International Journal of Research in Advent Technology, Vol.2, No12, December 2014
E-ISSN: 2321-9637

Optimizing the Query for Preparing the Dataset for Horizontal Aggregation Using Case and Pivot Method

AMRUT A. PATIL[1], D.M.THAKORE[2]
Computer Engineering Department, BVDUCOEP PUNE[1, 2]
Email: amrutpatil28@gmail.com[1], dmthakore@bvcoep.edu.in[2]

Abstract- Data mining is widely used domain for extracting the useful information from large historical data. We can't use the database directly for the data mining which is used by the various enterprises. Preparing a data set for analysis is generally the most time consuming task in a data mining project which requires many complex SQL queries, and complex operations such as joining tables and aggregating columns. Existing SQL aggregation produces the single result per column which result improved by the proposed horizontal aggregation. Horizontal aggregation defines the new class function. It also reproduces the SQL code automatically and generates the number of output per row. It performs various operations like CASE and PIVOT. Here PIVOT operation perform row to column transformation and CASE method exploring the programming which performs two scan method for the proposed horizontal aggregation. The pivot and case method generated same data set but pivot is required the less time for executing the quires and optimizing the result.

Index Terms- Case, dataset, horizontal aggregation, Pivot.

1. INTRODUCTION

In a relational database, distinctly with normalized tables, an important attempt is prescribe to provide a summary data set that can be used as input for data mining or statistical algorithm. Most algorithms requires as horizontal layout as an input with one variable or dimension per column. That is the case with multiple models like classification, regression, clustering, PCA and other types of tools. Each research discipline uses separate terminology to describe the data set. In data mining the common conditions are point-dimension. Observation-variable commonly use in Statistics literature. Instance features uses in Machine learning algorithm. In our research we can show a new aggregate function that can be produce a new dataset in horizontal layout with extending SQL capabilities, aggregation and automating SQL Query writing. We show evaluating horizontal aggregations is a challenging and exciting problem and we producing alternate methods and optimizations for their efficient evaluation. As mentioned above, building a proper data set for data mining intention is a time-consuming task. This task generally requires writing more complex SQL statements/Queries or customizing SQL code if it is generated automatically by different tools. There are two main elements in such SQL code: joins and aggregations. We concentrate on the second one. Joins includes the SPJ method that will be (select, project, join). This method can be requiring more time than the aggregation function. The most extensively-known aggregation is the sum of a column over groups of rows. Some other aggregations respond the average, minimum, maximum or row count over groups of rows.

2. RELATED WORK

'Data mining, also popularly known as Knowledge Discovery in Databases (KDD) refers to the nontrivial extraction of complicated, previously unknown and potentially beneficial information from data in databases. It is the process of analyzing useful information that can be used to increase income, costs or both. Data mining software is one of the numbers of analytical tools for analyzing data which allows users to analyze data from many different dimensions, angles, categorize it and summarized the relationships. We are in an age often referred to as the information age. In this information age, because we expect that information pass to power and success and thanks to fake technologies. We have been collecting tremendous total of information. Initially, with the coming of computers and indicate for mass digital storage. We begin collecting and storing all sorts of data, counting on the power of computers to serve sort through this blend of information. Unfortunately, these heavy collections of data stored on separate structures very rapidly became overwhelming. The efficient database management systems have been very important property for management of a huge corpus of data and particularly for effective and efficient recovery of particular information from a huge collection whenever needed. To prepare summarized format for data mining algorithm, many methods are produce by researchers in the past.

Carlos Ordonez, Zhibo Chen [1] introduces horizontal aggregations in SQL to prepare data sets for data mining analysis. This paper gives the information about three methods of horizontal aggregation. K. Anusha, P. Radhakrishna, and P. Sirisha[2] gives information about horizontal aggregation using spj method and equivalence of Methods. Vani A. Hiremani and Krupali R. Dhawale [14] give fundamental methods to evaluate horizontal aggregation in SQL, where it gives comparison of three horizontal aggregation methods.
SQL has been around since its inception and being used widely for interacting with relational databases both for storing and recover data. The SQL supply all kinds of constructs such as projections, selections, aggregations, sub queries and joins. Query optimization and using the result of query further is a necessary task in database operations. As part of queries the aggregations are used to get summary of data. Aggregate functions such as MIN, MAX,SUM, COUNT, and AVG are used for obtaining summary of data [1], [3] .These aggregations produce a single value output and can't provide data in horizontal layout which can be used for data mining operations. In other term, the vertical aggregations can't generate data sets for data mining. The association rule mining is the problems belong to OLAP processing. The aim of this is to verify data mining operations effectively. The drawback of this is that it is not effective of generate results in tabular format with horizontal layout fit for data mining operations. In a clustering algorithm is search which makes use of SQL queries internally.
Traditional query optimizations use tree-based plans for optimization. This is similar to SPJ process. CASE is also used with SQL optimizations. The PIVOT in SQL is used for pivoting results. Lot of research has been around on aggregations and optimizations of SQL operations. They also contain cross tabulation and explored much in case of cube queries. UNPIVOT and PIVOT are two operators on tabular data that exchange rows and columns, enable data transformation useful in data modeling, data analysis and data presentation [1], [3].
They can very easily be implemented inside a query processor, much similar select, project and join. In a design contribute opportunities for better performance, both during query execution and query optimization. For data mining operations that generate decision trees, vertical aggregations can be usage while the horizontal aggregations generate more suitable horizontal layout that is most suited for data mining operations. SQL Server both pivot and unpivot operations is made available.
Vertical Aggregation this type of aggregation uses aggregate functions supported by SQL are MIN, MAX, SUM, COUNT and AVG. These functions produce single value output [5]. The result of vertical aggregations is useful in computations or calculations. However, they can't be directly used in data mining operations. Existing SQL aggregations have limitations to prepare data sets because they return one column per aggregated group.
Limitations of Vertical aggregation
i) To return one column per aggregated group
ii) Existing SQL aggregations have limitations to prepare data sets.
Group function returns a result based on a group of rows. The functions are commonly mathematical functions. To find out the single value results likes finding minimum and maximum test score also sum of test scores of student of student these kinds of operations can be done by group function.

3. EXISTING SYSTEM

Methods of vertical aggregation: Aggregate functions are a special category with other rules. These functions calculate a return value across all the items in a result set, so it require a FROM clause in the query:
For example
Select count (product_Name) from product_catalog;
Select max (height), avg (height) from census_data where age > 30;
There are several methods of vertical aggregation:

i) SUM (): An aggregate function that returns the sum of a set of numbers. Its single argument can be the numeric result or numeric column of a function or expression can apply to the column value. Rows indicated a NULL value for the specified column is skipped. If the table is not contains any value or all the values supplied to MIN are NULL, SUM returns NULL. When the query contains a GROUP BY clause, it returns one value for each combination of grouping values.
SELECT student_name, SUM (test_score)
From student GROUP BY student_name
The example gives the result as a student name that has secure sum of test score.

ii)MAX():An aggregate function that returns the maximum value from a set of numbers. Its single argument can be the numeric result or numeric column of a function or expression can apply to the column value. Rows indicated a NULL value for the specified column is skipped. If the table is not contains any value or all the values supplied to MIN are NULL, MAX returns NULL. When the query contains a GROUP BY clause, it returns one value for each combination of grouping values.
Example: SELECT emp_name, MAX (Sal) From company GROUP BY emp_name
The example gives the result as an Employee name that has secure maximum Salary.

iii) COUNT (): It is an aggregate function that returns the number of row or the number of non-NULL rows.
Example:
Select COUNT (*) from student;
The result of above syntax measures the number of rows in the student table.
Select COUNT (Sal) from emp;
The result of above syntax count Salary amount.

iv) AVG: An aggregate function that returns the average value from a set of numbers. Its single argument can be the numeric result or numeric column of a function or expression can apply to the column value. Rows indicated a NULL value for the specified column is skipped. If the table is not contains any value or all the values supplied to MIN are NULL, AVG returns NULL. When the query contains a GROUP BY clause, it returns one value for each combination of grouping values.
Example:
SELECT student_name, AVG (test_score) from student
GROUP BY student_name
The example gives the result as a student name that has secure average of test score.

As horizontal aggregations are capable of producing data sets that can be used for real world data mining activities. In our project simple, powerful methods is use to generate SQL code to return aggregated columns in a horizontal tabular layout and returning a set of numbers instead of one number per row [2]. This operation is required in a number of data mining tasks such as unsupervised and data summation classification, as well as segmentation of huge heterogeneous data sets into the short homogeneous subsets those can be easily managed, separately analyzed and modeled. Then to created datasets for data mining related works, summary and efficient of data will be needed.
The main aim of horizontal aggregation is to define a template to generate SQL code combining aggregation and pivoting (transposition). The second aim is to extend the SELECT statement with a "Group By" clause that combines the transposition with aggregation. Horizontal aggregations provide several unique features and advantages [1]
First, they represent a template to generate SQL code from a data mining tool. SQL code automates writing SQL queries, optimizing and testing them for accuracy. This SQL code reduces manual work in the data preparation phase in a data mining projects. The second, since SQL code is automatically generated it is likely to be more efficient than SQL code written by an end user. For example a person who does not know SQL well or someone who is not familiar with the database schema. Third, it requires less time to create the data sets. Fourth, the data set can be created entirely inside the DBMS.

1) Methods: Horizontal aggregation is evaluated using three fundamental methods: case, SPJ (Select Project Join) and pivot [1], [3]
i) CASE: For this method we use the "case" programming construct available in SQL. The case statement respond a value selected from a set of values supported on Boolean expressions. In a relational database theory point of view this is equal to doing a simple projection/aggregation query where each non-key value is assumed by a function that respond a number based on some conjunction of conditions.
ii) SPJ: This method is based on the relational operators only. In SPJ method one table is created with vertical aggregation for each column. All such tables are joined in order to generate a table containing horizontal aggregations. This method performs operation on the fact table like Select, Project and Join.
iii) PIVOT: The pivot operator is a built-in operator which transforms row to columns. Pivot internally needs to determine how many columns are needed to store the transposed table and it can be combined with the GROUP BY clause

Table1.Example of Horizontal Aggregation from Vertical Aggregated Table

F

X	C1	C2	Y
1	3	A	9
2	2	B	6
3	1	B	10
4	1	B	0
5	2	A	1
6	1	A	NULL
7	3	A	8
8	2	A	7

FH

C1	C2A	C2B
1	NULL	10
2	8	6
3	17	NULL

Table1. Gives an example showing that F is the input table and a horizontal aggregation stored in FH. Table F is input table where X shows number of row, C1 is integer values, C2 consists of two values A and B. Table F is aggregated using horizontal aggregation and that horizontal aggregation is shown in the output table FH. In output table FH, column Y is aggregated and we get the aggregation table which requires less space as compare to the input table.

4. PROPOSED METHOD

Optimizing a workload of horizontal aggregation queries means we modifying the query optimizer, Applying Horizontal aggregation to Holistic function are most challenging tasks in horizontal aggregation for further research purpose. For

optimizing workload and reducing the time to execute query in this project we first perform vertical aggregation using input table and from vertically aggregated table we compute the horizontal aggregation. Using previous method we can compact the code but to execute the query require more time than proposed system.

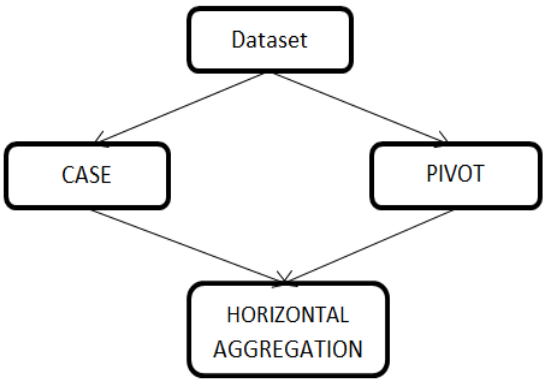


Fig 1.Arichitcure diagram for horizontal aggregation

In above Fig. 1 it shows the flow of the horizontal aggregation in which columns which is used to perform the operations such as CASE and PIVOT for dimensionality. In this diagram, CASE method performs two scan operations. Here it computes horizontal aggregation to obtain optimized solution.
Case-Two basic strategies to compute horizontal aggregations: The first strategy is to compute directly from input table. The second approach is to compute vertical aggregation and save the results into temporary table and then horizontal aggregation is computed by taking input from vertical table. In proposed system we use first approach of CASE method to compute horizontal aggregation.

Query- insert into FH
```
Select
C1
, Sum (case when C2='A' then Y
Else null END) as C2A
, Sum (case when C2='B' then Y
Else null END) as C2B
From F
Group BY C1;
```

Pivot-The pivot is the element of an array or a matrix which is selected first by an algorithm, to do certain calculations. In the case of matrix algorithms, a pivot entry is mainly need to be at least distinct from zero, and often separated from it; in this case support this element is called pivoting. Pivoting may be succeeding by an interchange of rows or columns to convey the pivot to a fixed position and allow the algorithm to proceed successfully and maybe to reduce round-off error. Pivot converts an arrangement of rows into a series of fewer rows with new columns. Data in one origin source column is used as the data for that new column and another column is used to determine the new column for a row and PIVOT transact on a table, like other operations, converting from narrow form to broad form.

Query- Insert into FH
```
Select
C1
, [A] as C2_A
, [B] as C2_B
From
Select C1, C2, Y from F
) As P
Pivot (
Sum (Y)
For C2 in ([A], [B])
As pvt;
```

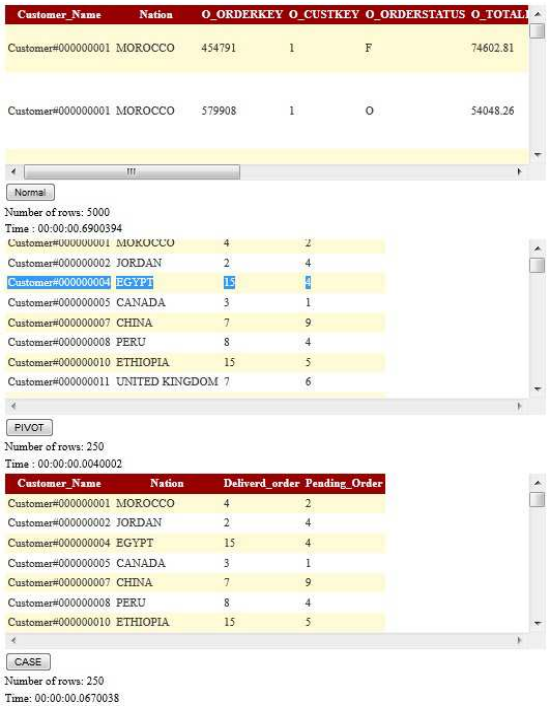
5. RESULTS AND DISCUSSION

Setup: Computer Configuration and Data Sets We used SQL Server 2008 r2, running on a DBMS server running at 2.4 GHz, core i5 processor, 4GB of RAM and 1 TB hard disk. The SQL code generator was programmed in a C#.net language and connected to the server. Here, we used large synthetic datasets which is generated by TPC-H generator. The SQL code generator was programmed in a C#.net language and connected to the server. We analyzed queries and only one horizontal aggregation, with different grouping and horizontalization columns. Each experiment repeated 3times and report the time in milliseconds. We evaluate horizontal aggregation queries using different tables and TPC-H generator as input for evaluating time require to evaluate aggregation queries.

In fig 2 shows that the result of the methods that can be used same dataset but different result and row counts. So we can easily understand the difference between different methods that be used for horizontal aggregation.

Experimental Evaluation

In Table 2 and 3 compare the three query optimization methods. We use the same N value as an input for each method and evaluated the time in second. In Table 2 we can use the N=150000 as an input and calculated the time.in table3 we use N=1500000 input and evaluating the time



Customer_Name	Nation	O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTAL
Customer#000000001	MOROCCO	454791	1	F	74602.81
Customer#000000001	MOROCCO	579908	1	O	54048.26

Normal
Number of rows: 5000
Time : 00:00:00.6900394

Customer#000000001	MOROCCO	4	2
Customer#000000002	JORDAN	2	4
Customer#000000004	EGYPT	15	4
Customer#000000005	CANADA	3	1
Customer#000000007	CHINA	7	9
Customer#000000008	PERU	8	4
Customer#000000010	ETHIOPIA	15	5
Customer#000000011	UNITED KINGDOM	7	6

PIVOT
Number of rows: 250
Time : 00:00:00.0040002

Customer_Name	Nation	Deliverd_order	Pending_Order
Customer#000000001	MOROCCO	4	2
Customer#000000002	JORDAN	2	4
Customer#000000004	EGYPT	15	4
Customer#000000005	CANADA	3	1
Customer#000000007	CHINA	7	9
Customer#000000008	PERU	8	4
Customer#000000010	ETHIOPIA	15	5

CASE
Number of rows: 250
Time: 00:00:00.0670038

Fig2. Short of experiment performed by one query
.

Table2. Query Optimization time in sec where N=15000

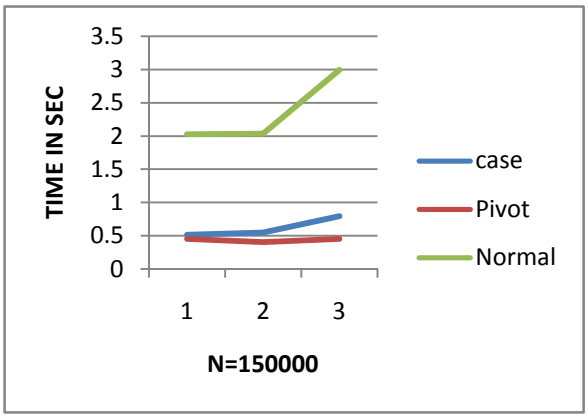
N	CASE	PIVOT	NORMAL
150000	0.514	0.453	2.028
	0.514	0.453	2.028
	0.514	0.453	2.028

Table3. Query Optimization time in sec where N=150000

N	CASE	PIVOT	NORMAL
1500000	5.319	2.698	42.182
	5.319	2.698	42.182
	5.319	2.698	42.182

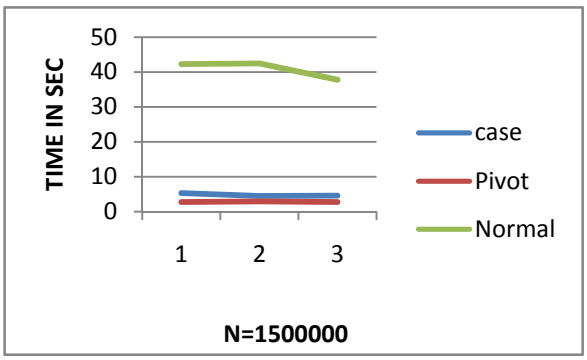
In figure 3 and 4 shows three methods with same input. The pivot and Case are used for optimizing the query result. The normal method required the more time than the case and pivot method.

Fig3. Short of experiment performed by one query



Also pivot and case produced the same result .But pivot is required the less time for executing the query than the case method.

Fig4. Short of experiment performed by one query



6. CONCLUSIONS

In this paper we can develop the new class for horizontal aggregation. We can compare the three methods for optimizing the quires .The normal method required the more time for executing quires in large dataset than pivot and case method.in normal dataset. The rows count also more than other two methods. The pivot and case method generated same data set but pivot is required the less time for executing the quires and optimizing the result. Because the in large data set case method not performed well so it will required more time than pivot method. Query optimization is most challenging task in the horizontal aggregation. We can try to achieve better query optimization.

REFERENCES

[1] Carlos Ordonez, Zhibo Chen, "Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis", IEEE Transactions on Knowledge and Data Engineering, Digital Object Indentifier 10.1109/TKDE.2011.16, 2012.

[2] K. Anusha, P. Radhakrishna, and P. Sirisha, "Horizontal Aggregation using SPJ Method and Equivalence of Methods", March 2012.
[3] V. Pradeep Kumar, Dr. R. V. Krishnaia, ."Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis", Nov. - Dec. 2012.
[4] C. Ordonez,"Vertical and horizontal percentage aggregations", In Proc.ACM SIGMOD Conference, pages 866–871, 2004.
[5] http://jpinfotech.blogspot.in/2011/09/horizontal-aggregations-in-sql-to.html
[6] Data sets: http://www.tpc.org/tpch/
[7] Carlos Ordonez "Horizontal Aggregations for Building Tabular Data Sets", ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2004.
[8] S. Sarawagi, S. Thomas, and R. Agrawal, "Integrating association rule mining with relational database systems: alternatives and implications", In Proc. ACM SIGMOD Conference, pages 343-354, 1998.
[9] Witkowski, S. Bellamkonda, T. Bozkaya, G. Dorman, N. Folkert, A. Gupta, L. Sheng and S. Shubramanian, "Spreadsheets in RDBMS for OLAP", In Proc. ACM SIGMOD Conference, pages 52-63, 2003.
[10] Jincy Annie V. V, J. A. M Rexie, "Evaluating Aggregation Function with Partial Aggregations", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 2, Issue 2, February 2013.
[11] A.Lakshman Rao, V.V.Satyanarayana Murty.S, "An Experimental Analysis using PIVOT Method in Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 11, November 2012.
[12] Lavina D. Panjwani, Richa K. Makhijani, "Performance Evaluation of Horizontal Aggregation Techniques in Sql", International Journal of Computer Science and Engineering (IJCSE) ISSN 2278-9960Vol. 2, Issue 3, July 2013, 27-34.
[13] Sonali Karle, Swati Avhad, Ashlesha Shelar, Prof. Suvarna Pawar, Kajal Dighe, "Datasets Preparation in SQL using Horizontal Aggregation", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Vol.2 Issue IV, April 2014.
[14] Krupali R. Dhawale 1, Vani A. Hiremani 2,"Fundamental methods to evaluate horizontal aggregation in SQL", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 10, October 2013.