International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

MULTI DIMENSIONALISED AGGREGATION IN HORIZONTAL DATASET USING ANALYSIS SERVICES

S. Aiswarya[1], S. Ramadevi[2]

[1]*PG Student, Department of Computer Science and Engineering, Arunai College of Engineering, Tiruvannamalai.*
[2]*Assistant Professor, Department of Computer Science and Engineering, Arunai College of Engineering, Tiruvannamalai.*

[1]E-mail: aishusanjeevi@gmail.com
[2]E-mail: spsubashini84@yahoo.co.in

Abstract

Projecting data in different dimensions is the core concept taken for this project. Preparing a data set for analysis is generally the most time consuming task in a data mining project. In the existing system they used simple, yet powerful, methods to generate SQL (Structured Query Language) code to return aggregated columns in a horizontal tabular layout, returning a set of numbers instead of one number per row. This new class of functions is called horizontal aggregations. [1] The horizontal aggregation is evaluated using three fundamental methods: case, SPJ (Select Project Join) and pivot. Transforming normal data into knowledge cube is one of the emerging fields in the current market. In the proposed system, a new standard of pivoting option is incorporated using Data mining. This can be achieved with the tool SAAS (SQL Server Analysis Services). The data will be taken and it will be transformed into knowledge cubes. This can be achieved with MDX (Multi Dimensional eXpression) queries. On top of that, the knowledge data will be customized based on "Generalized and Suppression Algorithm". In addition to this, the performance efficiency among case, SPJ and pivot methods will be analyzed.

***Keywords*- Aggregation, data cube, pivoting, data preparation, SQL.**

I. INTRODUCTION

Data mining is the process of extracting knowledge from large amount of data. It has attracted a great deal of attention in the information industry and in society as a whole in recent years due to the wide availability of huge amount of data and the imminent need for turning such data into useful information and knowledge. Data can be stored in many different kinds of databases and information repositories. One such data repository architecture that has emerged is the data warehouse. Data warehouse technology includes OLAP (Online Analytical Processing), that is, analysis technique with functionalities such as summarization, consolidation and aggregation. [5] Data aggregation is a process in which information is gathered and expressed in a summary form, and which is used for purposes such as statistical analysis. The most commonly used aggregation is the sum of a column and other aggregation operators return the average, maximum and minimum over group of rows. A new class of aggregation function called Horizontal aggregation, represents an extended form of traditional SQL (Structured Query Language) aggregation, which returns set of values in a horizontal layout. [6] Horizontal aggregation is evaluated using three fundamental methods: case, SPJ (Select Project Join) and pivot.

Case: This method uses the "case" programming construct available in SQL. The case statement returns a value selected from a set of values based on Boolean expressions. SPJ: It is interesting from a theoretical point of view because it is based on relational operators only. The basic idea is to create one table with vertical aggregation for each result column, and then join all those tables to produce another table. [1] Pivot: It is a built-in operator offered by some DBMS, which transforms row to columns. This method internally needs to determine how many columns are needed to store the transposed table and it can be combined with the GROUP BY clause. Data set for analysis is generally the most time consuming task in a data mining project, requiring many complex SQL queries, joining tables and aggregating columns. To overcome this problem, multidimensional data cube is used. It is also called as OLAP cubes. [13] A data cube is a collection of data that's been aggregated to allow queries to return data quickly. Transforming normal data into knowledge cube is one of the emerging fields in the current market. The data will be taken and it will be transformed into knowledge cubes. The data cube is created using the tool SAAS (SQL Server Analysis Services). Microsoft SQL Server OLAP Services provides architecture for access to multidimensional data.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

This data is summarized, organized and stored in multidimensional structure for rapid response to user queries. For expressing queries to multidimensional data, Microsoft SQL Server OLAP Services employs full-fledged, highly functional expression syntax: MDX (Multi Dimensional eXpression). The MDX expression can be used to view the actual output.

The paper is organized as follows: The reviews of the Related Works are present in the Section 2. Section 3 describes the proposed method for transforming normal data to knowledge cube, Section 4 describes the Result and Analysis and Conclusion and Future work described in Section 5.

II. Related Works

C. Cunningham (2004) developed two operators: Pivot and Unpivot. PIVOT and UNPIVOT are two operators on tabular data that exchange rows and columns, enable data transformation useful in data modeling, data analysis and data presentation. They can quite easily be implemented inside a query processor, much like select, project and join. Such a design provides opportunities for better performance, both during query optimization and query execution. This paper, discuss query optimization and execution implications of this integrated design and evaluate the performance of this approach using a prototype implementation in Microsoft SQL Server.

C. Ordonez (2004) introduced two aggregation functions. The first function returns one row for each percentage in vertical form like standard SQL aggregations. The second function returns each set of percentages adding 100% on the same row in horizontal form. These novel aggregate functions are used as a framework to introduce the concept of percentage queries and to generate efficient SQL code. Experiments study different percentage query optimization strategies and compare evaluation time of percentage queries. The advantage is that horizontal aggregation reduces the number of rows and columns. Disadvantage is vertical aggregation increase the number of rows and columns. This increases the complexity.

G. Luo and J.F. Naughton (2005) developed the immediate materialized view maintenance with transaction consistency is enforced by generic concurrency control mechanism. A latch pool for aggregate join view is introduced.

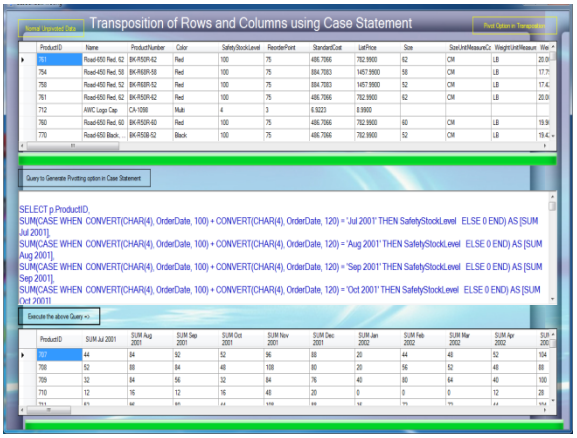
The latches in the latch pool guarantee that for each aggregate group, at most one tuple corresponding to this group exists in the aggregate join view. The main advantage, deadlock problem is solved. The main disadvantage is many join operations are used.

III. Proposed Method

A new class of aggregation function called Horizontal aggregation, represents an extended form of traditional SQL (Structured Query Language) aggregation, which returns set of values in a horizontal layout. Horizontal aggregation is evaluated using three fundamental methods: case, SPJ (Select Project Join) and pivot.

3.1 Case method

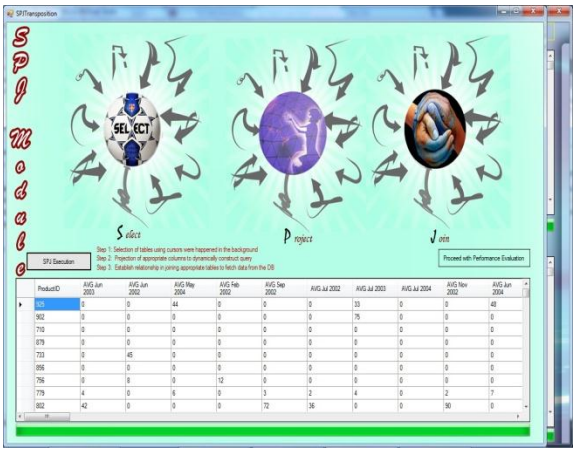
It can be used in any statement or clause that allows a valid expression. The case statement returns a value selected from a set of values based on Boolean expression. The Boolean expression for each case statement has a conjunction of K equality comparisons. Query evaluation needs to combine the desired aggregation with "case" statement for each distinct combination of values of $R_1,\ldots\ldots,R_k$.



3.2 SPJ (Select Project Join) Method

It is based on standard relational algebra operators (SPJ queries). The basic idea is to create one table with a vertical aggregation for each result column, and then join all those tables to produce another table. It is necessary to introduce an additional table F0 that will be outer joined with projected tables to get a complete result set.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.



3.3 Pivot Method

The pivot operator is a built-in operator which transforms row to columns. It internally needs to determine how many columns are needed to store the transposed table and it can be combined with the GROUP BY clause. Since this operator can perform transposition it can help in evaluating horizontal aggregation.

3.4 Performance comparison and evaluation methods

In this method, the performance of SPJ, Case and Pivot method are compared and the efficiency of each and every method is analyzed. Here we are going to compute the No of pre-emptive scheduling process, No of waiting resources, No of input and output operation, CPU and memory usage among case, SPJ and pivot method.

3.5 Knowledge cube generation

Transforming normal data into knowledge cube is one of the emerging fields in the current market. Most of the works are running behind analyzing the data and providing an estimated output. The data will be taken and it will be transformed into knowledge cubes. The data cube is created using the tool SAAS (SQL Server Analysis Services). Microsoft SQL Server Analysis Services is part of Microsoft SQL Server, a database management system. The data will be customized based on "Generalized & Suppression" algorithm. In this algorithm, only the authorized person can view the data. Microsoft SQL Server OLAP Services provides architecture for access to multidimensional data.

This data is summarized, organized and stored in multidimensional structure for rapid response to user queries.For expressing queries to multidimensional data, Microsoft SQL Server OLAP Services employs full-fledged, highly functional expression syntax: MDX (Multi Dimensional eXpression). The MDX expression can be used to view the actual output. In addition to this, the performance efficiency among case, SPJ and pivot methods will be analyzed.

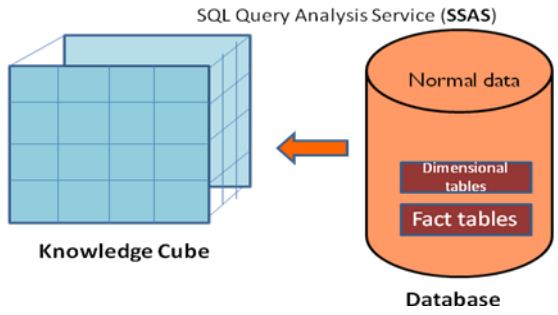


Fig 1: Transforming normal data into knowledge cube.

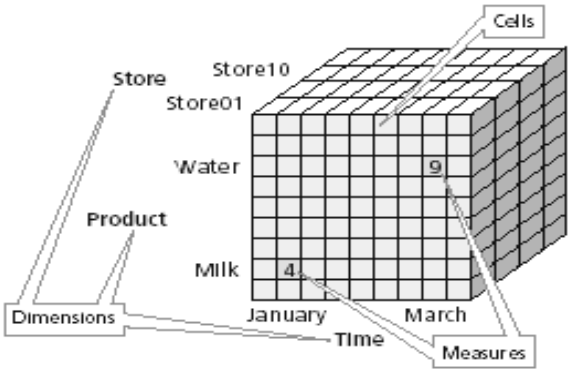


Fig 2: Multidimensional Data Cube

IV. Result And Analysis

Transforming normal data into knowledge cube is one of the emerging fields in the current market. Most of the works are running behind analyzing the data and providing an estimated output. The data will be taken and it will be transformed into knowledge cubes. The data cube is created using the tool SAAS (SQL Server Analysis Services). The Figure 3 shows the Exploring cube data in the cube browser

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

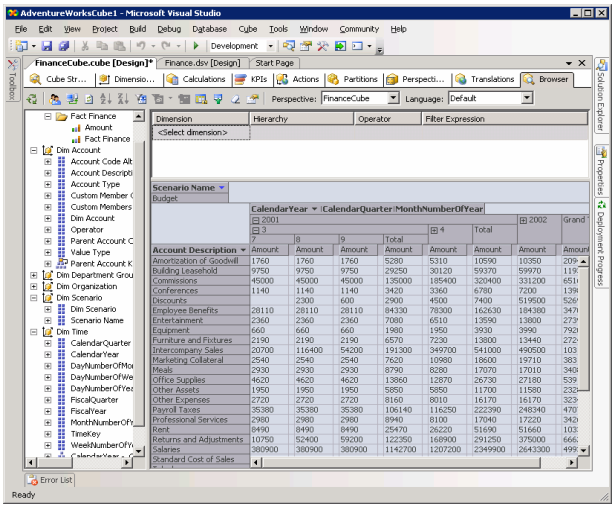


Fig 3: Exploring cube data in the cube browser

V. CONCLUSION AND FUTURE WORK

Multidimensionaling the data and followed by multidimensional cube generation is the scope of the project. Transforming normal data into knowledge cube is one of the emerging field in the current market. Most of the works are running behind analyzing the data and providing an estimated output. The data will be taken and it will be transformed into knowledge cubes. The data cube provides a multidimensional view of data and allows the fast accessing of summarized data.

REFERENCES

[1] C. Cunningham, G. Graefe, and C.A. Galindo-Legeria, "PIVOT AND UNPIVOT: Optimization and Execution Strategies in an RDBMS," Proc: 13th Int'l Conf. Very Large Data Bases (VLDS'04), pp.998-1009, 2004.

[2] C. Galindo-Legaria and A. Rosentahl, "Outer Join Simplification and Reordering for Query Optimization," ACM Trans. Database Systems, vol.22, no.1, pp.43-73, 1997.

[3] G. Graefe, U. Fayyed, and S. Chaudhuri, "On the Efficient Gathering of Sufficient Statistics for Classification from Large SQL Databases, "Proc. ACM Conf. Knowledge Discovery and Data Mining (KDD'98), pp. 204-208, 1998.

[4] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab and Sub-Total, "Proc: Int'l Conf. Data Eng., pp. 152-159, 1996.

[5] J. Han and M. Kamber, Data Mining: Concepts and Techniques, first ed. Morgan Kaufmann, 2001.

[6] G. Luo, J.F. Naughton, C.J. Ellmann, and M. Watzke, "Locking Protocols for Materialized Aggregation Join Views," IEEE Trans. Knowledge and Data Eng., vol. 17, no.6, pp. 796-807, June 2005.

[7] C. Ordonez, "Vertical and Horizontal Percentage Aggregations," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'04), pp. 866-871,2004.

[8] C. Ordonez, "Integrating K-Means Clustering with a Relational DBMS Using SQL, "IEE Trans. Knowledge and Data Eng., vol. 18, no. 2, pp.188-201, Feb. 2006.

[9] C. Ordonez, "Statistical Model Computation with UDFs," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 12, pp. 1752-1765, Dec. 2010.

[10] C. Ordonez, "Data Set Preprocessing and Transformation in a Database System," Intelligent Data Analysis, vol. 15, no. 4, pp. 613-631, 2011.

[11] C. Ordonez and S. Pitchaimalai, "Bayesian Classifiers Programmed in SQL," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 1, pp. 139-144, Jan. 2010.

[12] S. Sarawagi, S.Thomas, and R. Agrawal, "Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications, "Proc ACM SIGMOD Int'l Conf. Mnagement of Data (SIGMOD '98) , pp. 343-354, 1998.