International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-4, Issue-ICCIN-2K14, March 2014

Continuous New OLAP Operations on Data Streams

Bhawna Suri

Abstract— ***Data stream is a continuous, real time, multidimensional, ordered sequence of data elements produced by the data source. The efforts have been made to do the analytic processing on data streams. In order to do so, a portion of the stream is captured in a window and is mapped to the multidimensional structure onto which the existing OLAP operations are applied. We have also worked on the same approach but here the multidimensional structure - Cube is analyzed using two new OLAP operations. These OLAP operations are Virtual- Rollup and Restrict. This paper aim towards a new angle of handling continuous analysis of data stream data using the new set of OLAP operators.***

Keywords-data streams, OLAP.

I. INTRODUCTION

A data stream is a continuous, real time, ordered sequence of data values produced by a data source[1,2]. The main characteristics of data stream are - data flows-in and flows out dynamically, it is multidimensional and is visible only once. The various data stream applications are like Internet traffic analysis, financial tickers, sensor data and transaction log mining. The data stream being an unbounded data set which is produced incrementally over time, rather than being available in full, before its processing begins and is queried in finite portions. To query it, a portion of stream data is picked up at a time, referred to as a window [1,3,4] and is used for desired analysis. Data moves into the window from one direction and moves out from the other direction continuously. The data in a window after processing produces an output stream. Due to the continuous nature of data streams, queries that are executed for data streams are continuous in nature. A continuous query is a query that is logically issued once and runs forever [1]. While evaluating continuous queries, it may be noted that it is a single query which is repeatedly executed on every new incoming unique data sets. To do the analysis of the data, which is temporarily captured in a window, it is then stored in a multidimensional structure onto which the new proposed OLAP operations are applied. The data in the warehouse is analyzed using Online Analytical Processing (OLAP) operations, which are helpful in decision making. Also some efforts have been made to use the multidimensional characteristic of data stream for analysis. To do the high level OLAP analysis as needed by the analysts, the raw data at the most detailed level is collected in a multi-dimensional structure. The few systems have attempted analytic processing of data streams. In [5] emphasis was on converting data streams into cuboids which could subsequently form the basis for OLAP operations. However, they have not proposed OLAP operations.

Manuscript received March 2014
Bhawna Suri, Bhagwan Parshuram Institute of Technology Delhi, India.

The STREAM project [6] had worked on the aggregate analysis using standard OLAP operations like slice-dice, Roll-up with standard aggregate functions. In this project the stream is converted to relations onto whom OLAP operations are performed and the result is produced as an output. The CQL is an extended SQL language defined for the data streams. In [7] the streams were analyzed using OLAP operations on data streams. They have also worked on the similar path as converting the stream-to-cube, analyzing the cube using OLAP operations and finally converting the cube-to-stream producing the resultant output stream. For doing the high-level OLAP analysis, the stream is converted to cube. They have introduced a multidimensional stream query language SQL_{MS} [7]which supports standard OLAP operations slice, roll-up and drill-down. Moreover, the standard aggregate functions were used with an OLAP operation on a cube created for a window. In this system, the dimensions of the cube are predefined and the stream data is treated as facts. In order to do the OLAP analysis of stream data, we are also converting the stream-to-cube, onto the cube the new proposed set of OLAP operations are performed and the resultant cube is mapped to produced a continuous output stream. In this paper, we have defined a new set of OLAP operators defined on the cube for better analysis. The conversion from and to the stream are not considered here.

A. Motivation

Consider the Stock Stream which is the data stream produced by the stock exchanges. We have considered this as an example to understand the nature of decision support required when handling stream data. Let the stream data consists of price and the volume of the stocks as metrics for analysis. The price and volume represents the unit price and volume trade of each stock. In the Stock Stream the data is coming every second. Without loss of generality, we assume that data from the stream is picked up at every 30 seconds. In the cube, the price and volume of the stocks are stored at every 30 seconds. Let us assume that there exist Time dimension with the hierarchy defined as timestamp ≤30 seconds ≤ 1minute≤5 minute≤ hour≤ ALL, implying the data at the level, 30 seconds can be rolled up to 1minute and further roll-up can be performed accordingly as per the requirement. Our approach in this paper is to answer the continuous OLAP queries for the data in a window using the multidimensional Cube. To achieve this, the data captured in the time-based window is mapped to the cube. The cube is defined using the Cube Data Model defined in [8]. Once the cube is populated, it can be analyzed, thereby producing the resultant cube. The resultant cube is then mapped to the output stream. This sequence is now executed for the subsequent windows thereby appending the new result sets in the output stream. The continuous evaluation stops when the system is halted. The new OLAP operators Virtual-Rollup and Restrict were defined in terms of context in [10] and are now reproduced for the Cube. Also, since the OLAP analysis is incomplete without the Roll-up operator.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

The Roll-up operator for the cube is also defined with standard aggregate functions and three new aggregate functions [8]. Additionally, the quantifiers were introduced in [9] to enhance querying capability. We believe that the standard aggregate functions are not adequate in all situations. Consider the Stock market, example to find the average stock price of IBM for last five minutes. This query is evaluated using the standard aggregate function AVG and however, does not really give any indication of the behavior of the values. As a result, of the above query evaluated using AVG does not indicate whether the value of stocks were increasing or decreasing. The indication of behavior of the stocks with time was discussed in [8]. We have defined the non-aggregate functions which are evaluated without performing roll-up but the set of values involved in roll-up are used, to get the result. To understand this need, consider the following two queries:

"What is the maximum price of IBM stock during a chosen time period?"

"When did IBM stock peak during a chosen time period?"

In the first query, the result is a single maximum value computed using MAX function. On the other hand, for the second query it is not enough to give a single value as the MAX value. We need to specify the time at which the IBM has maximum share value. If the cell values along a dimension are picked up, then the maximum value indicates the exact time slot when it was the maximum with NULL values elsewhere. To cater to the above requirements we have defined the non-aggregate functions. These non-aggregate functions are applied in conjunction with the Virtual Roll-up which is one of the OLAP operators. The target applications investigated in this paper are those which are not critically real time. Rather, it is assumed that the user can wait for a nearly bearable delay for supported decision. The streams chosen are those which are defined using a discrete, ordered time domain t. Example applications are stock market, temperature sensor data etc. The output data is also produced at discrete time slots. The answer is streamed out as and when the output tuples are generated. We assume that the stream has data and an associated explicit timestamp t, the timestamp attribute is the ordering attribute for the data. Thus, if data arrives at time t then we assume that no data tuple with timestamp earlier than t will appear after time t[10,11].

B. Contribution

To address the issues discussed above we propose:

a) The creation of Cube using Cube Data Model [8],
b) New OLAP operators on the cube- Restrict and Virtual Roll-up. The two non-aggregate functions

Peak and Nadir, are proposed in conjunction with Virtual Roll-up

C. Paper Outline

The layout of the paper is as follows. In Section II, the Cube Data Model is defined which is used for creating the Cube in our work. The new aggregate and non-aggregate functions and their need are described next in section III. The proposed Cube OLAP operators are described in section IV. The case study is defined in section V. Section VI is the concluding section.

II. CUBE DATA MODEL (CDM)

Many data models for multidimensional data have been defined [13,14]. We define the data model which forms the basis of operations considered in this proposal. The data model for cube C is defined below:

The Cube C as per CDM consists of dimensions and cell values. In this section we first define the cube followed by the structure of the stream.

A. The Cube

We first define Dimension and Fact scheme. Subsequently, a cube is defined.

Dimension

A dimension Di consists of a finite set of attributes {Ai1,Ai2,...,Aik}. Some attributes can participate in a hierarchy such that there is a partial order Aij ≤ Aik which implies Aij can be rolled to Aik. The dom(Aij) defines the domain of values for the attribute Aij. These values may or may not be ordered. If an order exists, then this order has to be specified. There exist a mapping function such that g: dom(Aij) →dom(Aij+1) so that the roll-up can be performed. Without loss of generality, we assume that the attribute Ai1is the lowest level of the dimension Di. Further, there always exists a Time dimension T where the values of the lowest member timestamp t of the hierarchy reflect the time of arrival of the data stream data.

Fact Scheme

A fact scheme has $\{m_1,m_2,\ldots,m_m\}$as measures which will be analyzed along the attributes A_{ij} of dimension D_i.

Cube

A cube is defined as a tuple consisting of dimensions and cell values. It is defined as a tuple $<D_c, V_c>$ where

Dc = finite set of dimensions i.e $\{D_1, D_2,\ldots,D_{n-1}, T\}$ set of n-dimensions

Vc = finite set of cube cell values where the values of the measures of analysis are $\{m_1,m_2,\ldots,m_m\}$defined corresponding to the lowest level $\{A_{11},A_{21},\ldots,A_{n-1},t\}$ of dimensions $\{D_1,D_2,\ldots,D_{n-1},T\}$.

B. The Stream Structure

The multidimensional structure of the stream data is of the form:

$S(D_1.A_{11}, \ldots ,D_{n-1}.A_{n-1},T.t, m_1, .. ,m_m)$

In the stream structure $D_1.A_{11}, \ldots ,D_{n-1}.A_{n-1}$are attributes, $m_1,m_2,\ldots,m_m$ are the measures for analysis forming the tuple and T.t is the timestamp marking the incoming tuple picked up from the stream. The stream structure gives the lowest level attributes of the dimensional hierarchy. The stream structure is mapped to a cube defined above. The example used throughout the paper is given below**.** We consider the Stock market as the target application. The price and volume are the measures representing the unit price and volume traded respectively. The dimensions are Stocks, Sensex and Time. The stream structure is

Stockstream(symbol/*unique stock *symbol**/,
senname /*sensex name*/,
price /*share price of each symbol*/,
volume /* volume traded of each symbol*/
timestamp /* time of incoming tuple*/)

The symbol, senname and the timestamp become the lowest level attributes of the dimensions

Stocks, Sensex and Time respectively of the cube C defined below:

C = <Dc, Vc>

where Dc = { Stocks, Sensex , Time } are dimensions

Vc = {price, volume} are the measures

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-4, Issue-ICCIN-2K14, March 2014

The hierarchy of Stocks is symbol □□category □□ALL. Similarly, the senname is the lowest level attribute of the dimension Sensex. The hierarchy of Sensex is senname □□ALL. The timestamp attribute of the stream is mapped to Time dimension. The hierarchy of the Time dimension is timestamp □□30seconds □□minute□ 5minute □□hour □□ALL. The attribute values of symbol are {IBM ,INF, WIP, TCS, SATYM, CIP, RAN, TORR,DABUR, JOHN, SCI, ONGC, MTNL, HPCL, BPCL,TATAMOTORS, REL INDIA, MAH&MAH, ITC}, and those of category are {pharma, old, software, psu}.The attribute values of senname are {BSE, NSE}.

III. AGGREGATE AND NON-AGGREGATE FUNCTIONS

The functions on a cube C are evaluated by picking up the cell values along the dimension(s) of a cube. We have introduced five new functions besides supporting the five standard functions which are MAX, MIN, AVG, COUNT and SUM. The newly defined functions are divided in two categories - aggregate and non-aggregate. The aggregate functions are MAX, MIN, AVG, COUNT and SUM and three new aggregate functions -increase,decrease and fluctuate. The two non-aggregate functions introduced are Peak and Nadir.

A. Need for new Aggregate and Non-aggregate Functions

As brought out above that the standard aggregate functions are not adequate in all situations. These aggregate functions are evaluated using the cell values along a given dimension. The result of aggregation is a single value. Therefore, the evaluation is meaningful with roll-up operation along the dimension in question. Consider the Stock market, example "to find the maximum stock price of among stocks of BSE sensex for the last five minutes". This query is evaluated using the standard aggregate function MAX. To evaluate the function MAX, the symbol is rolled up to category. To obtain the result for last 5 minutes one more Roll-Up is to be performed in Time dimension rolling 30sec to 5minute. The output of the query is the maximum share value in each category in last 5 minutes without actually pointing that which stock among each category has the maximum share price. Thus, to identify the stock and its maximum value we need the non-aggregate function Peak. Therefore, for Peak, a set of cell values are required to compute the maximum value but the result is not a single value but a set of values. The degree of the two sets is same with the maximum value in one of the cell values and NULL in the remaining. Here, the actual roll-up is not performed but the maximum value is assigned to the cell in question and is NULL elsewhere.
Summarizing, then, in our approach we divide the functions into two categories. One of which are aggregate functions and the other non-aggregate functions. Every function is computed using a set of cell values. In the case of aggregate functions the result is a single value and in the case of non-aggregate functions the result is a set of values where the number of elements in the domain and in the range is identical. This is how the peak is different from the standard function MAX. The non-aggregate functions which are evaluated without performing roll-up but the set of values involved in roll-up are used, to get the result.

B. Aggregate Functions

We have incorporated the standard aggregate functions AVG, MAX, MIN, COUNT and SUM in our system. In addition we have proposed three new aggregate functions as defined in [8]. These are increase, decrease and fluctuate. The functions are evaluated by picking up the cell values along Time dimension. The cell values which participate in the evaluation of the function are examined. If the cell values are:

1. Increasing with time then function increase evaluates to 1 else to 0
2. Decreasing with time then function decrease evaluates to 1 else to 0
3. Neither increasing nor decreasing with time then function fluctuate evaluates to 1 else to 0.

C. Non-aggregate Functions

The two new non-aggregate functions which are defined are Peak and Nadir. The functions are evaluated by picking up the cell values along a Time dimension. The cell values which participate in the evaluation of the function are examined. For Peak, the value in the cell which has the maximum value is retained and the remaining is set to NULL. Similarly, for Nadir, the value in the cell which has the minimum value is retained and the remaining is set to NULL. Aggregate functions are used with Roll-up which is explained above. The non-aggregate functions are evaluated using Virtual Roll-up as defined below in the next section.

IV. CUBE OLAP OPERATORS

In this section we define the proposed OLAP operations which are performed on a cube. The OLAP operations help the user to analyze the historical information of an enterprise for comparative analysis. The analytical queries enable the analysis of data for decision support. The OLAP operations Restrict, Roll-up and Virtual Roll-up are defined in this section. Each operation is executed for a cube C and the result is a new cube.

A. Restrict Operator

We define Restrict on a cube C as performing selection on attribute value(s) of a dimension(s) or measure(s). The result is a sub-cube C´. For all dimensions for which a selection predicate is specified, only the attributes satisfying the predicate appear in C´ and there is no change in the other dimensions. If the selection predicate involves measure(s) then only the cell values which satisfy the predicate are there in C´.
Formally, the Restrict operation is defined as:
$C' = \sigma\, p(C)$
where **σ** is the Restrict operator applied on cube C,
p is a predicate which is composed of one or more simple predicates. The simple predicates are combined using & (AND) and each simple predicate can be either Di.Aij= val where val is a value of the attribute Di.Aij or mk θ t where mk is a measure, θ is one of the relational operators <, >, =,≤,≥, ≠ and t is a constant.
Therefore the resultant cube C' is defined as
C' = <Dc', Vc'>,
where Dc' = Dc where for the dimension(s) Dj and their values are the same except for Di where the attribute values for Aij are those which satisfy the condition.
Vc' = set of cell values in C' along the attributes Aij of Di or the attributes that satisfy the condition imposed on the measure m_k.
Traditionally, Slice operation performs a selection on one dimension of the given cube, thus resulting in a sub-cube. On the other hand, Dice restricts the values of one or more

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

dimensions. It may be noted that using Restrict, both Slice and Dice can be realized in our system.

B. Roll-up Operator

The Roll-up operation in OLAP is defined as moving in the hierarchical structure of dimension from the finer-granularity data to coarser granularity applying the aggregate functions thereby summarizing the large amount of data. The Roll-up operation operates on a cube and produces another cube. It is defined as:

$$C' = \underset{Di.Aij}{\overset{Di.Aik}{\rho_f(mi)}}\,(C)$$

where ρ is the Roll-up operator, f is one of the aggregate functions defined in sec. III applied on the measure mi€ {m1,m2,……,mm}.

The resultant cube C' = <Dc', Vc'>

Dc' = Dc where the dimension(s) Di is rolled from Aij to Aik

Vc' = set of cell values in C' produced at the attribute level Aik of Di .

C. Virtual Roll-up Operator

In addition to the Roll-up operation with aggregate functions, we have defined a new OLAP operation called Virtual Roll-up for non-aggregate functions. The Virtual Roll-up is defined for the new non-aggregate functions Peak and Nadir**.** In the case of Virtual Roll-up, actual Roll-up is not performed. The data is not actually moved from the lower level to higher coarser level. However, the hierarchy defined for a dimension is utilized for the operation. Notice that if the hierarchy for a dimension Di is specified as Aij →Aik then for a single value of Aik there is a set of values of Aij associated with it. The cell values corresponding to this set of values of Aij are examined to compute the two non-aggregate functions Peak and Nadir**.** So, therefore, the attribute Aij of dimension Di is virtually rolled up to Aik**.** After computing these functions the values of Aij exist for which the corresponding cell value has peaked/dipped and the rest are set to NULL. When the Virtual Roll-up is performed on the cube the result is also a cube. The Virtual Roll-up operation is expressed as:

$$C' = V\,\underset{D_i.A_{ij}}{\overset{D_i.A_{ik}}{\rho\_f(m_i)}}\,(C)$$

where **V** stands for Virtual,

ρ is the Roll-up operator,

f is one of the non-aggregate function Peak or Nadir and the measure m_i €{m_1,m_2,……,m_m}

The resultant cube C' = <Dc', Vc'>

where Dc' = Dc

Vc' = set of cell values in C' are at the same level as in C but each cell value is either NULL or there exist exactly one value for the attribute A_{ij} of D_i. The single value is either the maximum or minimum depending on the f which is Peak or Nadir respectively.

V. CASE STUDY

Consider the Stock market data stream. The data in the Stock stream is coming every second. Without loss of generality, we assume that the data is picked up every thirty seconds. In order to do the OLAP analysis of data stream the cube for the stream structure is created as per the CDM defined above. Let the cube C for the Stock market has dimensions Stocks,Sensex and Time. The measures are price and volume representing the unit price and volume traded respectively.

The cube C is defined using CDM as

C = <Dc, Vc>

where Dc = { Stocks, Sensex , Time } are the dimensions and Vc = {price, volume} are the measures The hierarchical structure and the mapping function g of each dimension of cube C is defined above in section II. The cube C is initialized with the data at the lowest levels timestamp, symbol and senname of the dimensions Time, Stocks and Sensex at every thirty seconds interval respectively.

Example 1: Select the stocks within price band of 400 and 1000.

Here the restriction is applied to the measure price on cube C with a condition that the resultant cube has the stocks in the range of 200 to 400. The query can be expressed as

$C'=\sigma_{\text{price} \le 400\ \&\ \text{price} \ge 1000}\,(C)$

where σ is the Restrict predicate involves the condition applied on measure price.

Example 2: Select IBM and INF stocks of BSE sensex.

IBM and INF are the attribute values of symbol of dimension Stocks and BSE is the attribute value of senname of dimension Sensex. The selection when applied on the cube C results in a C'. The operation can be written as:

$C' = \sigma_{\text{Stocks.symbol=IBM}\ \&\ \text{Stocks.symbol=INF}\ \&\ \text{Sensex.senname=BSE}}\,(C)$

where the Restrict operator σ involves the selection on more than one dimension.

Example 3: Whether the stocks in IT category of BSE sensex are increasing for the chosen time period say 1minute?

The answer to this query is obtained using two step jobs. In the first step two Roll-up operations are performed. First Roll-up is performed on the Time dimension with aggregate function increase marking the stocks that are increasing. The second Roll-up is performed on the Stocks dimension for viewing the result at the category level rather than viewing it at the symbol level. The second step is to do the selection of stocks of BSE sensex using Restrict operator. The query can be expressed as:

$$C' = \sigma_{\text{Sensex.senname=BSE}}\ \underset{\text{Stocks.symbol}}{\overset{\text{Stocks.category}}{\rho}}\ \underset{\text{Time.timestamp}}{\overset{\text{Time.minute}}{\rho_{\text{increase}}}}(\text{price})\,(C)$$

Example 4: At what time did IBM peaked at the specified time period?

The output of the query is the time at which the IBM has the maximum share price in the specified time period. The price of IBM is checked at all times for the specified time period. The time at which IBM is high noted is marked and rest at all places the value for IBM is set NULL. There is no Roll-up in the Time dimension rather it only tells which set of values are involved in group from which the maximum is to be found. The query can be written as:

$$C' = V\,\underset{\text{Time.timestamp}}{\overset{\text{Time.hour}}{\rho_{\text{Peak}}}}(\text{price})\,(C)$$

VI. CONCLUSION

In this paper, the Cube Data Model is defined. This data model is used to define the basic data structure which is a cube. The cube can be analyzed using new OLAP operations. Dice and Roll-Up can be performed in our

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-4, Issue-ICCIN-2K14, March 2014

system. Normally, the hierarchy among attributes of a dimension allows one to perform roll up. We have used this hierarchy to define Virtual Roll-Up, which has lead to the definition of new non-aggregate functions -Peak and Nadir. Additionally, the three new aggregate functions, namely increase, decrease and fluctuate[9] can be used to analyze the movement of values with respect to time. The use of quantifiers with reference to database queries is not new [14,15,16]**.** By incorporating quantifiers in our OLAP queries, we have increased the querying capability of the system.

REFERENCES

[1] Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems”, In Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, pp.1-16, 2002.

[2] L. Golab and M. T. Ozsu, “Issues in data stream management”, ACM SIGMOD, vol-32(2), 2003, pp. 5-14.

[3] L. Golab, “Querying Sliding Windows over On-Line Data Streams”, Ph.D. Thesis, David R. Cheriton School of Computer Science, University of Waterloo.

[4] J. Gehrke, F. Korn, D. Srivastava, “On Computing Correlated Aggregates Over Continual Data Streams”, In Proc. ACM Int. Conf. on Management of Data, pp. 13-24, 2001.

[5] M. Riedewald, D. Agrawal, A. El Abbadi, “Flexible Data cubes for Online Aggregation”, In Proc. Int. Conf. on Database Theory (ICDT),pp.159-173, 2001.

[6] R. Motwani, J. Widom, A. Arasu, B. Babcock, S.Babu, M. Datar, G.Manku, C. Olston, J. Rosenstein, R. Varma, “Query Processing, Approximation, and Resource Management in a Data Stream Management System”, In Proc. 1st Biennial Conf. on Innovative Data Syst. Res., pp. 245-256 ,2003.

[7] Xuepeng Yin and Torben Bach Pedersen, “What Hierarchies can do for Data Streams?”, A DB Technical Report, Waterloo University, 2005.

[8] Parimala N, Bhawna, “On New Roll-up possibilities of Data Stream Warehouse”, In Proc. of Third IASTED International Conference, ACST,pp. 523-528, April 2007.

[9] Bhawna, Parimala N., Quantifiers for Data warehouse operations*, InProc. of Sixth ICICS International Conference on Information,Communications andSignal Processing*, (2007),Singapore, IEEE Catalog Number07EX1685C,ISBN 1-4244-0983-7.

[10] Parimala N, Bhawna,"Continuous Multiple OLAP Queries for Data Streams", International Journal of Cooperative Information Systems(IJCIS), June 2012, Vol. 21, No. 02 : pp. 141-164

[11] R. Motwani, J. Widom, A. Arasu, B. Babcock, S.Babu, M. Datar, G.Manku, C. Olston, J. Rosenstein, R. Varma, Query Processing,Approximation, and Resource Management in a Data Stream Management System, *In Proc. 1st Biennial Conf. on Innovative Data Syst. Res.,* (2003), pp. 245-256.

[12] S.Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V.Raman, F. Reiss,M. Shah,TelegraphCQ: Continuous Data flow Processing for an Uncertain World, *In Proc. 1st Biennial Conf. on Innovative Data Syst. Res,* (2003),pp. 269-280.

[13] Panos Vassiliadis, “Modeling Multidimensional Databases, Cubes and Cube Operations”, SSDBM, pp. 53-62, 1998.

[14] R.Agarwal, A.Gupta and S. Sarawagi. Modeling Multidimensional Databases. Int. Conf. on Data Engineering (ICDE’97). IEEE, 1997.

[15] Ping-Yu Hsu, Douglas Stott Parker Jr., "Improving SQL with Generalized Quantifiers", ICDE, 1995, pp. 298-305.

[16] Gyssens, M., Van Gucht, D., and Badia, A., "Query Languages with Generalized Quantifiers", In R.Ramakrishnan (Ed.), Application of Logic Databases, Kluwer Academic Publishers, 1995, pp. 235-258.

[17] Badia Antonio,"Cooperative Query Answering with Generalized Quantifiers", Journal of Intelligent Information Systems, vol- 12, 1999, pp75–97.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.*