Henry F. Korth, S. Sudarshan



Thank you

ABHIRUP BAG

abhirup7477@gmail.com

ROLL NO. : 13000122082