Analysis of OLAP queries execution for evaluating a fragmentation schema in data warehouse

Hacène Derrar
USTHB, Faculty of electronic and computer science, LSI
Bp 32 El Alia 16111, bab Ezzouar/ Algiers/Algeria
Hderrar@yahoo.fr

Mohamed Ahmed-Nacer
USTHB, Faculty of electronic and computer science, LSI
Bp 32 El Alia 16111, bab Ezzouar/ Algiers/Algeria
Anacer@mail.cerist.dz

Omar Boussaid
University of Lyon 2, ERIC Laboratory
5 avenue Pierre Mendes-France 69676 Bron Cedex-France
Omar.Boussaid@univ-lyon2.fr

Abstract— **Data fragmentation improves query performance and data administration. The fragmentation schema is designed from statistical analysis of data access collected from the execution of more frequent queries. The application of data fragmentation to data warehouse is faced with the evolution of data model and workload, due more particularly to the specific characteristics of OLAP queries. So, at the end of data fragmentation, there are no techniques to know that the implemented schema remain the more optimal and the produced partitions are relevant. We propose in this paper an approach based on analysis of the OLAP queries execution to evaluate the effectiveness and the relevance of the data fragmentation schema.**

Keywords: Data warehouse, data fragmentation, OLAP queries, optimization.

I. INTRODUCTION

Data warehousing applications [1,2,3] cope with enormous amounts of data ranging in Gigabytes and Terabytes. While transactional (OLTP, online transaction processing) DBMS like bank applications usually use simple query patterns to retrieve a very small part of a database (usually one record) by a primary key access, data processing in data warehousing (OLAP, online analytical processing) involves complex queries that usually access a large portion of the database [4,6].

The design for data warehouse (DW) is known as a star schema, which consists of a set of dimension tables and a fact table. Each dimension table is related to the fact table in a one-to-many fashion. The fact table contains a number of measures. A measure is usually of numerical type, which represents an important performance indicator such as sales, unit price, or number of units sold. The values for a measure are often called facts. Each fact value is associated with an array of dimension members, which relate the fact to its “context”. Data for dimensions are stored in dimension tables. Facts are stored in a fact table whose primary key is the composite of foreign keys from all dimension tables.

Data processing in DW applications retrieves aggregated measures organized or classified according to several dimensions or hierarchies over the dimensions. For this reason multidimensional data models MD, multidimensional query languages or the OLAP approach have been developed by the research community and implemented as commercial products. Typical OLAP operations are drill-down, roll-up and slice-and-dice [3] and usually multiple dimensions are restricted at the same time. In general one can state that these operations in a MD model lead to range restrictions on the lowest hierarchy level of each dimension [5].

So, clustering, indexing and fragmentation (also called partitioning) are common techniques to facilitate the administration, the management and the maintenance of data warehouse and more particularly to improve query performance.

Thus, traditional fragmentation techniques and more particularly horizontal and vertical fragmentation, developed in relational DBMS, were applied to the data warehouse.

Partitioning tables, indexes and materialized views in fragments stored and accessed separately improve significantly data manageability, accessibility and query execution time. Horizontal partitioning consists in dividing a relation into partitions with the same schema. Each one preserves part of the tuples according to restriction criteria. Vertical partitioning, it consists in dividing a relation into partitions of different schema, by projection with duplicating the key. As for a mixed fragmentation technique proposed simultaneously applies horizontal and vertical fragmentation on a relation (or vice versa) [12].

These approaches are designed from a statistical analysis of more frequent queries based on both qualitative and quantitative information. So, algorithms used to design an optimal partitioning schema are static algorithms. Their entries are bases on workload gathered from data exploitation. If a change occurs in the inputs of these algorithms, they must be rerun to determine a new optimal fragmentation schema. Moreover, these algorithms are based on the clustering principle which is considered as combinatorial problem and requires for its resolution to use heuristics methods [7]. So, in the case of models evolution and / or changes in workload these algorithms become very complicated, or unworkable.

In the context of relational and object oriented databases and in any environment (centralized, parallel, distributed) much of the literature has addressed this issue. Researchers concentrate their work on data redistribution or fragments reallocation in the event of performance degradation. So, it was considered that the solution lies at the physical level by applying load balancing strategies of treatment and data between nodes. The logical aspect, namely the design of the fragmentation schema, itself, remains adapted because the workload is almost stable.

Conversely, in data warehousing the evolution of data model and workload are dynamic. This is due more particularly to the specific characteristics of OLAP queries. So, an inappropriate and badly conceived fragmentation schema have a considerable influence on the system's performance and more particularly during the execution of the expensive operations such as the joint and the multi-joint which characterize the decisional queries. In [21] authors have clearly demonstrated through theorems and lemma that the choice of partition keys and how to arrange the records in the fact table have a great impact on the OLAP queries response time.

For efficient use of fragmentation technique in data warehouse, it is not only to analyze the data access frequencies to choose an optimal fragmentation schema, but to make that choice dynamic and adapted to changing workload.

Also, there are no techniques to know that the implemented schema remain the more optimal and the produced partitions are relevant. We propose in this paper an approach based on analysis of the OLAP queries execution to evaluate the effectiveness and the relevance of the data fragmentation schema in order to determine a fragmentation schema optimizing the performance. This paper is organized as follows, in the first section, we describe the problem generated by the fragmentation techniques and we motivate our approach as well as the awaited contribution. In section 3, we present our approach. In section 4, we present the experimental results obtained by using benchmark APB-1 release II implemented under Oracle 10g. Lastly, we finish this article by a conclusion and prospects.

II. MOTIVATION

There are three fragmentation approaches: vertical fragmentation [11], [15], [24], horizontal fragmentation [18] and hybrid fragmentation [12], [17],[19],[25]. Vertical fragmentation (VF) consists in dividing a relation into partitions of different schema, by projection with duplicating the key. It consists in grouping together attributes that are frequently accessed by queries.

Horizontal fragmentation (HF) consists in dividing a relation into partitions with the same schema using query predicates. Each partition preserves part of the tuples according to restriction criteria. It reduces query processing costs by minimizing the number of irrelevant accessed instances. Two versions of HF are cited by the researchers : primary HF and derived HF. Primary HF of a relation is performed using predicates that are defined on that relation. On the other hand, derived HF is the partitioning of a relation that results from predicates defined on another relation.

Finally, hybrid fragmentation consists of either horizontal fragments that are subsequently vertically fragmented or vertical fragments that are subsequently horizontally fragmented.

To design an optimal fragmentation schema, two types of algorithms are generally used: the minterms generation-based approaches [20], [21], and affinity-based approaches [22], [23]. The algorithms based on the completeness and the minimality of predicates requires a combinatorial calculation of access probabilities.

Regarding to the algorithms run by the affinity, is based on predicates or attributes affinity concept, where affinity defines query frequency. Their disadvantage lies in that the matrix of affinity is expressed only between pairs of predicates or attributes.

Thus, these algorithms are very complex. In the case of HF, if n simple predicates are considered to perform primary HF, 2^n is the number of horizontal fragments using minterm predicates. If there are k network nodes, the complexity of allocation horizontal fragments is $O(k^{2n})$. So, this is part of NP-hard problems and the selection of partitions requires the exploration of the entire search space. In the case of VF, the possible fragments are given by the Bell number which approximately $B(m) \approx m^m$ where m is non-primary key attributes of a relation.

With this number of possible fragments, the fragment allocation using a suitable cost model is of the complexity $O(k^{m^m})$ with k as the number of networks nodes

Furthermore, although it is most adapted for data warehouse, the derived horizontal fragmentation, which consists in splitting fact table based on fragmentation schemes of set of dimension tables, may increase the number of fragments of the fact tables dramatically and makes their maintenance very costly. For example: Let a star schema with d dimension tables and a fact table. Let g ($g \leq d$) is the number of fragmented dimension tables. The number of horizontal fragments of the fact table (denoted by N) is given by: $N = \prod_{i=1}^{g} m_i$ where mi is the number of fragments [7], [8].

All fragmentation approaches, horizontal, vertical or mix, are based at the time of their design on the analysis of statistical data collected starting from the execution of most frequent queries. So, the adaptation of fragmentation techniques to data warehouse proving more delicate because mainly of nature of the OLAP queries. These queries are long, complex and require sometimes a great number of selection operations and aggregation. They can handle hundreds not to say thousands of tuples. The analytical queries are extremely variable, they are made up generally in an interactive way and can be executed once or many times. This type of queries called also ad hoc queries corresponds to queries seized on line without a long preliminary reflection [9]. All these characteristics makes, with time, the fragmentation schema, inappropriate since it was conceived starting from unstable statistical data.

In addition, works which treated the adaptation of the fragmentation techniques are focused on the logical aspect by completely dissociating it from the physical aspect of the data warehouse design. Indeed, the fragmentation schema design of and the strategy of the fragments placement are two completely dependent approaches. They are based at the time of their designs on the same information collected at the time of data exploitation, in order to achieve the same objective, namely the improvement of the system's performances.

Of this fact, the fragmentation techniques, initially conceived for performances improvement, can constitute in the data warehouse the main obstacles for a better data exploitation. Indeed, that's not a question to carry out a simple application of these techniques but also to ensure a perfect adaptability to the specific characteristics of data warehouse. To profit fully from their advantages, the fragmentation techniques and data placement must be constantly re-examined and adapted to data warehouse exploitation. In this framework, we propose an approach based on the access frequencies of the decisional queries to evaluate the effectiveness of the implemented fragmentation schema.

Our approach consists to evaluating the effectiveness of the existing fragmentation schema. Nevertheless, to measure the quality of a fragmentation schema which one does not know a priori the used objective function is a task which is not always obvious. Furthermore, the majority of the logical design algorithms of fragmentation are directed by the affinity measurement. i.e. the calculation of the queries access frequency only between one pair of attributes which does not allow, consequently, to measure affinity between all the attributes of a partition.

It proves necessary to define an approach making possible to evaluate and measure the affinity of a fragmentation schema in order to study the possibility of determining another more optimal schema. For this purpose, we describe below an approach based on a new objective function for evaluating an fragmentation schema. This objective function will have to be generic and flexible allowing to take possibly into account of another metric such as: the type of queries, the data placement information, the storage capacity and the transfer cost of the data between sites.

III. EVALUATING A FRAGMENTATION SCHEMA

Since any fragmentation approach is conceived starting from information relating to the access frequency of queries. The idea consists in using this common denominator between all the approaches to define a new cost function. This one will make it possible to evaluate a fragmentation schema according to the access frequency of the queries to the various fragments. thus, to measure the quality of a fragmentation schema, we adapt a technique of evaluation used in the statistical field to knowing the criterion of the square error. It consists of evaluating a fragmentation schema by calculating the square error of the access frequency of the queries on the attributes of the various fragments.

The formulation, below, relating to the definition of our objective function was inspired by works of Jain and Dubes [13] and Muthuraj [14], who used it within the framework of a method of unsupervised learning for the regrouping of attributes. For reasons of simplification, one considers in our case a data warehouse evolving in a context of centralized treatment and whose fact table is fragmented according to a vertical approach. We consider that the dimension tables are small size and consequently, they will not be fragmented.

It should be noted that the work of Benmessahel et al. in [26] used the square error function to design the vertical fragmentation schema using PSO algorithm.

Moreover, our approach does not consider a matrix of affinity of attributes, but a matrix of use of attributes composed, in columns of the attributes and lines of the frequent queries. The terms of the matrix are the access frequencies of the queries to the attributes. The calculation of square error will make possible to measure the affinity of the partitions of different sizes. let us consider the following formulation:

n : total number of the attributes of the fact table;
Q : total number of the frequent queries;
f_q : access frequency of the query q for $q = 1,2. , Q$;
M : total number of fragments ;
n_i : number of attributes in fragment i ;
f_{qj}^{i}: the access frequency of the query q to the attribute j in fragment i, with

$$f_{qj}^{i} \neq 0 ;$$

A_{ij} : the vector attribute of the attribute j in fragment i, where f_{qj}^{i}: is a component of this vector;
S_{iq} : the whole of attributes of fragment i acceded by the query q; equal to 0 if the query q does not access to the fragment I ;
$| S_{iq} |$: number of attributes of fragment i acceded by the query q;

Given a fact table F of N attributes vertically fragmented in M fragments $(F_1, F_2, \ldots, F_M)$ containing each one n_i attributes. Thus $\sum_{i=1}^{M} n_i = n$. The average vector Vi for fragment i is defined by :

$$V_i = \frac{1}{n_i} \sum_{j=1}^{n_i} A_{ij} \qquad 0 < i < M \tag{1}$$

The average vector V represents the average of the access of the queries to all the attributes of fragment i. For a attributes vector A_{ij}, $(A_{ij} - V_i)$ is called « the difference vector » of the attribute j in fragment i. The square error for the fragment F_i is the sum of the squares of the length of the differences vectors of all attributes in fragment i. It is calculated by the following formula:

$$e_i^2 = \sum_{j=1}^{n_i} (A_{ij} - Vi)^Q (A_{ij} - Vi) \qquad 0 < i < M \tag{2}$$

If $A_{ij} = V_i$ then $e_i^2 = 0$. This case means: either that there is only one attribute in each fragment or that all the attributes in each fragments are necessary for the query. In this paper, one is interested in case where $A_{ij} \neq V_i$ in order to be able to compare the fragments according to the relevance of the attributes.

The square error of global fragmentation schema is calculated by the following formula:

$$E_M^2 = \sum_{i=1}^{M} e_i^2 \tag{3}$$

Then :

$$E_M^2 = \sum_{i=1}^{M} \sum_{j=1}^{n_i} (Aij - Vi)^Q (Aij - Vi) \tag{4}$$

Another writing of equation 4 will make possible to better perceive the contribution of each query for the calculate of square error of each fragment. Thus, the average vector V_i for fragment i can be defined as follows:

$$Vi = \begin{bmatrix} \frac{|s_{i1}| * f_1}{n_i} \\ \frac{|s_{i2}| * f_2}{n_i} \\ \dots\dots. \\ \dots\dots. \\ \frac{|s_{iq}| * f_q}{n_i} \end{bmatrix}$$

The attributes vector $A_{ij},$ whose components are the frequencies of access, is:

$$A_{ij} = \begin{bmatrix} f_{1j}^i \\ f_{2j}^i \\ \dots\dots. \\ \dots\dots. \\ f_{qj}^i \end{bmatrix}$$

From where :

$$E_M^2 \sum_{i=1}^{M}\sum_{j=1}^{n_i}\begin{bmatrix} f_{1j}^i - \frac{|s_{i1}| * f_1}{n_i}, \dots, f_{qj}^i - \frac{|s_{iq}| * f_q}{n_i}\end{bmatrix}^Q \begin{bmatrix} f_{1j}^i - \frac{|s_{i1}| * f_1}{n_i} \\ f_{2j}^i - \frac{|s_{i2}| * f_2}{n_i} \\ \dots\dots. \\ \dots\dots. \\ f_{qj}^i - \frac{|s_{iq}| * f_q}{n_i} \end{bmatrix} \quad (5)$$

In order to identify the components in which the attributes are not relevant for query, the equation can be written as follows:

$$E_M^2 = \sum_{i=1}^{M}\sum_{j=1}^{ni}\sum_{q=1}^{Q}\left[\delta_{jq} * f_q^2\left(1 - \frac{|s_{iq}|}{n_i}\right)^2 + (1 - \delta_{jq})\left(f_q * \frac{|s_{iq}|}{n_i}\right)^2\right] \quad (6)$$

Where : δ_{jq} = 1 if the attribute j is acceded by the query q; 0 if not. The first term $f_q^2\left(1 - \frac{|s_{iq}|}{n_i}\right)^2$ represents the accesses to the fragments containing the relevant attributes (required by the query) and the second term represents the accesses to fragments containing of the nonrelevant attributes (attributes which are not required by the query).

Then :

$$E_M^2 = \sum_{i=1}^{M}\sum_{q=1}^{Q}\left[|s_{iq}| * f_q^2\left(1 - \frac{|s_{iq}|}{n_i}\right)^2 + (n - |s_{iq}|)\left(f_q * \frac{|s_{iq}|}{n_i}\right)^2\right] \quad (7)$$

Where :

$\sum_{j=1}^{ni} \delta_{jq} = |s_{iq}|$ and $\sum_{j=1}^{ni}(1 - \delta_{jq}) = (n_i - |s_{iq}|)$

Thus :

$$E_M^2 = \sum_{i=1}^{M}\sum_{q=1}^{Q}\left[f_q^2 * |s_{iq}|\left(1 - \frac{|s_{iq}|}{n_i}\right)^2 + f_q^2 * (n_i - |s_{iq}|)\left(\frac{|s_{iq}|}{n_i}\right)^2\right] \quad (8)$$

If $n_i = |s_{iq}|$, then $E_M^2 = 0$. What means that with each execution, the query q accedes all the attributes of fragment i. One can always reduce equation 8 as follows:

$$E_M^2 = \sum_{i=1}^{M}\sum_{q=1}^{Q}\left[f_q^2 * |s_{iq}|\left(1 + \frac{|s_{iq}|^2}{{n_i}^2} - 2 * \frac{|s_{iq}|}{n_i}\right) + f_q^2 |s_{iq}|(n_i|s_{iq}|)\left(\frac{|s_{iq}|}{{n_i}^2}\right)\right] \quad (9)$$

By simplification of the equation above, one will have:

$$E_M^2 = \sum_{i=1}^{M}\sum_{q=1}^{Q}\left[f_q^2 * |s_{iq}|\left(1 - \frac{|s_{iq}|}{n_i}\right)\right] \quad (10)$$

This equation is the same one as equation 4, in another form. We perceive according to this equation the contribution of the access to the fragments containing attributes which are not required by queries for the calculation of E_M^2. It means that the value of E_M^2 is proportional to the cost due to the fragments access containing a nonrelevant attributes. More the value of this error approaches 0 plus the fragmentation schema is optimal.

IV. Experimental Study

To test our approach, we used benchmark APB-1 release II Council (1998) implemented under Oracle 10g. This benchmark, uses a star schema made up of four dimension tables (Prodlevel of 9000 tuples, Custlevel of 900 tuples, Timelevel of 24 tuples and Chanlevel of 9 tuples) and a fact table (Actvars of 24.000.000 tuples). For the calculation of the queries execution times, we used the utility Aqua Data Studio 2.0.7. To carry out our tests, we used a set of 50 decisional queries including various operators: operations of joint, selection and the functions of calculation and aggregations (SUM, COUNT, AVG, MIN, MAX).

We started our test by executing this queries sets on a fragmented and non fragmented data warehouse whose results are illustrated in figure 1.

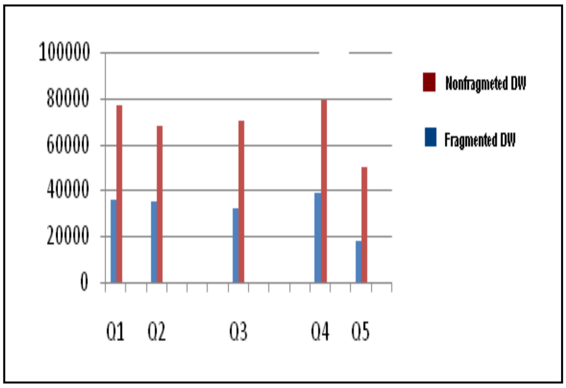

Figure 1. Queries execution time on fragmented and non fragmented data warehouse

We vertically fragmented the fact table “Actvar” on several fragmentation schema and we performed various tests by the calculation of square error according to different fragmentation schema. At the end of these tests, we noted that the number of fragments is inversely proportional to the value of the square error. More the number of fragments is large plus the access cost to the impertinent attributes becomes tiny (figure 2). What also shows the contribution of vertical fragmentation if the attributes of fragmentation were well defined.

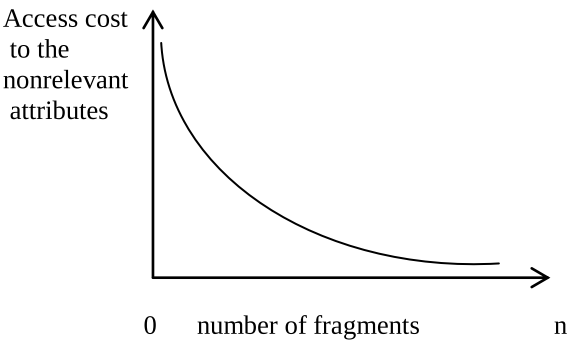

Figure 2. Influence of the number of fragments on the access cost to the attributes

To validate our approach, we considered a use matrix of attributes represented in Table 1. We conceived various fragmentation schemas. For each schema one calculated the value of his square error E_M^2 of attributes according to the access frequency of queries. The optimal fragmentation schema determined by the majority of the algorithms used for vertical fragmentation in particular those of the Navathe et al, [15,16] corresponds to the fragmentation schema whose its square error is minimal. This confirms indeed the validity of our approach.

Attributs / Requets	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
q1	0	25	0	0	25	0	25	0	0	0
q2	50	5	50	0	0	0	0	50	50	0
q3	0	0	0	25	0	25	0	0	0	25
q4	35	0	0	0	0	0	35	35	0	0
q5	25	25	25	0	25	0	25	25	25	0
q6	0	25	0	0	25	0	0	0	0	0
q7	0	0	25	0	0	0	0	0	25	0
q8	0	0	15	15	0	15	0	0	15	15

Table 1. Use Matrix of use of attributes

Results :

- Fragmentation schema 1 : one fragment consist of the attributes:(A1,A2,A3, A4,A5,A6,A7,A8,A9,A10)
  The square error for this case is : $\boldsymbol{E_M^2}$ **= 12577**
- Fragmentation schema 2 : consist of two fragments: F1: (A1,A4,A5,A6,A7,A10) ; F2: (A2,A3,A8,A9).
  The square error is : $\boldsymbol{E_M^2}$ **= 5949**
- Fragmentation schema 3: consist of three fragments : F1:(A1,A5,A7) ; F2:(A2,A3,A8,A9) ; F3:(A4,A6,A10).
  The square error for this schema is: $\boldsymbol{E_M^2}$ **= 3312**
- Fragmentation schema 4: consist of four fragments : F1:(A1,A5) ; F2:(A2,A3,A8,A9); F3:(A4,A6,A10); F4:(A7).
  The square error is: $\boldsymbol{E_M^2}$ **= 3519**

After this calculation according to the access frequencies of queries describe in the table above, we conclude that the fragmentation schema 4 is the optimal schema, because the value of its square error is the smallest $\boldsymbol{E_M^2}$ **= 3312**.

V. Conclusion

In this article we presented an approach for evaluating a fragmentation schema in data warehouse based on the analysis of the execution of OLAP queries in order to determine a fragmentation schema optimizing the performances. For this purpose, as was shown the access frequency of the queries to the data proves be an important parameter for the design of a fragmentation schema for the data warehouse framework. The use analyze of this parameter made possible to evaluate the relevance of the implemented fragmentation schema and to seek a more optimal schema bases on one gathering of the most questioned attributes. For our future works, we plan to apply our approach to the horizontal fragmentation and we extend square error formulation to be multi-objectives by considering other metric such as: queries execution time and transfer time of data.

REFERENCES

[1] B. Devlin. Data Warehouse from Architecture to Implementation. Addision-Wesley Longman, Inc. 1997.

[2] W.H. Inmon. Building the Data Warehouse. John Wiley & Sons, Inc., 2nd edition, 1996.

[3] R. Kimball. The Data Warehouse Toolkit. John Wiley & Sons, New York. 1996.

[4] J. Gray, A. Bosworth, A. Layman, H. Pirahesh: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. Proc. of ICDE 1996, pp. 152-159

[5] S. Sarawagi. Indexing OLAP data. Data Engineering Bulletin 20 (1), 1997, pp. 36-43.

[6] M.C. Wu and A.P. Buchmann. Research Issues in Data Warehousing. BTW'97. 1997.

[7] L.Bellatreche, K.Boukhalfa. (2005). An Evolutionary Approch to Schema Partitioning Selection in a Data Warehouse Environment, Proceeding of the International Conference on Data Warehousing and Knowledge Discovery (DAWAK'05).

[8] L.Bellatreche, Boukhelfa.K, H.I.Abdalla.Sage, (2006). A combinaison of genetic and simulated annealing algorithms for physical data warehouse design. In 23rd British National Conference on Database.

[9] G.Gardarin (2005). Base de données. Edition Eyrolles.

[11] M.Golfarelli , D.Maio, S.Rizzi (1999). Vertical Fragmentation of Views in Relational Data warehouse. Université de Bologne, Italie.

[12] E.Ziyati, L.Bellatreche, D.Aboutajdine (2006). Un Algorithme génétique pour la sélection d'un schéma de fragmentation mixte dans les entrepôts de données. Atelier Systèmes décisionnels. Agadir.Maroc.

[13] A.Jain, R. Dubes (1998). Algorithms for clustering Data. Prentice Hall Advanced Reference Series, Englewood Cli_s, NJ.

[14] J.Muthuraj. (1992), A formal approach to the vertical partitioning problem in distributed database design. Université de Florida. USA.

[15] S.Navathe, S. Ceri, G.Wierhold, et J.Dou (1984). Vertical Partitioning Algorithms for Database Design. ACM Transactions on Database Systems, Vol. 9, No. 4, December 1984, pages 680-710.

[16] B.Navathe, M. Ra (1989). Vertical Partitioning for Database Design: A Graphical Algorithm. ACM SIGMOD International Conference on Management of Data, 1989, pp. 44-450, Conference on Management of Data, 1989, .pp. 44-450,

[17] G.Sacco (1986) 'Fragmentation: A Technique for Efficient Query Processing', ACM Transaction on Database Systems, 11: 113-133.

[18] S.Ceri, Negri, M., and Pelagatti, G. (1982) 'Horizontal data partitioning in data base design', Proceeding of the ACM SIGMOD, International Conference on Management of Data. SIGPLAN Notices, page 128-136, 1982.

[19] Y.Zhang., and Orlowska, M.E. (1994) 'On Fragmentation Approaches for Distributed Database Design'. Information Science, 1: 117-132.

[20] K.Karlapalem, Navathe, S.B., and Ammar, M. (1996) 'Optimal redesign policies to support dynamic processing of applications on a distributed.

[21] A.Y.Noaman and Barker, K. (1999) 'A horizontal fragmentation algorithm for the fact relation in a distributed data warehouse', In th 8th International Conference on Information and Knowledge Management (CIKM'99), 154–161.

[22] M.T.Ozsu and Valduriez, P. (1999) 'Principles of Distributed Database Systems', Second Edition. Prentice Hall.

[23] R.Blankinship , Hevner, A.R., and Yao.S.B. (1991) 'An interative method for distributed database design', In Proceedings of the 17th International Conference on Very Large Data Bases, G.M. Lohman, A.Sernadas, and R.Camps, Eds., Morgan Kaufmann, pp.389-400.

[24] M. Bouakkaz, Ouinten, Y. ; Ziani, B. 'VERTICAL FRAGMENTATION OF DATA WAREHOUSES USING THE FP-MAX ALGORITHM'. International Conference on Innovations in Information Technology (IIT), Abu Dhabi , march 2012.

[25] N.Gorla, Vincent. N, Dik.M.L. ' Improving database performance with a mixed fragmentation design'. Journal of intelligent information systems,December 2012, volume 39, issue 3, pp 559-576.

[26] B.Benmessahel, Touahria. M. ' An improved Combinatorial Particle Swarm Optimization Algorithm to Database Vertical Partition. Journal of Emerging Trends in Computing and Information Sciences, 2011, volume 2.