A COMPARATIVE ANALYSIS OF HORIZONTAL LAYOUT REPRESENTATION OF DATA

S. Brintha Rajakumari[1], Dr.C.Nalini[2]

[1]Research Scholar, [2]Professor, Department of CSE,
Bharath University, Chennai (India)

ABSTRACT

Data Mining is a process of extracting useful knowledge from the database. To do that one can need aggregative function for data set preparation. Now a day, it is time consuming process and cumbersome process for the researcher. This paper compared the horizontal layout representation of aggregated data using structure query language and the data generated by data conda software tool.

Keywords: Aggregation, Dbms, Data Mining, Data Conda, Sql.

I. INTRODUCTION

Data mining refers the extracting or mining knowledge from huge amount of the data. The main objective of the data mining techniques is to extract regularities from a large amount of data. Data mining has a variety of fields which provides the different tools and the techniques for handling the large database. As we know that the relational database are the best approach to handle the structured data but now a days the data are not only available in relational form but also available in multiple form. If we gather all the tables in centrally then which is made up of attributes that summarizes or aggregate the information found in other tables. The data preparation step covers all activities to construct the final dataset for modeling from the raw data. Tasks include database, table, record, and field selection as well as cleaning, aggregation and transformation of data. Aggregating functions have a set of values as input and produce one value as output.

In the section 2 describe the related research carried out in the field of aggregation. Section 3 presents the comparative analysis of various research works with different parameters. Section 4 and 5 describe the SPJ method and horizontal representation using data conda followed by conclusion.

II. RELATED WORKS

Aggregation concept is a powerful tool in database design, and consequently, preserving aggregation in database implementation is essential. The aggregation problem becomes especially acute in a database management system (DBMS) since such a system contains a large volume of data that could form aggregates that are more sensitive than their constituent parts. It is the intent of this paper to investigate the aggregation problem in the context of a database [1]. Aggregation is an important concept in database design where composite objects can be modeled during the design of database applications. Therefore, maintaining the aggregation concept in database implementation is essential [2]. Aggregation is a composition (part-of) relationship, in which a composite object (“whole”) consists of other component objects (“parts”) [3]. In this paper [4] introduced three SQL implementations of the popular K-means cluster rule to integrate it with a relative DBMS: (1) an easy translation of K-means computations into SQL. (2) Associate degree optimized version supported improved knowledge organization, economical categorization, adequate statistics and rewritten queries. (3) An incremental version that uses the optimized version as a building block with fast convergence and automated reseeding.

In the paper [5] have proposed two aggregate functions to compute percentages. The first function proceeds one row for each computed percentage and it was called a vertical percentage aggregation. The second function returns each set of percentages adding 100% on the same row in horizontal form and it was called a horizontal percentage aggregation. They are used as a framework to study percentage queries. Two practical issues when computing vertical percentage queries were identified: missing rows and division by zero. Horizontal percentages do not present the missing row issue. it with efficiency assess proportion queries with many optimizations together with categorization, computation from partial aggregates, victimization either row insertion or update to supply the result table, and reusing vertical percentages to induce horizontal percentages. And compared proposed percentage aggregations against queries using OLAP aggregations. Each projected aggregations area unit considerably quicker than existing OLAP mixture functions.

The Bayesian classifier could be a basic classification technique. In the paper [6], they targeted on programming Bayesian categoryifiers in SQL and introduced two classifiers: Naive Bayes and a classifier supported class decomposition victimization K-means cluster and regarded two complementary tasks: model computation and marking an information set. They analyzed the way to remodel equations into economical SQL queries and introduced many question optimizations. We have a tendency to conduct experiments with real and artificial knowledge sets to guage classification accuracy, question optimizations and quantifiability. The Bayesian classifier is a lot of correct than Naïve Bayes and call trees. Distance computation is considerably accelerated with horizontal layout for tables, denormalization and pivoting. They compared the Naive Bayes implementations in SQL and C++: SQL is concerning fourfold slower. Bayesian classifier in SQL achieves high classification accuracy, will with efficiency analyze massive knowledge sets and has linear quantifiability.

Association rules area unit is an information mining technique accustomed discover frequent patterns in a very data set. In the paper [7] association rules area unit employed in the medical domain, wherever knowledge sets area unit usually high dimensional and tiny. The chief disadvantage concerning mining association rules in a very high dimensional knowledge set is that the variety of patterns that area unit discovered, most of that area unit moot or redundant. Many constraints area unit projected for filtering functions, since our aim is to get solely important association rules and accelerate the search method. A greedy rule is introduced to reason rule covers so as to summarize rules having the same consequent. The importance of association rules is evaluated victimization three metrics: support, confidence and raise. Experiments specialize in discovering association rules on a true knowledge set to predict absence or existence of cardiovascular disease. Constraints area unit shown to considerably scale back the quantity of discovered rules and improve time period. Rule covers summarize an outsized variety of rules by manufacturing a compact set of rules with prime quality metrics.

For a good type of classification algorithms, scalability to massive databases are often achieved by perceptive that the majority algorithms area unit driven by a group of adequate statistics that area unit considerably smaller than the information within the paper [8]. By counting on a SQL backend to reason the adequate statistics, they leverage the question process system of SQL knowledge bases and avoid the requirement for moving data to the shopper. They need bestowed a brand new SQL operator for Unpivot that allows economical gathering of statistics with least changes to the SQL backend. This approach provides the ends up in important increase in performance while not requiring any changes to the physical layout of the information. They showed analytically however this approach outperforms an alternate that needs dynamic within the knowledge layout. They conjointly compared impact of information illustration and show that a “dense” illustration is also most popular to a “sparse” one, even once the information area unit fairly distributed.

In the paper [9] introduced a brand new category of mixture functions, referred to as horizontal aggregations. Horizontal aggregations area unit helpful to make knowledge sets in tabular type. A horizontal aggregation returns a set of varietys rather than one number for every group. They projected a straightforward extension to SQL customary mixture functions to reason horizontal aggregations that solely needs specifying subgrouping columns. They explained the way to assess horizontal aggregations with customary SQL victimization two basic ways. the primary one (SPJ) depends on relative operators. The second (CASE) depends on the SQL case construct. The SPJ strategy is attention-grabbing from a theoretical purpose of read as a result of it's supported choose, project, natural be part of an outer be part of queries. The CASE strategy is very important from a sensible position given its potency. The projected horizontal aggregations are often used as a way to mechanically generate economical SQL code with three sets of parameters: grouping columns, subgrouping columns and collective column. On the opposite hand, if customary SQL mixture functions area unit extended with the "BY" clause, this work suggests the way to modify the SQL program and question optimizer. The impact on syntax is least. The fundamental distinction between vertical and horizontal aggregations, from the user purpose of read, is simply the inclusion of subgrouping columns.

III. SPJ METHOD AND EVALUATION

Table is a collection of records organized in rows and columns. The definition is in OLAP terms. Let F be a temporary table or view based on a star join query on several tables. In that k is a primary key of an integer attribute and D1, D2 are different dimensions of nominal and numerical attribute. An example showing the input table is on the Table 1. And the traditional SQL aggregation sum () result is on the Table 3. The horizontal aggregation of query result stored in the Table 3. The main goal of horizontal aggregation is to transform the Table 2 into Table 3 representation. A method, SPJ method, is used to evaluate a horizontal aggregation, which relies on relational operations. That is, select project, join and aggregation queries.

In order to evaluate this query the query optimizer takes three input parameters:

The input table F,
The list of grouping columns L1;….; Lm ,
The column to aggregate (A).

The Select syntax is as follows:

```
SELECT L1; … ; Lj
FROM F
GROUP BY L1; . . . ; Lj;
```

Experiment was performed this method using MS SQL Server 2008 and found out the size of the table [11,12]. Tables 2 and 3 show horizontal layout representation reduced the rows than vertical representation. A data table 1 is presented containing 3 attributes, such as D1, D2 and A.

Table 1. Example data table

Sl. No.	D1	D2	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

Table 2. Vertical aggregation of table

D1	D2	A
1	X	Null
1	Y	10
2	X	8
2	Y	6
3	X	17

Table 2 shows vertical aggregation of similar data and the SQL query is given as,

```
SELECT D1, D2, SUM (A)
FROM Table 2
GROUP BY D1, D2 ORDER BY D1, D2
```

Table 4 shows horizontal layout of similar data and the SQL query for this is given as,

```
SELECT D1,[X] AS D2X,[Y] AS D2Y
FROM
(SELECT D1, D2, A FROM Table2 AS source table
PIVOT(SUM(A) FOR D2 IN([X],[Y])) AS pivot table;
```

Table 3. Horizontal layout of data

	D2X	D2Y
1	Null	10
2	8	6
3	17	Null

IV. HORIZONTAL LAYOUT REPRESENTATION USING DATACONDA

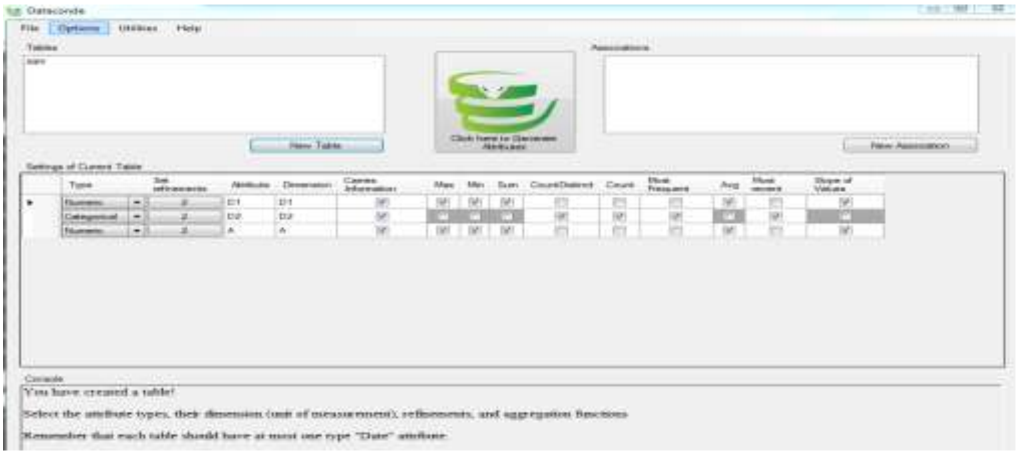
Data conda [13, 14] tool embeds the data mining process but the user only has to organize the available data in a relational database and indicate a dependent variable. Data conda will automatically compute a large number of predictors without the need to formulate hypotheses. These predictors are built by selecting, aggregating, and filtering the information available in the database. Data conda automatically generates the hypotheses to explain the dependent variables. There are three important concepts used in Data conda: the concepts of Table, Attribute, and Association.

A Table is a set of records organized in rows and columns. The columns of a table are called attributes. Generally, attributes represent characteristics of the entities. Each attribute is of one of the four types.

- ID or key: An ID attribute has the goal of identifying and referring to records in a table. There are two types of IDs: primary keys and foreign keys. Primary keys are unique identifiers within a table. Foreign keys are pointers to primary keys.
- Date: A date attribute is the timestamp that characterize the entities of the table.
- Numeric: A numeric attribute takes only numeric values.
- Categorical: A categorical attribute takes only a finite set of values.

In Data conda is in fig1, categorical attributes are stored as text, even if they are number which can be 0 or 1. Categorical attributes are also known as factor or nominal attributes. Attributes are also characterized by a dimension, which represents the unit of measurement of that attribute. Data conda allows attributes of the same dimension to be compared to each other.

An association A → B is a relationship between two tables A and B. Data conda considers only two types of associations: 1-to-1 and 0-to-N. A →B is a 1-to-1 association if every record of A is associated to exactly one record of B. On the other hand, A → B is a 0-to-N association if every record of A is associated to any number of records in B. The first step is to load the individual tables in memory. Click on "New Table" and select the file. Note that the software will generate an error if the file is being used by another process. If an attribute carries information, then the user may select what aggregating functions should be applied to it. Aggregating functions receive a list of values as input and return a single value as output. Click on the central button "Click here to generate attributes" to open the Generate Attributes form. This form contains the main options to perform the attribute generation.



Type	Set refinements	Attribute	Dimension	Carries Information	Max	Min	Sum	CountDistinct	Count	Most Frequent	Avg	Most recent	Slope of Values
Numeric	2	D1	D1	☑	☑	☑	☑	☐	☐	☐	☑	☐	☑
Categorical	2	D2	D2	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Numeric	2	A	A	☑	☑	☑	☑	☐	☐	☐	☑	☐	☑

You have created a table!

Select the attribute types, their dimension (unit of measurement), refinements, and aggregation functions

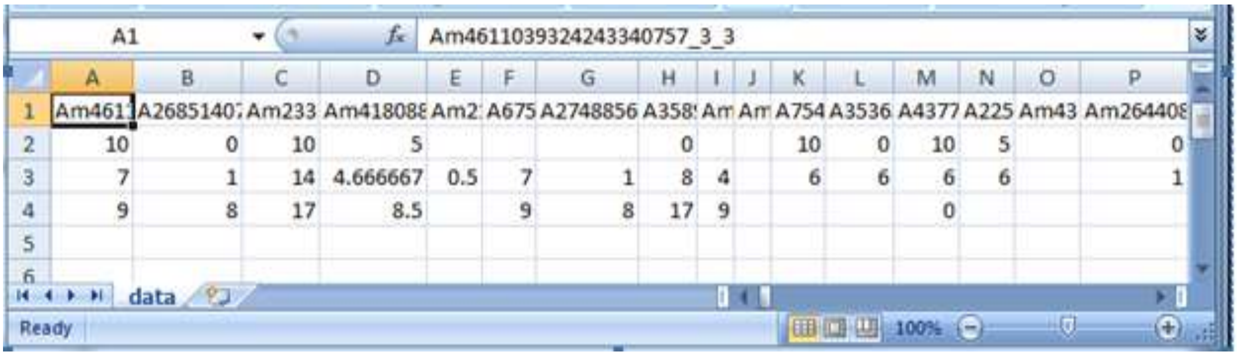
Remember that each table should have at most one type "Date" attribute

Figure1: Data Conda

The following sample generated by the data conda and the result will automatically stored in the data.csv excel file is in the fig2.

Am4611039324243340757_3_3:
Numeric,A
DESCRIPTION: Max(A) among tableK
0:Target->Max(A)
1:tableD1=>Max(A)
2:tableK.A

A2685140705483091023_3_3:
Numeric,A
DESCRIPTION: Min(A) among tableK
0:Target->Min(A)
1:tableD1=>Min(A)
2:tableK.A



A1 Am4611039324243340757_3_3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Am4611	A2685140	Am233	Am41808	Am2	A675	A2748856	A358	Am	Am	A754	A3536	A4377	A225	Am43	Am26440
2	10	0	10	5				0			10	0	10	5		0
3	7	1	14	4.666667	0.5	7	1	8	4		6	6	6	6		1
4	9	8	17	8.5		9	8	17	9				0			
5																
6																

Figure 2. Generated attributes in 'data' file From that the attributes A3589638704703334615_3p1_3 and A4377301488491884376_3p1_3ch contains the value which is similar to the Structure query language horizontal representation of data.

V. CONCLUSION

This paper presented the comparison between the horizontal layout representation of aggregated data using structure query language and the data generated by data conda software tool. From that analysis the dataconda gives the result faster than the standard structure query language. And it is the easiest tool to find the aggregate values. In future we have planned to analyze whether this tool works well for very large data set and performance better than the traditional system.

REFERENCES

[1] Thomas H. Hinke.,Inference," Aggregation Detection In Database Management Systems",*IEEE*,1988.

[2] Johanna Wenny Rahayu and David Taniar ,"Preserving Aggregation in an Object-Relational DBMS", *Springer-Verlag Berlin Heidelberg* , pp. 1–10, 2002.

[3] Rumbaugh, J. et al, "*Object-Oriented Modelling and Design*"(Prentice-Hall, 1991).

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

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

[6] 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.

[7] Carlos Ordonez Norberto Ezquerra Cesar A. Santana ,”Constraining and Summarizing Association Rules in Medical Data”, *Knowledge and Information Systems* , 9(3):259-283, 2006.

[8] 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.

[9] 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.

[10] Javier Garcia-Garcia, Carlos Ordonez, “Extended Aggregations for Databases with Referential Integrity Issues”, Elsevier DKE, 69(1):63-95, 2010.

[11] S.Brintha Rajakumari and C.Nalini ,”An efficient Data Mining data Set preparation using aggregation in relational database” , *Indian Journal of Science and Technology*,Vol 7(S5),Pp 44-46, June 2014.

[12] S.Brintha Rajakumari and C.Nalini ,”An efficient cost Model for data storage with horizontal layout in the cloud” , *Indian Journal of Science and Technology*,Vol 7(S3),Pp 45-46, March 2014.

[13] Michele Samorani, *Dataconda*, UserGuide, 2015.

[14] Samorani et al,”A Randomized Exhaustive Propositionalization Approach for Molecule Classification”, *INFORMS Journal on Computing*, 23(3), pp. 331–345,2011.