

(An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2014

Design and Implementation of Multi Dimensionalized Aggregation in Horizontal Dataset Using Data Analysis Services

R.Suguna, S.Aiswarya, S.Ramadevi

Associate Professor, Department of CSE, Arunai College of Engineering, Tiruvannamalai, Tamilnadu, India Assistant Professor, Department of CSE, Velammal Institute of Technology, Chennai, India Assistant Professor, Department of CSE, Arunai College of Engineering, Tiruvannamalai. Tamilnadu, India

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

Copyright to IJIRCCE

10.15680/ijircce.2014.0212055 www.ijircce.com



(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

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. 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, R_k .



(An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2014

	Productill	Name	EnductNumber	Cár	Safety/Stavial evel	Banatian/Prant	9anderCod	LigPane	Sie	Stel httMana	eCr. Westeilstille	and W
	70	Road-650 Red. 62	BK-8508-62	Fed	100	75	436,7365	782,9900	62	CH	18	20
1	754	Road-450 Red, 58	BK-REER-SE	Red	100	75	884 7083	1457,9900	58	CM	LB	17
	758	Road-450 Red, 52	BK-REER-52	Red	100	75	884.7083	1457.9900	52	CN	LB	17
	761	Road-650 Red, 62	BK-850R-62	Red	100	75	486.7066	782,9900	62	CH	LB	2
	712	AWC Logo Cap	CA-1056	Nuti	4	3	6.9223	8.9900				
	760	Road-650 Red, 60	BK-RSOR-60	Red	100	75	486.7066	782,9900	60	CH	LB	19
	770	Road-451 Black	BK-9508-52	Back	100	75	416 7566	782 9960	52	CH	1.9	10
LE M(20 M(22	y to Generate Rv CCT p. Produ (CASE WHI 2011) (CASE WHI 2001)	otting option in Case Sta IctID, EN CONVERT(C EN CONVERT(C	CHAR(4), Ord	erDate, 100; erDate, 100;	+ CONVERT(C) + CONVERT(C)	IAR(4), Orde	rDate, 120) = ' rDate, 120) = '	Jul 2001' THE Aug 2001' TH	EN SafetyStoc	kLovol ELSE IckLovel ELS	0 END) AS (S E 0 END) AS [UM
LE M(20) M(20) M(20 M(20) M(20) M(M(20) M(M(20) M(M(M()) M(M(M(M(M(M(M(M(M(M(M(M(M(y to Generate Riv (CASE WHI 001], (CASE WHI 0001], (CASE WHI 2001], (CASE WHI 2001], (CASE WHI 0011 0011	uting option in Case Sta IceliD, ENI CONVERT(0 ENI CONVERT(0 ENI CONVERT(0 ENI CONVERT(0	conert CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord	erDate, 100) erDate, 100) erDate, 100) erDate, 100)	+ CONVERT(C) + CONVERT(C) + CONVERT(C) + CONVERT(C)	IAR(4), Orde IAR(4), Orde IAR(4), Orde IAR(4), Orde	rDate, 120) = 1 rDate, 120) = 1 rDate, 120) = 1 rDate, 120) = 1	Jul 2001' THE Aug 2001' Th Sep 2001' Th Oct 2001' Th	EN SafetyStoc IEN SafetySto IEN SafetySto EN SafetySto	klovel ELSE Icklovel ELS Icklovel ELS Icklovel ELSE	0 END) AS [S E 0 END) AS [E 0 END) AS [E 0 END) AS [UM SUM SUM
LE M(20 M(y to Generate Riv (CASE WHI 001) (CASE WHI 2001), (CASE WHI 2001), (CASE WHI 2001), (CASE WHI 0011 0011 (CASE WHI 0011 (CASE WHI 0011 (CASE WHI 0011 (CASE WHI 0011 (CASE WHI 0011) (CASE WHI (CASE WHI 0011) (CASE WHI (CASE	eting aption in Case Str IetHD, ENI CONVERT(C ENI CONVERT(C ENI CONVERT(C ENI CONVERT(C INV ->	ERNERT CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord	erDate, 100; erDate, 100; erDate, 100; erDate, 100; erDate, 100; sum Sep 200	+ CONVERT(C) + CONVERT(C) + CONVERT(C) + CONVERT(C) + CONVERT(C)	IAR(4), Orde IAR(4), Orde IAR(4), Orde IAR(4), Orde	iDate, 120) = * iDate, 120) = * iDate, 120) = * iDate, 120) = * iDate, 120) = *	Jul 2001' TH Aug 2001' TH Sep 2001' Th Oct 2001' Th Oct 2001' Th	EN SafetyStoc KEN SafetySto KEN SafetySto EN SafetySto Syll Fee	klovel ELSE cklovel ELS cklovel ELS cklovel ELSE SJUI Ner 202	0 END) AS (S E 0 END) AS (E 0 END) AS (E 0 END) AS (E 0 END) AS (Stan Awr	
LE M(20 M(2) M(2) M(2) M(2) M(2) M(2) M(2) M(2)	y to Generate Riv CCT p Produ (CASE WHI 001], (CASE WHI 00], (CASE WHI 0)	etting uption in Case Sta ettiD, EN CONVERT(IC EN CONVERT(IC E	ement CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord	erDate, 100 erDate, 100 erDate, 100 erDate, 100 som Sep 2001	+ CONVERT(C) + CONVERT(C) + CONVERT(C) + CONVERT(C) + CONVERT(C) 32	IAR(4), Orde IAR(4), Orde IAR(4), Orde IAR(4), Orde	iDate, 120) = ' iDate, 120) = ' iDate, 120) = ' iDate, 120) = ' iDate, 120) = '	Jul 2001' THE Aug 2001' Th Sep 2001' Th Oct 2001' TH Oct 2001' TH	EN SafetySloc IEN SafetySlo IEN SafetySlo EN SafetySto EN SafetySto	kLovel ELSE kokLovel ELS kokLovel ELS kokLovel ELSE S2012 2002	0 END) AS [S E 0 END) AS [E 0 END) AS [E 0 END) AS [E 0 END) AS [State of the state of the s	UM SUM SUM SUM
LE M(20 M(2) M(2) M(2) M(2) M(2) M(2) M(2) M(2)	y to Generate Riv CCT p. Phodual (CASE WHE 2001), (CASE WHE 2001), (CASE WHE 2001), (CASE WHE 2001), (CASE WHE 2001), (CASE WHE 2001), CASE WHE 2001), 2001 Roduet20	Intergraphics in Date Sta Interface Converting EN	Esment CHAR(4), Ord CHAR(4), Or	erDate, 100 erDate, 100 erDate, 100 erDate, 100 sum Sep 2001 sum Sep 2001 sum Sep 2001 sum Sep 2001 sum Sep 2001 sum Sep	+ CONVERT(C) + CONVERT(C) + CONVERT(C) + CONVERT(C) + CONVERT(C) 2001 52 52 52 52 52 52 52 52 52 52 52 52 52	IAR(4), Orde IAR(4), Orde IAR(4), Orde IAR(4), Orde IAR(4), Orde	rDate, 120) = ' rDate, 120) = ' sum Dec 2011 88 80	Jul 2001' THE Aug 2001' Th Sep 2001' Th Oct 2001' TH 2002 20 20 20 20	EN SafetyStoc IEN SafetySto IEN SafetySto EN SafetySto StafetySto 2002 44 56	kLovel ELSE ckLovel ELS ckLovel ELS ckLovel ELSE State 44 52	0 END) AS [S E 0 END) AS [E 0 END) AS [E 0 END) AS [E 0 END) AS [State of the state of the s	SUM SUM SUM SUM
LE M(20 M(y to Generate Riv CCT p. Phodu (CASE WHE 2001), (CASE WHE	etting spion in Cave 32 cellD, EN CONVERTI(EN CONVERTI(EN CONVERTI(EN CONVERTI(EN CONVERTI(SUM JA 2001 44 52 32	Ement CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord CHAR(4), Ord State S	erDate, 100 erDate, 100 erDate, 100 erDate, 100 erDate, 100 82 82 84 84 84	+ CONVERT(CF + CONVERT(CF + CONVERT(CF + CONVERT(CF + CONVERT(CF 2007 52 53 52 53 54 54 52 53 52 53 52 53 52 53 52 53 52 53 53 55 55 55 55 55 55 55 55 55 55 55	4AR(4), Orde 4AR(4), Orde 4AR(4), Orde 4AR(4), Orde 54 56 56 56 56 56	rDate, 120) = ' rDate,	Jul 2001' THE Aug 2001' TH Sep 2001' TH Oct 2001' TH 2002 20 20 40	EN SafetyStoc IEN SafetySto IEN SafetySto EN SafetySto 2002 44 44 56 10	kLovel ELSE ckLovel ELS ckLovel ELS ckLovel ELSE Striker 44 45 56 64	0 END) AS (S E 0 END) AS (E 0 END) AS (E 0 END) AS (E 0 END) AS (S E 0 END) AS (S 0 END	SUM SUM SUM SUM SUM

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.



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

Copyright to IJIRCCE

10.15680/ijircce.2014.0212055 www.ijircce.com



(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

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 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



(An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2014

					÷				
FinanceCube.cube [Design	 Finance.dsv [Design] 	Start Pag	je						+ ×
🔍 Cube Str 💇 Dimensio	😭 Calculations 🥃	KPIs 🛃	Actions	Partition:	s 👩 Persp	ecti	Translation	s 🔍 Brow	vser
종 🔊 이 히 김 🛪	18 - 18 1. 3 /	Pers	pective: Fin	nanceCube	▼ Land	uage: Defa	ult	-	
Eart Enance									
- Amount	Dimension	Hierarchy		Oper	ator	Hiter Expre	ssion		
Fact Finance	Select dimension>								
The fact minance									
E Dim Account									
Account Code Alt									
E Account Descripti									
📧 🚦 Account Type	Scenario Name 🔻								
표 🚦 Custom Member 🤇	Budget								
📧 🚦 Custom Members		Calendar	Year VICa	lendarOua	rter Month	NumberOf	/ear		
🗉 🧮 Dim Account		E 2001						ER 2002	Grand
🗉 🗮 Operator		E3				FE 4	Total		Grand
E Parent Account C		7	8	9	Total		10000		
Value Type	Account Description *	Amount	Amount	Amount	Amount	Amount	Amount	Amount	Amou
	Amortization of Goodwill	1760	1760	1760	5280	5310	10590	10350	209
Parent Account K	Building Leasehold	9750	9750	9750	29250	30120	59370	59970	119
🗄 🔝 Dim Department Grou	Commissions	45000	45000	45000	135000	185400	320400	331200	651
Dim Organization	Conferences	1140	1140	1140	3420	3360	6780	7200	139
🖃 🙋 Dim Scenario	Discounts		2300	600	2900	4500	7400	519500	526
Dim Scenario	Employee Benefits	28110	28110	28110	84330	78300	162630	184380	3471
🗉 🧮 Scenario Name	Entertainment	2360	2360	2360	7080	6510	13590	13800	273
🖃 🕼 Dim Time	Equipment	660	660	660	1980	1950	3930	3990	792
E CalendarOuarter	Furniture and Fixtures	2190	2190	2190	6570	7230	13800	13440	272
Calendar Quarter	Intercompany Sales	20700	116400	54200	191300	349700	541000	490500	103
Calendar vear	Marketing Collateral	2540	2540	2540	7620	10980	18600	19710	383
E DayNumberOfMoi	Meals	2930	2930	2930	8790	8280	17070	17010	340
🕑 🧱 DayNumberOfWe	Office Supplies	4620	4620	4620	13860	12870	26730	27180	539
표 🚦 DayNumberOfYea	Other Assets	1950	1950	1950	5850	5850	11700	11580	232-
🖅 🧮 FiscalQuarter	Other Expenses	2720	2720	2720	8160	8010	16170	16170	323
🕫 🚦 FiscalYear	Payroll Taxes	35380	35380	35380	106140	116250	222390	248340	470
Monthly mber Of y	Professional Services	2980	2980	2980	8940	8100	17040	17220	342
TimeKey	Rent	8490	8490	8490	25470	26220	51690	51660	103
The shitten har Off	Returns and Adjustments	10750	52400	59200	122350	168900	291250	375000	666;
+ weekNumberOfy	Salaries	380900	380900	380900	1142700	1207200	2349900	2643300	499.

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

- 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.
- C. Galindo-Legaria and A. Rosentahl, "Outer Join Simplification and 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.