ISSN: 2393-8528

IJICSE
Open Access

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 4 Issue 6; November-December 2017; Page No. 01-06

# FRAGMENTATION: Easy Way to Retrieve Records in DDBMS

**Preetpal Kaur**

*Assistant Professor in Computer Science*

*Guru Nanak National College(co-ed) Nakodar (Punjab)*

**ABSTRACT**

In this paper, fragmentation technique of distributed database system is used to retrieve data fast and generate new table from social network user database. This database has the data of users related to social networking site. The data can be retrieved according to the requirement of any organization with the help of fragmentation technique. In fragmentation, the new table will be generated from the global relation with required fields and rows. However, selected by rows, selected by columns and combination of both selections are provided by fragmentation in distributed database management system. In this scenario, DDBMS distribute data or set of data over more than one computer for ease of availability. Fragmentation is the key feature of DDBMS.

**Keywords:** *DDBMS; Subset; Global Relation, Hybrid Fragmentation*

## I. INTRODUCTION

A distributed database is a database technique in which storage devices are not all attached to common processor .It may be stored on multiple computers and can share or access data from any computer [1]. DDBMS shows the relationship between database technology and computer network. It provides data independence of application logically and physically. It represented data at local level or global level. For an instance, in social network site database where organization required only specific fields from whole user database. For this purpose; DDBMS provides fragmentation technique. In this technique, selected rows and selected columns generate a new table from the original table. The purpose of fragmentation is to determine non overlapping fragment which are "logic unit of allocation" [2].In this paper, social network user database used to justified the fragmentation technique.

## II. Fragmentation in DDBMS

### *a). Horizontal Fragmentation*

Fragment data from table according to row wise and generate new table is horizontal fragmentation. It distributes the selected data from original relation to another new relation row wise [horizontally].

***Horizontal Fragmentation=All columns+ selected Rows***

The new relation will be generated by fragment the existing relation row wise and selected items can choose.

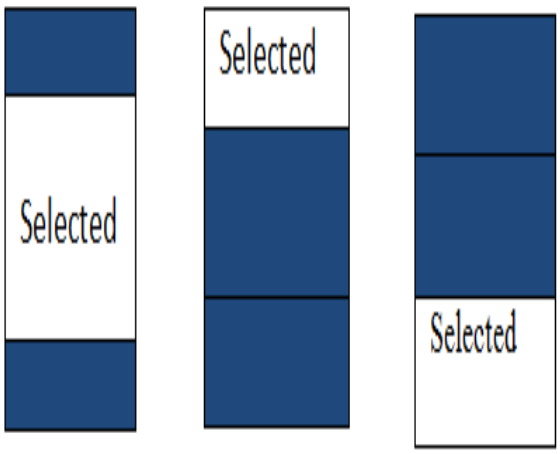


**Figure 1: Select particular row by horizontal fragmentation**

### *b) Vertical Fragmentation*

Fragment data from table according column wise and generate new table. When application require particular column then this task can perform by vertical fragmentation.

***Vertical fragmentation=All rows+ Selected columns***

**Corresponding author: Preetpal Kaur*

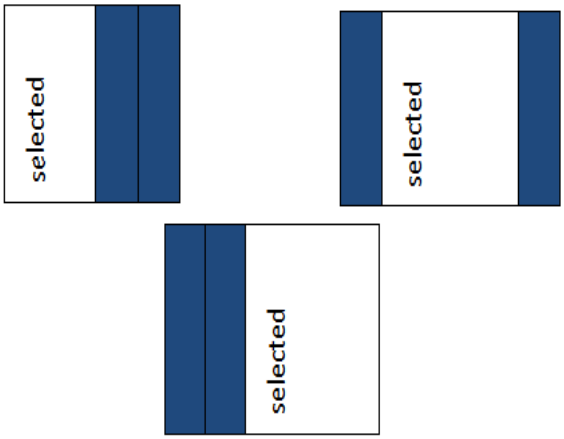


**Figure 2: select particular column by vertical fragmentation**

### *c) Mixed Fragmentation*

It is a combination of both fragmentations. By this, we can fragment and allocate the table row wise as well as column wise.

***Mixed Fragmentation=selected rows+ selected columns***

Two types of mixed fragmentation are:

a) Vertical fragmentation followed by horizontal fragmentation

b) Horizontal Fragmentation followed by vertical fragmentation

## III. Proposed Design

Fragmentation and distribution of data are major design issues of DDBMS. The efficiency and performance of DDBMS is mainly measured by its proper design and by network communication cost between site[3].In this paper, the design of system arise b/w social network user database with fragmentation techniques to generate new required table. The following flowchart shows the steps to reach at the final step of selection data.

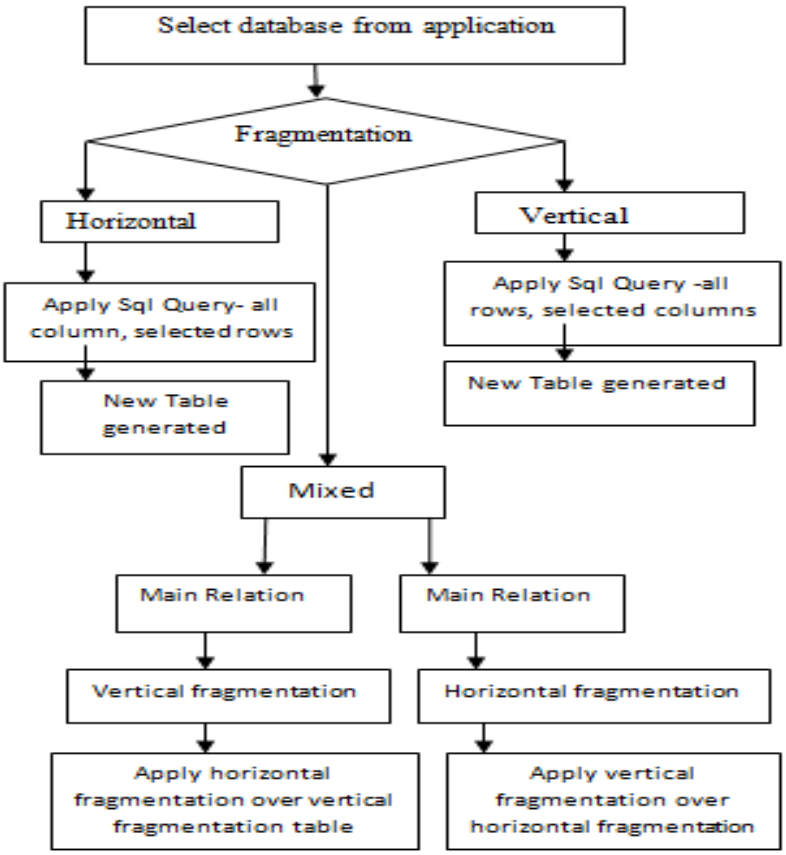


**Figure 3: Steps of Fragmentation**

## IV. Methodology

To achieve selected data row wise or column wise with efficiency, fragmentation technique of DDBMS is best option. It will generate new table as record and fast data retriever. In this paper, database of social network user data used and apply fragmentation techniques over it. All three techniques of fragmentation will be apply to database.

**Table 1: User_Details**

| id | User Name | Total Friends | Group | Like Count 7days | Comment Count 7days | Share Count 7days |
|---|---|---|---|---|---|---|
| 1 | John | 50 | 3 | 55 | 250 | 150 |
| 2 | michal | 155 | 6 | 225 | 150 | 200 |
| 3 | Anny | 250 | 7 | 567 | 200 | 25 |
| 4 | Roy | 90 | 10 | 200 | 350 | 250 |
| 5 | Jimmy | 35 | 15 | 150 | 200 | 100 |
| 6 | Maria | 200 | 4 | 135 | 350 | 55 |

From Table1; we can fragment the table as row wise or column wise and generate new table.

***a) query 1:*** *Create table Table2: max_friends by select all columns from user_detail table with selection of total_Friends>=50.*

It will perform horizontal fragmentation because all columns are required with selected rows >=50.

**Solution:**

Create table max_friends as
Select * from user_detail
Where total_friends>=50;

Now the new table created with name max_friends from table user_detail as follow:

**Table 2: max_Friends**

| id | User Name | Total Friends | Group | Like Count 7days | Comment Count 7days | Share Count 7days |
|---|---|---|---|---|---|---|
| 1 | John | 50 | 3 | 55 | 250 | 150 |
| 2 | michal | 155 | 6 | 225 | 150 | 200 |
| 3 | Anny | 250 | 7 | 567 | 200 | 25 |
| 4 | Roy | 90 | 10 | 200 | 350 | 250 |

***b) query2:*** *Create table Table3: min_friends by select all columns from user_detail table with selection of total_Friend<50.*

**Solution:**

Create table min_friends as
Select * from user_detail
Where total_friends<50;

Now the new table created with name min_friends from table user_detail as follow:

***Table 3:min_Friends***

| id | User Name | Total Friends | Group | Like Count 7days | Comment Count 7days | Share Count 7days |
|---|---|---|---|---|---|---|
| 5 | jimmy | 35 | 15 | 150 | 200 | 100 |
| 6 | maria | 200 | 4 | 135 | 350 | 55 |

So, table 2( max_friends) and table3 (min_friends) perform horizontal fragmentation where the selected rows are chosen with all columns and generate new tables.

***c) query 3****: Create table Table4: user_detail1 by select columns as id,username,group,likecount_7days from user_detail table.(vertical fragmentation)*

**Solution:**

```
Create table user_detail1 as
Select id,username,group,likecount_7days
from user_detail;
```

Now, the Table4 created with nameuser_detail1 with selected columns but all rows as:

**Table 4: User_detail1**

| id | User name | Group | Like Count 7days |
|---|---|---|---|
| 1 | john | 3 | 55 |
| 2 | michal | 6 | 225 |
| 3 | anny | 7 | 567 |
| 4 | roy | 10 | 200 |
| 5 | jimmy | 15 | 150 |
| 6 | maria | 4 | 135 |

***d) query 4****: Create table user_detail2 by select columns as id,commentcount_7days,sharecount_7days from user_detail table.(vertical fragmentation)*

**Solution:**

```
Create table user_detail2 as
Select id, commentcount_7days, sharecount_7days
from user_detail;
```

Now, the Table 5 (user_detail2) with selected columns but all rows as:

**Table 5: user_detail2**

| id | Comment Count 7days | Share Count 7days |
|---|---|---|
| 1 | 250 | 150 |
| 2 | 150 | 200 |
| 3 | 200 | 25 |
| 4 | 350 | 250 |
| 5 | 200 | 100 |
| 6 | 350 | 55 |

***Now from e-h queries will be mixed fragmentation***

***e) query 5: vertical fragmentation over horizontal fragmentation***

*Create table user_detail3 by select columns all columns from user_detail table where likecount_7days<=150.(horizontal fragmentation)*

**Solution:**

Create table user_detail3 as
Select * from user_detail
where *likecount_7days<=150;*

**Table 6: user_detail3**

| id | User Name | Total Friends | Group | Like Count 7days | Comment Count 7days | Share Count 7days |
|---|---|---|---|---|---|---|
| 1 | John | 50 | 3 | 55 | 250 | 150 |
| 5 | Jimmy | 35 | 15 | 150 | 200 | 100 |
| 6 | Maria | 200 | 4 | 135 | 350 | 55 |

Now from above user_detail3 table, we can apply vertical fragmentation over horizontal according to next query.

***f) query6:vertical fragmentation over horizontal fragmentation***

*Create table user_detail4 by select id, sharecount_7days columns from user_detail3 table.(vertical fragmentation)*

**Solution:**

Create table user_detail4 as
Select *id,sharecount_7days*
from user_detail3;

**Table 7: user_detail4.**

| id | Share Count 7days |
|---|---|
| 1 | 150 |
| 5 | 100 |
| 6 | 55 |

***g) query 7: Horizontal fragmentation over vertical; fragmentation(Mixed Fragmentation)***

*Create table user_detail5 by select id, username, commentcount_7days columns from user_detail table.(vertical fragmentation)*

**Solution:**

Create table user_detail5 as
Select *id,username, commentcount_7days*
from user_detail;

**Table 8: user_detail5.**

| id | User Name | Comment Count 7days |
|---|---|---|
| 1 | John | 250 |
| 2 | michal | 150 |
| 3 | Anny | 200 |
| 4 | Roy | 350 |
| 5 | Jimmy | 200 |
| 6 | Maria | 350 |

***h) query 8: Horizontal fragmentation over vertical fragmentation(Mixed Fragmentation)***

*Create table user_detail6 by select all columns from user_detail5 table where commentcount_7days<=200;.(horizontal fragmentation over vertical fragmentation)*

**Solution:**

Create table user_detail6 as
Select * from user_detail5
Where commentcount<=200;

**Table 9: user_detail6.**

| id | User Name | Comment Count 7days |
|---|---|---|
| 2 | michal | 150 |
| 3 | Anny | 200 |
| 5 | Jimmy | 200 |

## V.CONCLUSIONS

From aforementioned discussion, it is clear that fragmentation has the ability to access and retrieve data very easily from database .It provides the feature to generate new table as well. By using SQL commands, we can only retrieve and save data in another data from the original huge database with different techniques of fragmentation. User can create separate record of data in the form of new relation. The selected new table can distribute and allocated to multiple connected computer in distributed database environment. However, the purpose to select the database regarding social network is just to derive selected record from database with the help of fragmentation commands so that speed of retrieving selected particular records could increase instead of selecting all records. Even it provides efficiency and availability of record with fast speed whenever the user required.

## REFERENCES


1. http://www.its.bldrdoc.gov/fs-1037/di-d12/_1750.html
2. Stefano ceri, pelagati G: Distributed databases principles and system,Mcgraw-Hill,2013.
3. J. Biskup, D.W. Embley, and J. Lochner. Reducing inference control to access control for normalized database schemas. Information Processing Letters, 106(1):8–12, 2008.
4. Sumathi, S., Esakkirajan, S., (2007). Fundamentals of Relational Database Management Systems. (pp. 96- 97).Springer-Verlag Berlin Heidelberg.
5. Omar, R. (2012). Personalized Patient Treatments. Retrieved from https:// www. experiencesaphana.com/ docs/DOC-1577.
6. M. Sami, and A. Habib, "Controller-Agent based approach for Solving Distributed Constraint Problem," LITIS Lab. – INSA of Rouen – France, 2006.