ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

Customized SQL Aggregations for Generating Data Sets

Suneetha.M[1], Kiran Kumar.G[2]

Student, Department of CSE, MLRIT, Hyderabad, India [1]

HOD, Department of CSE, MLRIT, Hyderabad, India [2]

Abstract: Data mining is the domain which has utility in real world applications. Data sets are prepared from regular transactional databases for the purpose of data mining. However, preparing datasets manually is time consuming and tedious in nature as it involves aggregations, sub queries and joins. Moreover the traditional SQL aggregations such as MAX, MIN etc. can generate single row output which is not useful in generating datasets. Therefore it is essential to build horizontal aggregations that can generate datasets in horizontal layout. These data sets can be used further for data mining in the real world applications. This paper focuses on building user-defined horizontal aggregations such as PIVOT, SPJ and CASE whose underlying logic uses SQL queries. We built a prototype that demonstrates the efficiency of the proposed horizontal aggregations. The empirical results revealed that the prototype is effective and can be used in real world applications.

Index Terms: SQL, aggregations, horizontal aggregations

I. INTRODUCTION

Database model such as RDBMS has been used widely for storing and retrieving real world business data. Though the databases support mechanism for data storage and retrieval, they are used in data to day operations. For making well informed business decisions, it is essential to mine such data to extract trends or patterns from the data. However, transactional database cannot be used directly for data mining. Therefore preparing datasets for data mining purposes assumes significance. However, the existing aggregation functions available in SQL do not support to generate datasets as they can only produce single row outputs. The summary of business data can be given for data mining purposes instead of giving the whole business data. This is the idea behind preparing datasets for data mining. As the vertical aggregations fail to deliver goods, it is essential to have horizontal aggregations. However, SQL does not support them [1]. But the vertical aggregations are useful in statistical algorithms [2], [3]. As data mining requirements expect data set to have horizontal layout (set of rows and columns), it is important to generated datasets with that layout. There are data mining techniques like clustering, classification, regression, PCA and so on [4].

Horizontal aggregations proposed in this paper include PIVOT, SPJ and CASE. Actually these are new programming constructs that can produce output with horizontal layout which is further used in data mining operations. These horizontal aggregations are made using the underlying SQL commands and other logic required. The data mining operations generally come under OLAP (Online Analytical Processing) as it acts on historical data rather than regular data. We built a web based prototype to demonstrate the proposed horizontal aggregations. The proposed horizontal aggregations work faster as they are made pre-compiled objects. This will save time as the horizontal aggregations are pre-compiled objects. They are executed faster when compared to normal SQL queries. The application is user-friendly with provision for generating data sets for data mining. The remainder of this paper is organized as follows. Section II reviews literature. Section III describes horizontal aggregations. Section IV presents experimental results while section V concludes the paper.

II. RELATED WORK

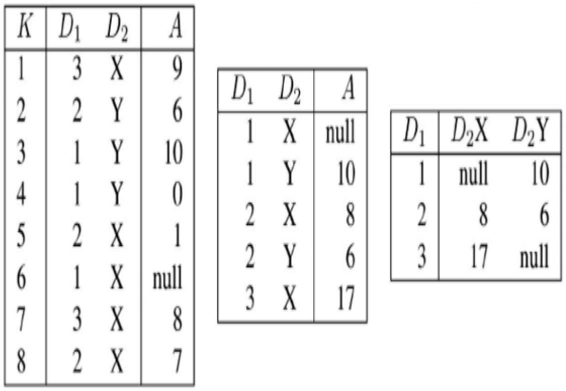
SQL is the de facto standard to interact with relational databases. It is widely used in all kinds of applications where connectivity to database containing valuable business data is required. SQL provides commands of various categories such as DML, DDL, TCL and DCL. Using SELECT query it is possible to use aggregations, sub queries and joins. The vertical aggregations supported by SQL include COUNT, MIN, AVG, MAX and SUM. These are known as aggregate functions as they produce summary of data [5]. The output of these functions is in the form of single row values. These values can't be directly used for data mining. Therefore it is essential to use some data mining procedures in order to generate data sets. Association rule mining [6] is used in OLAP applications as they can generate trends in the data [7]. In this paper we extend the SQL aggregate functions in order to build new constructs namely PIVOT, SPJ and CASE. SQL queries are used in clustering algorithms also as explored in [5]. Spreadsheet like operations as extensions to SQL queries are proposed in [8]. The paper also discussed optimizations for joins and other operations. However, it is known that CASE and PIVOT can be used to avoid joins. New class of aggregations can be generated by using algebra that has been used traditionally [9]. In fact this paper focuses on generating new class of aggregations known as horizontal aggregations which will optimize the joins as presented in [10].

For optimizing queries tree-based plans are used traditionally [11]. On aggregations also there is lot of research found in the literature. Literature also includes cube queries and cross tabulations [12]. Relational tables can unpivoted as presented din [13]. Transformations are available that can be used for horizontal aggregations [14]. Unpivot and TRANSPOSE operators are similar. When compared with PIVOT transpose can reduce the number of operations required. They have inverse relationship between them. They can produce vertical aggregations and decisions tree required by data mining. Both operations are available in SQL Server [15].

Horizontal aggregations are also presented by researchers in [16] and [17] with known limitations. The limitation is that the resultant data cannot be directly used for data mining. In this paper we proposed new operators that are best used for horizontal aggregations. The results of these operations can be used for data mining purposes further. The proposed operations include SPJ, PIVOT and CASE.

III. HORIZONTAL AGGREGATIONS

Horizontal aggregations are the operations that perform horizontal summary of data in tabular format or horizontal layout. The base tables used to describe the proof of concept are presented in fig. 1. These tables are used for describing operations of SPJ, PIVOT and CASE.



K	D_1	D_2	A
1	3	X	9
2	2	Y	6
3	1	Y	10
4	1	Y	0
5	2	X	1
6	1	X	null
7	3	X	8
8	2	X	7

D_1	D_2	A
1	X	null
1	Y	10
2	X	8
2	Y	6
3	X	17

D_1	D_2X	D_2Y
1	null	10
2	8	6
3	17	null

Fig. 1 – Input table (a), traditional vertical aggregation (b), and horizontal aggregation (c)

As seen in fig. 1, sample data is given in input table. Vertical aggregation result is presented in (b). In fact the result generated by SUM function of SQL is presented in (b). Horizontal aggregation results are presented in (c).

Steps Used in All Methods

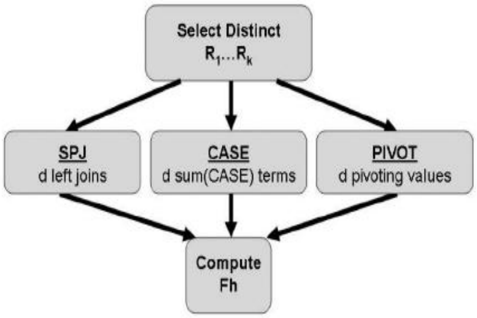


Fig. 2 shows steps on all methods based on input table

As seen in fig. 2, for all aggregations such as PIVOT, CASE and SPJ certain steps are carried out. However, the first step of all operations starts with SELECT query. Then based on the operation other activities are performed for computing horizontal aggregations.

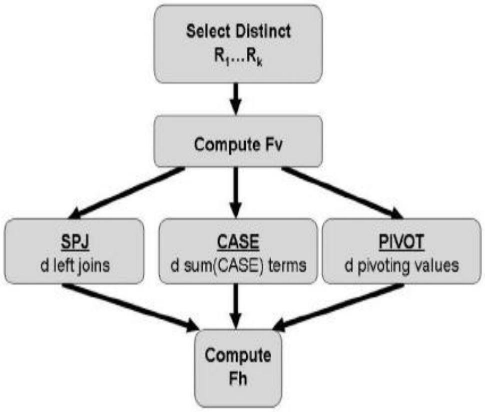


Fig. 3 shows steps on all methods based on table containing results of vertical aggregations

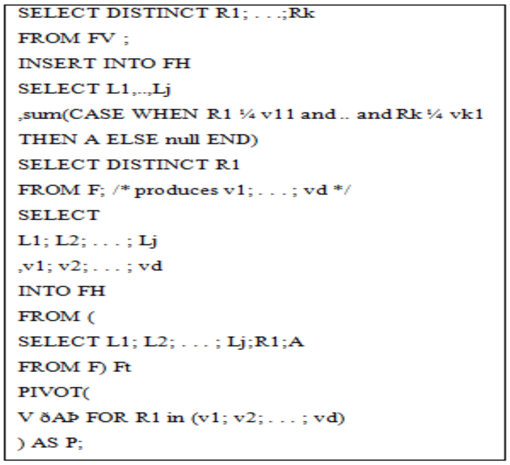
As seen in fig. 2, for all aggregations such as PIVOT, CASE and SPJ certain steps are carried out. However, the first step of all operations starts with SELECT query. Then based on the operation other activities are performed for computing horizontal aggregations.

SPJ Method

Vertical operations are used in SPJ method. For every column one table is generated in this model. Afterwards, the tables generated are joined in order to obtain final horizontal aggregations. The procedure followed is as given in [18].

PIVOT Method

RDBMS has built in PIVOT operation. This is used by the PIVOT operation we proposed in this paper. This construct can provide transpositions. Therefore for evaluating horizontal aggregations it can be used.



```
SELECT DISTINCT R1; . . .;Rk
FROM FV ;
INSERT INTO FH
SELECT L1,..,Lj
,sum(CASE WHEN R1 ¼ v11 and .. and Rk ¼ vk1
THEN A ELSE null END)
SELECT DISTINCT R1
FROM F; /* produces v1; . . . ; vd */
SELECT
L1; L2; . . . ; Lj
,v1; v2; . . . ; vd
INTO FH
FROM (
SELECT L1; L2; . . . ; Lj;R1;A
FROM F) Ft
PIVOT(
V ðAÞ FOR R1 in (v1; v2; . . . ; vd)
) AS P;
```

Listing 1 – Shows optimized instructions for PIVOT construct

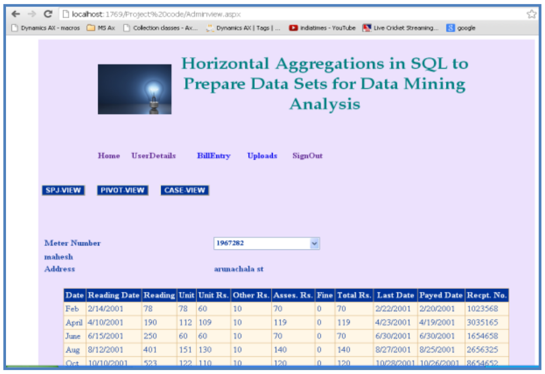
As seen in listing 1, the queries have been optimized by choosing only the columns that are required by horizontal aggregations.

CASE Method

This operation is based on the CASE structure provided by SQL. It has many built in Boolean expressions. Out of them one of the expressions is returned. Projection or aggregation is similar to this from relational query point of view. It is achieved internally by using many conditions with conjunctions. In this case horizontal aggregations exhibit two strategies. The first one does the computations directly from the table given as input while the second one performs vertical aggregation and the results are sent to an arbitrary table. This table is used again in horizontal aggregation generation. The procedure used here is as presented in [18].

IV. EXPERIMENTAL EVALUATION

The prototype is web based application which is built using the environment containing a PC with 4 GB RAM, core 2 dual processor running Windows 7 operating system. The application is built using Microsoft .NET platform. ASP.NET is used for designing application while SQL server is used as backend. For rich user experience AJAX is used. Coding is done using C#. Figure 4 shows SPJ results.



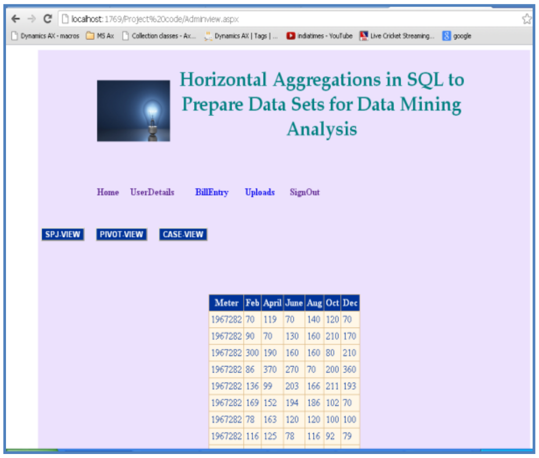
Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis

Meter Number 1967282
mahesh
Address arunachala st

Date	Reading Date	Reading	Unit	Unit Rs.	Other Rs.	Asses. Rs.	Fine	Total Rs.	Last Date	Payed Date	Recpt. No.
Feb	2/14/2001	78	78	60	10	70	0	70	2/22/2001	2/20/2001	1023568
April	4/10/2001	190	112	109	10	119	0	119	4/23/2001	4/19/2001	3035165
June	6/15/2001	250	60	60	10	70	0	70	6/30/2001	6/30/2001	1654658
Aug	8/12/2001	401	151	130	10	140	0	140	8/27/2001	8/25/2001	2656325
Oct	10/10/2001	523	122	110	10	120	0	120	10/28/2001	10/26/2001	8654652

Fig. 4 – Results of SJP aggregation

As seen in fig. 4, SPJ operation's results are presented in horizontal layout. This kind of data can be used further for data mining operations.



Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis

Meter	Feb	April	June	Aug	Oct	Dec
1967282	70	119	70	140	120	70
1967282	90	70	130	160	210	170
1967282	300	190	160	160	80	210
1967282	86	370	270	70	200	360
1967282	136	99	203	166	211	193
1967282	169	152	194	186	102	70
1967282	78	163	120	120	100	100
1967282	116	125	78	116	92	79

Fig. 5 – Result of Pivoting Aggregation

As seen in fig. 5, PIVOT operation’s results are presented in horizontal layout. This kind of data can be used further for data mining operations.

Fig. 6 – Result of CASE Aggregation

As seen in fig. 6, CASE operation’s results are presented in horizontal layout. This kind of data can be used further for data mining operations.

V. CONCLUSIONS

In this paper we built new class of aggregations known as horizontal aggregations. They aggregate operators we proposed and built include PIVOT, CASE and SPJ. The operators produce horizontal aggregations resulting data in tabular format or horizontal layout. This kind of data which is suitable for data mining operations is known as horizontal aggregations. They are used in OLAP applications. As vertical aggregations such as MIN, MAX, SUM, COUNT and AVG can’t produce horizontal layout, it is essential to have custom-built constructs for horizontal aggregations. We built a web application that demonstrates the proof of concept. The empirical results revealed that the prototype is effective and help in producing horizontal aggregations.

REFERENCES

[1] C. Ordonez, “Data Set Preprocessing and Transformation in a Database System,” Intelligent Data Analysis, vol. 15, no. 4, pp. 613- 631, 2011.
[2] 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.
[3] C. Ordonez, “Statistical Model Computation with UDFs,” IEEE Trans. Knowledge and Data Eng., vol. 22, no. 12, pp. 1752-1765, Dec. 2010.
[4] J. Han and M. Kamber, Data Mining: Concepts and Techniques, first ed. Morgan Kaufmann, 2001.
[5] 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.
[6] 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.
[7] 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.
[8] A. Witkowski, S. Bellamkonda, T. Bozkaya, G. Dorman, N. Folkert, A. Gupta, L. Sheng, and S. Subramanian, “Spreadsheets in RDBMS for OLAP,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’03), pp. 52-63, 2003.
[9] H. Garcia-Molina, J.D. Ullman, and J. Widom, Database Systems: The Complete Book, first ed. Prentice Hall, 2001.
[10] C. Galindo-Legaria and A. Rosenthal, “Outer Join Simplification and Reordering for Query Optimization,” ACM Trans. Database Systems, vol. 22, no. 1, pp. 43-73, 1997.
[11] G. Bhargava, P. Goel, and B.R. Iyer, “Hypergraph Based Reorderings of Outer Join Queries with Complex Predicates,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’95), pp. 304-315, 1995.
[12] 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.
[13] G. Graefe, U. Fayyad, 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.
[14] J. Clear, D. Dunn, B. Harvey, M.L. Heytens, and P. Lohman, “Non-Stop SQL/MX Primitives for Knowledge Discovery,” Proc. ACM SIGKDD Fifth Int’l Conf. Knowledge Discovery and Data Mining (KDD ’99), pp. 425-429, 1999.
[15] C. Cunningham, G. Graefe, and C.A. Galindo-Legaria, “PIVOT and UNPIVOT: Optimization and Execution Strategies in an RDBMS,” Proc. 13th Int’l Conf. Very Large Data Bases (VLDB ’04), pp. 998-1009, 2004.
[16] 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.
[17] C. Ordonez, “Vertical and Horizontal Percentage Aggregations,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’04), pp. 866-871, 2004.
[18] Carlos Ordonez and Zhibo Chen, “Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 4, APRIL 2012.

BIOGRAPHIES

M.Suneetha is student of MLRIT, Hyderabad, and AP, INDIA. She has received B.Tech Degree Computer Science and Engineering, M.Tech Degree in Computer Science and Engineering. Her main research interest includes data mining, Databases and DWH.

G.Kiran Kumar is working as an Associate Professor in MLRIT JNTUH, Hyderabad, and Andhra Pradesh, India. He has completed M.Tech (C.S.E) from JNTUH. His main research interest includes data mining, Databases and DWH.