SUBMITTED TO:

Dr. Muhammad Naeem

SUBMITTED BY:

Omair Ahmed (FA16-BCS-015)

Hamza Arif (FA16-BCS-026)

Umer Arif (FA16-BCS-028)

Osama Rana (FA16-BCS-030)

Areeba Abid (FA16-BCS-036)

Sheikh Muhammad Abdullah (FA16-BCS-120)

Table of Contents

INTRODUCTION

The question is "Why do we need fragmentation?" Consider fragmenting your tables if improving at least one of the following is your goal:

- **Single-user response time**
- **Concurrency**
- **Availability**
- **Backup-and-restore characteristics**
- **Loading of data**

Each of the preceding goals has its own implications for the fragmentation strategy that you ultimately implement. Your primary fragmentation goal determines, or at least influences, how you implement your fragmentation strategy. When you decide whether to use fragmentation to meet any of the preceding goals, keep in mind that fragmentation requires some additional administration and monitoring activity.

DATA FRAGMENTATION

Fragmentation separately is defined as the process or state of breaking or being broken into different smaller fragments.

Data fragmentation whereas allows you to break a single object into two or more segments. The object might be a user's database, a system database, or a table. Each fragment can be stored at any site over a computer network. Information about data fragmentation is stored in the distributed data catalogue (DDC), from which it is accessed by the TP to process user requests.

TYPES OF FRAGMENTATION

There are mainly three types of fragmentation

- Vertical Fragmentation
- Horizontal Fragmentation
- Mixed Fragmentation

VERTICAL FRAGMENTATION

Vertical fragmentation refers to the division of a relation into attribute (column) subsets. Each subset (fragment) is stored at a different node, and each fragment has unique columns—with the exception of the key column, which is common to all fragments.

Example:

Vertical fragmentation of PROJ relation

PROJ1: information about project budgets

PROJ2: information about project names and locations

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ₁

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000
P5	500000

PROJ₂

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Develop.	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris
P5	CAD/CAM	Boston

HORIZONTAL FRAGMENTATION

Horizontal fragmentation refers to the division of a relation into subsets (fragments) of tuples (rows). Each fragment is stored at a different node, and each fragment has unique rows. However, the unique rows all have the same attributes (columns). In short, each fragment represents the equivalent of a SELECT statement, with the WHERE clause on a single attribute

Example:

Horizontal fragmentation of PROJ relation

PROJ1: projects with budgets less than 200, 000

PROJ2: projects with budgets greater than or equal to 200, 000

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ₁

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York

PROJ₂

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

Advantages of fragmentation

Four major advantages of fragmenting a relation are as under:

Usage

In general, applications work with views rather than entire relations. Therefore, for data distribution, it seems appropriate to work with subsets of relation as the unit of distribution.

Efficiency

Data is stored close to where it is most frequently used. In addition, data that is ,not needed by' local applications is not stored.

Parallelism

With fragments as the unit of distribution, a transaction can be divided into several sub queries that operate on fragments. This should increase the degree of concurrency, or parallelism, in the system, thereby allowing transactions that can do so safely to execute in parallel.

Security

Data not required by local applications is not stored, and consequently not available to unauthorized users.

Disadvantages of fragmentation

Fragmentation has two primary disadvantages:

Performance

The performance of global application that requires data from several fragments located at different sites may be slower.

Integrity

Integrity control may be more difficult if data and functional dependencies are fragmented and located at different sites.

How We Applied Fragmentation

We have applied the fragmentations to the given project the techniques we have used are as under

Horizontal Fragmentation

For the horizontal fragmentation, we have given by default value and an object name. So, to apply the fragmentation we have decided to distribute records into multiple tables in the basis of value, the algorithm we have implemented works in such a way that if the value is greater than or equal to 100 the values are added to sillyobject in database and in the other case the values are added to testobject.

Vertical Fragmentation

For the vertical fragmentation, we have given by default value and an object name. So, to apply the fragmentation we have decided to add a city attribute which can be either "Islamabad" or "Lahore" and to distribute records into multiple tables we have taken the basis of city, the algorithm we have

implemented works in such a way that if the city is "Islamabad" the values are added to Obj1 in database and in the other case which is "Lahore" the values are added to Obj2.

Conclusion

From the above analysis and information we have learnt different sort of fragmentation techniques and have seen the advantages and drawbacks of different types fragmentation techniques, then we have learned the techniques how we apply fragmentation of any type .i.e. horizontal and vertical fragmentation.