

Name: Junaid Girkar  
Sap ID: 60004190057  
Division: SE A (A3) Comps

## **CASE STUDY ON DATA FRAGMENTATION**

### **INTRODUCTION TO FRAGMENTATION**

Fragmentation is the task of dividing a table into a set of smaller tables. The subsets of the table are called fragments.

Fragmentation is a database server feature that allows you to control where data is stored at the table level. Fragmentation enables you to define groups of rows or index keys within a table according to some algorithm or scheme. You can store each group or fragment (also referred to as a partition) in a separate dbspace associated with a specific physical disk. You use SQL statements to create the fragments and assign them to dbspaces.

Information about the fragmentation of the data is stored in DDC. When user sends a query, this DDC will determine which fragment to be accessed and it points that data fragment.

Fragmentation of data can be done according to the DBs and user requirement. But while fragmenting the data, below points should be kept in mind:

- **Completeness:** While creating the fragment, partial records in the table should not be considered. Fragmentation should be performed on whole table's data to get the correct result.
- **Reconstructions:** When all the fragments are combined, it should give whole table's data. That means whole table should be able to reconstruct using all fragments
- **Disjointedness:** There should not be any overlapping data in the fragments. If so, it will be difficult to maintain the consistency of the data. Effort needs to be put to create same replication in all the copies of data.

### **WHY AND WHEN DO WE USE FRAGMENTATION?**

Fragmenting tables is required if at least one of the following is the goal:

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

Hence fragmenting the relation in the database allows:

- **Easy usage of Data:** It makes most frequently accessed set of data near to the user. Hence these data can be accessed easily as and when required by them.
- **Efficiency:** It in turn increases the efficiency of the query by reducing the size of the table to smaller subset and making them available with less network access time.
- **Security:** It provides security to the data. That means only valid and useful records will be available to the actual user. The DB near to the user will not have any unwanted data in their DB. It will contain only the information, which is necessary.
- **Parallelism:** Fragmentation allows user to access the same table at the same time from different locations. Users at different locations will be accessing the same table in the DB at their location, seeing the data that are meant for them. If they are accessing the table at one location, then they have to wait for the locks to perform their transactions.
- **Reliability:** It increases the reliability of fetching the data. If the users are located at different locations accessing the single DB, then there will be huge network load. This will not guarantee that correct records are fetched and returned to the user. Accessing the fragment of data in the nearest DB will reduce the risk of data loss and correctness of data.
- **Balanced Storage:** Data will be distributed evenly among the databases in DDB.

### **TYPES OF FRAGMENTATION**

Fragmentation can be of three types: horizontal, vertical, and hybrid (combination of horizontal and vertical).

#### **ORIGINAL TABLE FOR EXAMPLE IMAGE 1 AND 2:**

PNo	Pname	Budget	Location
P1	Banking	250000	Mumbai
P2	CAD/CAM	500000	Hyderabad
P3	Insurance	400000	Pune

A University database keeps records of all registered students in a Student table having the following schema:

STUDENT

Regd_No	Name	Course	Address	Semester	Fees	Marks
---------	------	--------	---------	----------	------	-------

## 1. Vertical Fragmentation

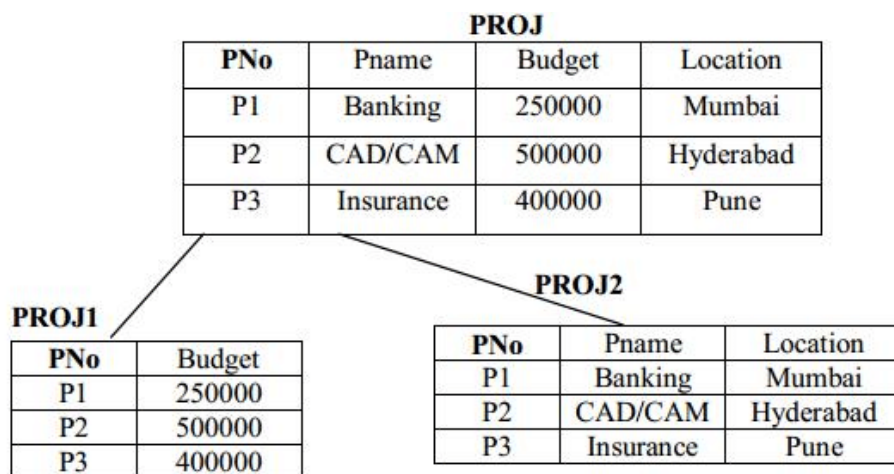
In vertical fragmentation, the fields or columns of a table are grouped into fragments.

In order to maintain re-constructiveness, each fragment should contain the primary key field(s) of the table.

Vertical fragmentation can be used to enforce privacy of data.

**For example**, in the student schema, the fees details are maintained in the accounts section. In this case, the designer will fragment the database as follows

```
CREATE TABLE STD_FEES AS
SELECT Regd_No, Fees
FROM STUDENT;
```



**EXAMPLE IMAGE 1**

## 2. Horizontal Fragmentation

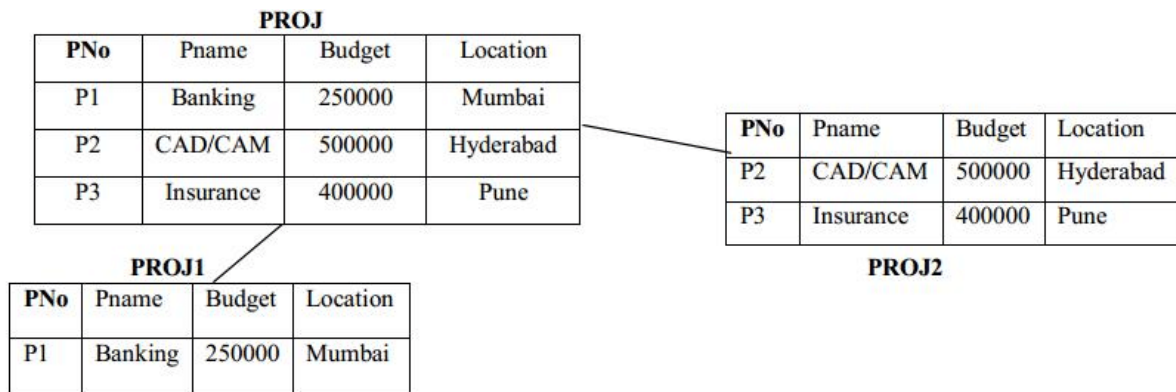
Horizontal fragmentation groups the tuples of a table in accordance to values of one or more fields.

Horizontal fragmentation should also confirm to the rule of re-constructiveness.

Each horizontal fragment must have all columns of the original base table.

Thus, this kind of horizontal fragmentation of the whole table into multiple sub-tables without changing the table structure is called Horizontal Fragmentation.

The concept is usually used to keep tuples (records) at the places where they are used the most, to minimize data transfer between far locations.



**EXAMPLE IMAGE 2**

**For example**, in the student schema, if the details of all students of Computer Science Course needs to be maintained at the School of Computer Science, then the designer will horizontally fragment the database as follows:

```
CREATE COMP_STD AS
  SELECT * FROM STUDENT
  WHERE COURSE = "Computer Science";
```

## Types of Horizontal Fragmentation

### 1. Primary Horizontal Fragmentation (PHF)

Primary Horizontal Fragmentation is a table fragmentation technique in which we fragment a single table and this fragmentation is row-wise and using a set of simple conditions.

#### Example:

Acc_No	Balance	Branch_Name
A_101	5000	Pune
A_102	10,000	Baroda
A_103	25,000	Delhi

For the above table we can define any simple condition like, Branch\_Name= 'Pune', Branch\_Name= 'Delhi', Balance < 50,000

Fragmentation1:

```
SELECT * FROM Account WHERE Branch_Name= 'Pune' AND Balance < 50,000
```

Fragmentation2:

```
SELECT * FROM Account WHERE Branch_Name= 'Delhi' AND Balance < 50,000
```

## 2. Derived Horizontal Fragmentation (DHF)

The process of creating horizontal fragments of a table in question based on the already created horizontal fragments of another relation (for example, base table) is called Derived Horizontal Fragmentation. DHF is defined on a member relation of a link according to a selection operation specified on its owner.

Given a link L where  $\text{owner}(L)=S$  and  $\text{member}(L)=R$ , the derived horizontal fragments of R, are defined as  $R_i = R \text{ F } S_i$ ,  $1 \leq i \leq w$  where w is the maximum number of fragments that will be defined on R and  $S_i = \sigma F_i (S)$  where  $F_i$  is the formula according to which the primary horizontal fragment  $S_i$  is defined.

Given link L1 where  $\text{owner}(L1) = \text{SKILL}$  and  $\text{member}(L1) = \text{EMP}$

$\text{EMP1} = \text{EMP SKILL1}$

$\text{EMP2} = \text{EMP SKILL2}$

where

$\text{SKILL1} = \sigma \text{SAL} \leq 30000 (\text{SKILL})$

$\text{SKILL2} = \sigma \text{SAL} > 30000 (\text{SKILL})$

EMP <sub>1</sub>			EMP <sub>2</sub>		
ENO	ENAME	TITLE	ENO	ENAME	TITLE
E3	A. Lee	Mech. Eng.	E1	J. Doe	Elect. Eng.
E4	J. Miller	Programmer	E2	M. Smith	Syst. Anal.
E7	R. Davis	Mech. Eng.	E5	B. Casey	Syst. Anal.
			E6	L. Chu	Elect. Eng.
			E8	J. Jones	Syst. Anal.

## 3. Hybrid Fragmentation

In hybrid fragmentation, a combination of horizontal and vertical fragmentation techniques is used.

This is the most flexible fragmentation technique since it generates fragments with minimal extraneous information.

However, reconstruction of the original table is often an expensive task.

Hybrid fragmentation can be done in two alternative ways

At first, generate a set of horizontal fragments; then generate vertical fragments from one or more of the horizontal fragments.

At first, generate a set of vertical fragments; then generate horizontal fragments from one or more of the vertical fragments.

Example: Consider the following table which consists of employee information.

Emp_ID	Emp_Name	Emp_Address	Emp_Age	Emp_Salary
101	Hrithik	Hyderabad	48	15000
102	Junaid	Pune	53	12000
103	Kanaad	Mumbai	32	20000

Fragmentation1:

SELECT \* FROM Emp\_Name WHERE Emp\_Age < 40

Fragmentation2:

SELECT \* FROM Emp\_Id WHERE Emp\_Address= 'Pune' AND Salary < 14000

### **ADVANTAGES OF FRAGMENTATION**

- Since data is stored close to the site of usage, efficiency of the database system is increased.
- Local query optimization techniques are sufficient for most queries since data is locally available.
- Since irrelevant data is not available at the sites, security and privacy of the database system can be maintained.

### **DISADVANTAGES OF FRAGMENTATION**

- When data from different fragments are required, the access speeds may be very high.
- In case of recursive fragmentation, the job of reconstruction will need expensive techniques.
- Lack of back-up copies of data in different sites may render the database ineffective in case of failure of a site.

### **CONCLUSION**

Data can be stored in different computers by fragmenting the whole database into several pieces called fragments. Each piece is stored at a different site. To design an effective distributed model, it is important to manage an appropriate methodology for data fragmentation and fragment allocation. In this case study, we studied the necessity of fragmentation in distributed database systems and we studied the various types of data fragmentations and the advantages and disadvantages of data fragmentation.