ISSN 2349-9842, Volume 1, Issue 1, March 2014

International Journal of Innovative Trends and Emerging Technologies

AGGREGATION ON HORIZONTAL DATASETS IN KNOWLEDGE CUBE BY MULTIDIMENSIONAL WAY USING SSAS

ANUPREENA.P.S
E-mail : anu1310@ymail.com

Abstract— *P*rojecting data in different dimensions is the core concept taken for this project. The data will be dimensionalised by three different concepts like, CASE: Exploiting the programming CASE construct; SPJ: Based on standard relational algebra operators (SPJ queries); PIVOT: Using the PIVOT operator, which is offered by some DBMSs. Existing SQL aggregations have limitations to prepare data sets because they return one column per Aggregated group. In common, a important manual effort is required to build data set, where a parallel layout is necessary. There is a simple, yet powerful, technique to engender SQL code to arrival cumulative columns in a parallel tabular outline, frequent a set of statistics in its place of one integer per row. This new class is defined as horizontal aggregations. We propose three elementary methods to appraise horizontal aggregations.

Along with the existing approaches, we have included another variety of area in data mining that is extracting data from Knowledge cubes. This can be achieved with the tool SQL Server Analysis Services. The data will be taken and it will be transformed into knowledge cubes. This can be achieved with Multi Dimensional queries. In addition to that, this project introduces a performance evaluation on three methods CASE, SPJ and PIVOT. Experiments with large tables compare the performance of proposed method with existing three methods. Extracting the data from knowledge cube by multi dimensional query is much more efficient and time complexity is less comparing the three methods.

Key words: Aggregation, Pivoting, SQL, SSAS

1.INTRODUCTION

Data mining is an interdisciplinary subfield of computer science. It loosely refers to the process of semi-automatically analyzing the large data base to find useful patterns. It is the computational progression of discovering patterns in large data sets connecting methods at the meeting point. Like knowledge discovery in artificial intelligence or statistical analysis, datamining attempts to discover rules and patterns from data. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for additional use. Some type of knowledge discovered from a database can be represented as set of rules.

1.1 Motivation:

OLAP tools generate SQL code to transpose results sometimes called PIVOT. OLAP transposition can be more proficient if there are methods combining aggregation and transposition together. Overcome the SQL aggregation, a new class of aggregate functions is proposed that the aggregation of numeric terms and transposition of data set into a horizontal tabular layout. Functions of this class are defined to be horizontal aggregations. It represents an extended form of traditional SQL aggregations because SQL aggregation has certain limitations. Horizontal Aggregation returns a set of values, whereas SQL aggregation returns single value per row. This paper explains how to evaluate and optimize horizontal aggregations generating standard SQL code and this can be done by three methods. They are CASE, PIVOT and SPJ. We evaluate the performance of these three methods.

Transforming Normal data into Knowledge cubes 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. In this project we are trying to PIVOT the valuable data into a different transformation. We are trying to focus the possible area of SPJ (Select Project Join), Case and Pivot methods. Along with the existing approaches, we are trying to incorporate a new standard of Pivoting option using Data mining. Horizontal improvisation of aggregated items will be achieved using the Pivot operator and generic aggregated methods can be achieved with the help of MDX querying concept. The knowledge data will be customized based on Generalized and Suppression Algorithm. On top of this, we will be analyzing the possibility of performance efficiency among these methods.

Our proposed horizontal aggregations has some features, the data set can be created entirely inside the Knowledge cube by fact and dimension table. Fortunately, this can proposed with the help of SSAS. Therefore, we provide a more efficient and more secured aggregation when compared to external aggregation Methods. Multidimensional aggregation is more compact and efficient. This project is the option of Horizontal improvisation of Aggregated items will be achieved using the Pivot operator and generic aggregated methods can be achieved with the help of MDX querying concept.

1.2 Related Works

Horizontal aggregation reduces the number of rows and columns. Existing SQL aggregate functions present important limitations to compute percentages [1]. Percentage evaluation has wide applicability and can be efficiently evaluated. **C.Ordonez [1]** introduced a novel aggregation performing two tasks. For each percentage the first task returns one row in vertical form like standard SQL aggregations. For each percentage the second task returns adding 100% on the same row in horizontal form. These tasks are used in a framework to introduce the concept of percentage queries and to generate efficient SQL code. Experimental study on different percentage query optimization strategies and comparing evaluation time of percentage queries. Disadvantage is vertical aggregation increase the number of rows and columns. This increases the complexity.

A latch pool for aggregate join view is introduced. The latches in the latch pool certify that for each aggregate group, at most single tuple subsequent to this group be presents in the aggregate join view.[1] The main advantage, deadlock problem is solved. **G. Luo and J.F.Naughton [2]** developed the immediate materialized view maintenance with transaction consistency is enforced by generic concurrency control mechanism. The main disadvantage is many join operations are used. PIVOT and UNPIVOT are two operators on tabular data that exchange rows and columns, enable data transformation useful in data modelling, data analysis and data presentation.[3] Without difficulty they can be applied inside a query processor, like select, project and join. Above design provides a chance for enhanced performance, in cooperation of query optimization and query execution.

C. Cunningham [3] developed these two operators: Pivot and Unpivot. Paper [3], 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.

2 EXISTING SYSTEM

Existing SQL aggregations have limitations to prepare data sets because they return one column per aggregated group. It cannot handle complicated relationship between features. Data set for analysis is generally the most time consuming task in a data mining project, need many composite SQL queries for joining multiple tables and aggregating the columns. PIVOT operator can transpose by only one column. No system focuses in the Data Mining area with involvement of Knowledge Cubes. Performance evaluation on the common data with different scenarios was not evaluated properly in the existing system.

Issues of Existing System

- Sequential queries are not possible in this system.
- PIVOT operator can transpose by only one column in existing system.
- PIVOT operator requires eliminating unwanted columns (trimming) from the input table for efficient evaluation.
- One column per aggregation only achieved in this system

3 PROPOSED SYSTEM

Multidimensional data and followed by multidimensional cube generation is the scope of the project. Once the cube is generated, the option of pivoting the data of transferring the data from row to column and columns to rows is achieved. In addition, the option of Horizontal improvisation of Aggregated items will be achieved using the Pivot operator and generic aggregated methods can be achieved with the help of MDX querying concept. Performance is evaluated on the common data with the existing and proposed techniques for data mining analysis. Multicolumn in single aggregation can possible in this system. It is a Multidimensional, fastest and secured system. Comparing the performance with existing three techniques extracting the data from knowledge cube by multi dimensional query is much more efficient. It is evaluated based upon the CPU utilization, Compilation plan, execution plan and maximum work done.

ADVANTAGES

- Aggregated items will be achieved by MDX queries from the knowledge cube
- Less time consuming
- Extracting the data from knowledge cube by multi dimensional query is much more efficient and yield high performance.

4 ARCHITECTURAL DIAGRAM

The data will be dimensionalised and horizontal aggregations is performed by three different concepts like, CASE: Exploiting the programming CASE construct; SPJ: Based on standard relational algebra operators (SPJ queries); PIVOT: Using the PIVOT operator, which is a transposition of row and columns in DBMSs.

Along with the existing approaches, another variety of area in data mining that is extracting data from Knowledge cubes. The tool “SQL Server Analysis Services” is an application of data base can help in

ISSN 2349-9842, Volume 1, Issue 1, March 2014

International Journal of Innovative Trends and Emerging Technologies

constructing the knowledge cube and MDX query. BY using the facts and dimension table data set are prepared and it will be transformed into knowledge cubes. From this mutidimensional aggregation can be achieved with Multi Dimensional queries. In addition to that, project introduces a performance evaluation on three methods CASE, SPJ and PIVOT.

The performance of case, Pivot and SPJ techniques are evaluated and compared with performance of MDX query check. Evaluating various factors like CPU utilization, Compilation Plan, Execution plan and maximum work done. This analysis and evaluation depends upon time optimization and resource utilization like files, network connections and memory areas. After analysis and performance evaluation, knowledge cube yields more optimization and it is more efficient than the other three techniques.Extracting the data from knowledge cube by multi dimensional query is much more efficient and time complexity is less comparing the three methods.

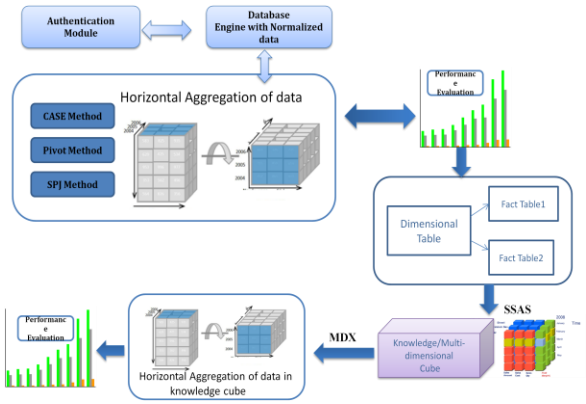


Fig 1 System Architecture

5 MODULES DESCRIPITION:

A. CASE STATE

The case statement returns a value selected from a set of values based on Boolean expressions. CASE can be used in any statement or clause that allows a valid expression. From a relational database this is equivalent to doing a simple projection/aggregation query where each non key value is given by a function.

B. SPJ CHECK

The basic idea is to create one table with a vertical aggregation for each result column, and then join each and every tables to create another table. It is based on standard relational algebra operators (SPJ queries).

C. PIVOT CHECK

The PIVOT method internally needs to determine how many columns are needed to store the transposed table and it can be merged with the GROUP BY clause. The vital syntax to utilize in the PIVOT operator to compute a horizontal aggregation assuming one BY column for the right key columns.

D. KNOWLEDGE CUBE GENERATION

In this module, we are going to create knowledge cubes from normal data by using SQL Server Analysis service. Microsoft SQL Server Analysis Services is component of Microsoft SQL Server, a database management system. These knowledge cubes have various aspects to retrieve data fast.

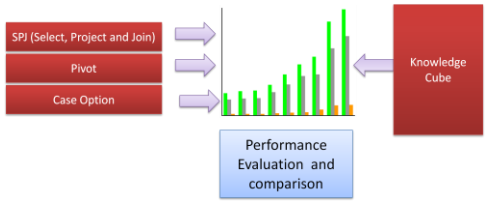
- Configure Data source
- Configure Dimensions
- Configure Cubes

E. MDX QUERYCHECK

By using Multidimensional query we can access Cube instead of table. 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.

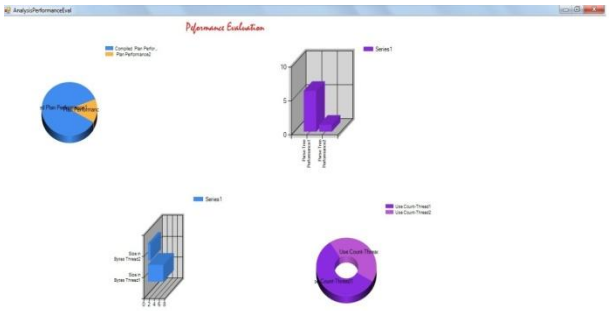
F. PERFORMANCE COMPARISON AND EVALUATION

In this module, we are going to compare the performance of SPJ, CASE and Pivot method and going to find the efficiency of each and every method. 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. Performance of knowledge cube is evaluated based upon the CPU utilization, Compilation plan, execution plan and maximum work done and compared with the performance with the existing system. Horizontal aggregation with multidimensional query is more efficient and the time complexity is decreased.



Based upon the CPU utilization, Compilation plan, execution plan and maximum work done in multidimentional aggregation the time complexity is decreased. Experiments with large tables compare the performance of multidimensional aggregation method with existing Horizontal aggregation methods. The sample results is shown below.

Process	MDX QUERY(%)	CASE, PIVOT,SPJ(%)
Cpu utilization	80	20
Compilation plan	6	2
Execution plan	4	1
Maximum work done	60	40



6 CONCLUSION

In this project, I have described the new class of extended aggregate functions, called pivoted aggregations which help preparing data sets for data mining and OLAP cube exploration. In addition, the option of Horizontal improvisation of Aggregated items will be achieved using the Pivot operator and generic aggregated methods can be achieved with the help of MDX querying concept. Basically, a horizontal aggregation returns a set of numbers instead of a single number for each group, resembling a multi-dimensional vector. The performance of case, Pivot and SPJ techniques are evaluated and compared with performance of MDX query check. I am evaluating various factors like CPU utilization, Compilation Plan, Execution plan and maximum work done. This analysis and evaluation depends upon time optimization and resource utilization like files, network connections and memory areas. After analysis and performance evaluation, knowledge cube yields more optimization and it is more efficient than the other three techniques.

7 FUTURE ENHANCEMENT:

Horizontal aggregations create tables with smaller number of rows, but with maximum number of columns. Thus query optimization methods intended for standard (vertical) aggregations are unfortunate for horizontal aggregations. I plan to develop more entire I/O cost models for cost-based query optimization and want to study optimization of horizontal aggregations method in parallel in a shared-nothing DBMS architecture. Cube properties can be simplifyd to multi-valued aggregation results produced by a horizontal aggregation. Optimizing a workload of horizontal aggregation queries is another challenging problem.

REFERENCES:

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

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

[3] 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.

[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] C. Ordonez, “Horizontal Aggregations for Building Tabular Data Sets,” Proc. Ninth ACM SIGMOD Workshop Data Mining and Knowledge Discovery (DMKD ’04), pp. 35-42, 2004.

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

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

[9] H. Wang, C. Zaniolo, and C.R. Luo, “ATLAS: A Small But Complete SQL Extension for Data Mining and Data Streams,” Proc. 29th Int’l Conf. Very Large Data Bases (VLDB ’03), pp. 1113- 1116, 2003.

[10]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.

[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.

.