View Design and Analysis for Fast Query Processing Using Mathematical Approach

[1]Poorva Rathore, [2] Pravin Malviya
Department of Computer Science & Engineering ,
Shri Balaji Institute of Technology and Management , Betul, MP, India
Email – poorvarathore05@gmail.com, malviyapravin2010@gmail.com

Abstract: *In this paper, we describe the effective design of materialized view selection and preservation in a data warehousing system. This design implementation aims to aid users in retrieving data effectively for business analysis. The skeleton design of this data warehousing system employs the dimensional modeling concepts of snowflake as well as star schemes. Here, some of frequently accessed queries are stored in various user files on which we apply materialized view selection process to create materialized views in order to minimize the query processing cost. A cost analysis model was developed to enable the estimate the total cost and benefit involved in selecting each materialized view. For effective materialized views selection and preservation methodology, MVSA and VPA algorithms have been implemented and results are shown below.*

This algorithm takes into account an effective cost variables associated with the materialized views Selection and preservation method which includes query access frequencies, materialized view access frequencies, query processing costs, materialized view access cost ,query storage cost , materialized view storage cost and the availability of the system's storage. The algorithm has been applied to dummy tables containing student information to create cost effective set of materialized summary views, , thereby resulting in an efficient data warehousing system where storage and query processing of the system is optimized.

Key Words: *View, View-selection, Access Frequency, storage cost, preservation, fast query processing*

1. INTRODUCTION:

A data warehouse is a repository that stores a large volume of extracted and summarized data for On-Line Analytical processing and decision support systems [10]. To reduce the cost of executing join queries in a data warehousing environment, frequently used join queries are often pre-computed and materialized into physical summary views so that future queries can utilize them directly. Without a doubt, materializing these physical summary views can minimize query response time. On the other hand, if the source data changes frequently, keeping these materialized views updated will certainly incur a high maintenance cost. In addition, for a system with limited storage space and/or with thousands of virtual summary views, we may be able to materialize only a small fraction of the views and preserve the created materialized view. Therefore, different parameters used to select and preserve materialized view which includes query access frequencies, materialized view access frequencies, query processing costs, materialized view access cost ,query storage cost , materialized view storage cost and the availability of the system's storage. views defined over distributed data sources are significant for many applications to ensure high Availability, efficient access and reliable performance This work emphasizes an efficient optimization of query processing with the help of materialized view over the data warehousing environment.

There are many advantages of prominent views section mechanism such as
• It decreased CPU consumption
• Obviously faster response times
• It required less physical reads (Base table read)
• Less writes
• Materialized Views offer us elasticity of basing a view on Primary key
• Users, Applications, Developers and others can take benefit of the fact that the answer has been already stored for them.
• In a read-only / read-intensive environment will provide reduced query response time and reduced resources needed to actually process the queries.

This paper is organized as follows. We describe a related work of materialized view selection and materialized view preservation in preservation section Views Selection and preservation framework implementation details is explaining in section given below.

2. LITERATURE REVIEW:

The difficulty of finding appropriate summary views to materialize for answering repeated queries has been studied under the name of materialized view selection methodology. Further created materialized views are needed to be preserve according to their access frequency and storage which has been studied under the name materialized view preservation .The major task is to maintain the created materialized view whenever base table information changes. All these methodologies are studied by various researchers and provide the well suited solution according to the environment in which materialized view are created, preserve and maintain.

The various researches proposed as well as implementation work o n materialized view selection ,preservation and maintenance are describe below.

Dr. T.Nalini et al. proposes an cost effective algorithm for the selection and maintenance of materialized views so that query evaluation costs can be optimized as well as storage cost was evaluated in this piece of work. [18]

Ashadevi, B and Balasubramanian developed framework for materialize view selection problem, which takes into account all the major cost metrics associated with the materialized views selection, including query processing frequencies, base relation ,update frequencies, query access costs, view maintenance costs and the system's storage space constraints and then selects the most efficient views to materialize and thus optimizes the maintenance, storage and query processing cost.[4]

Himanshu Gupta and Inderpal SinghMumick developed an algorithm to integrate the maintenance cost and storage constraint in the selection of materialized views for data warehouse environment [3].

Yang, J et al. proposed a heuristics algorithm based on individual optimum query information .This framework is based on specification of multiple views processing plan (MVPP), which is used to present the problem formally.[17]

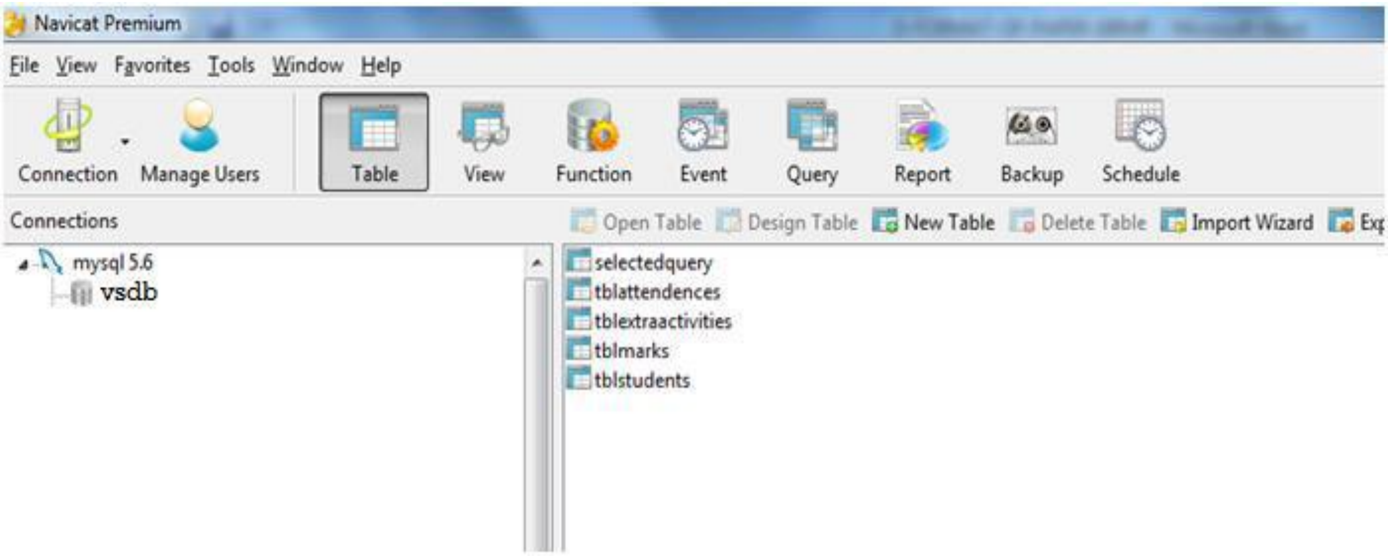
Harinarayan et al. developed an algorithm for the materialized views selection so that query processing cost can be optimized in the unique cases of "data cubes". This paper provides good trade-offs between the space used by the data cubes and the average time to answer query. Here, the costs for view maintenance and storage were not addressed in this piece of work.[16]

Amit Shukla et al. developed a very simple and fast heuristic algorithm, PBS, to select aggregates for pre computation. PBS algorithm runs faster than BPUS, and is fast enough to make the exploration of the time-space trade -off feasible during system configuration [15]

Wang, X et al. proposed view maintenance techniques which are classified into four major categories: self maintainable recomputation, not self-maintainable recomputation, self maintainable incremental maintenance and not self maintainable incremental maintenance. Self-maintainable Incremental maintenance performs the best in terms of both storage and number of rows accessed.[17]

3. EXPERIMENTAL RESULTS AND DISCUSSION:

Following tables are created in mysql database vsdb and its snap shot are shown in table 1. A different table contains different attributes these tables are initially empty but storing the information after suitable random record insertion in every table except selected query table.



Here we elaborates the running experiment results and their discussions that we have carried out using dummy database schema by applying view selection and view preservation algorithm. The bunch of queries are analyze using MVSA algorithm after pressing Analyze queries button where each query is analyze using three parameter i.e. query frequency, query processing time and query space . After finding all these three values for each query we calculate frequency cost, processing cost and storage cost to make every parameter value should be in one form that are shown in fig2 i.e in between 0 and 1 then apply formula for selection cost with impact weight 0.5, 0.3 and 0,2 respectively.

Formula: 0.5*sqrt(FrequencyCost)+0.3*sqrt(ProcessingCost)+0.2*(1-StorageCost)

Thereafter specific materialized view selection threshold is provided by the view analyzer to create useful materialized views that threshold value is sum of all selection cost value divided by number of selection cost.

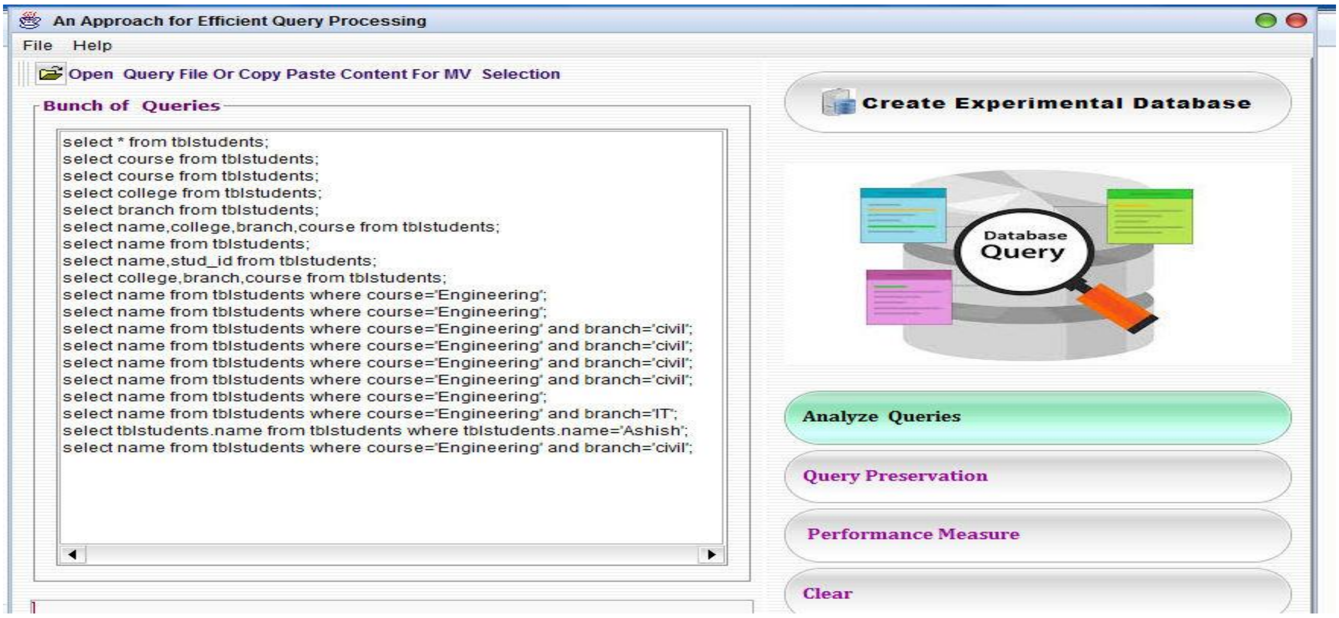
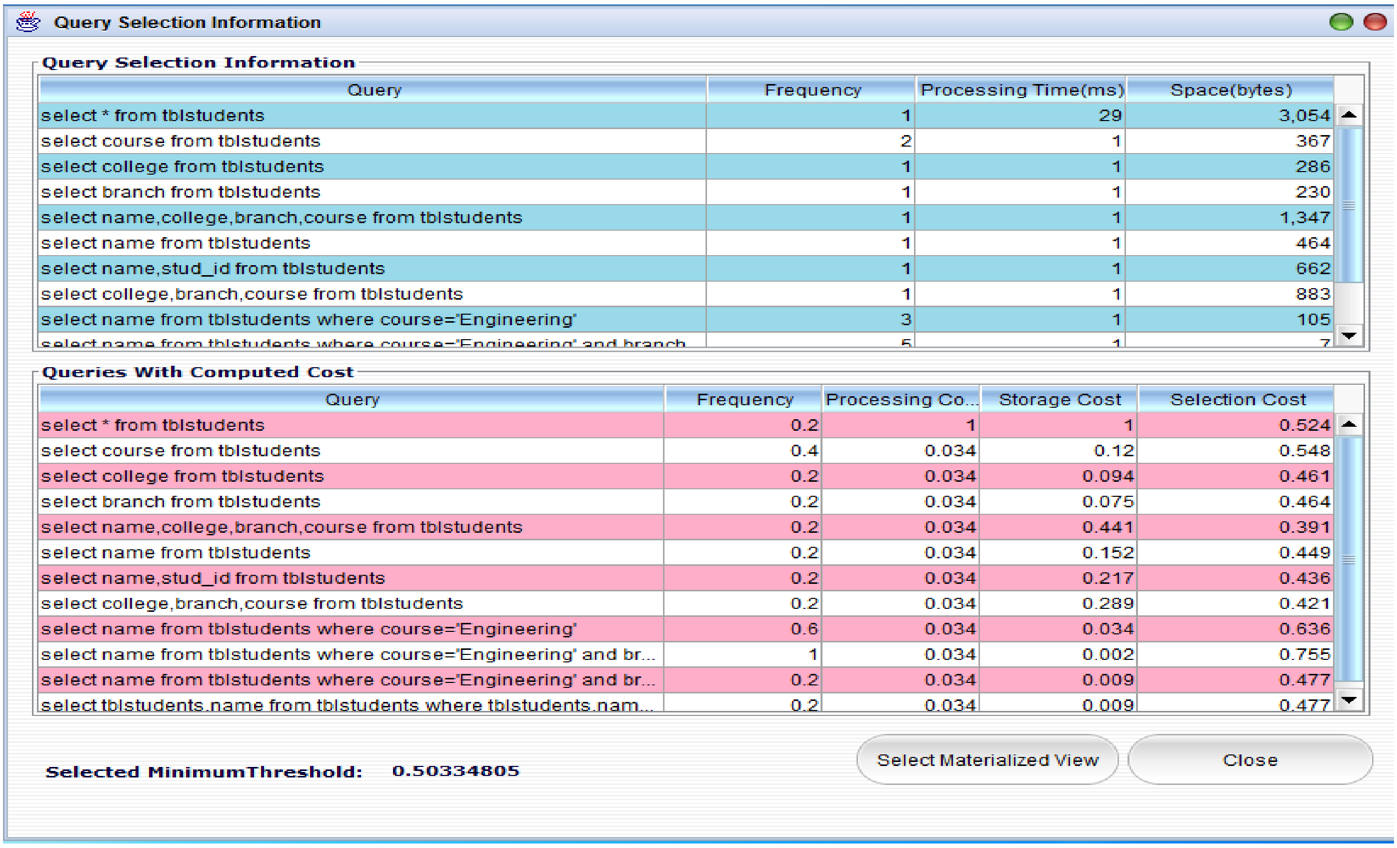


Fig1: Input Queries for View Selection Processs


Query Selection Information

Query Selection Information

Query	Frequency	Processing Time(ms)	Space(bytes)
select * from tblstudents	1	29	3,054
select course from tblstudents	2	1	367
select college from tblstudents	1	1	286
select branch from tblstudents	1	1	230
select name,college,branch,course from tblstudents	1	1	1,347
select name from tblstudents	1	1	464
select name,stud_id from tblstudents	1	1	662
select college,branch,course from tblstudents	1	1	883
select name from tblstudents where course='Engineering'	3	1	105
select name from tblstudents where course='Engineering' and branch	5	1	7

Queries With Computed Cost

Query	Frequency	Processing Co...	Storage Cost	Selection Cost
select * from tblstudents	0.2	1	1	0.524
select course from tblstudents	0.4	0.034	0.12	0.548
select college from tblstudents	0.2	0.034	0.094	0.461
select branch from tblstudents	0.2	0.034	0.075	0.464
select name,college,branch,course from tblstudents	0.2	0.034	0.441	0.391
select name from tblstudents	0.2	0.034	0.152	0.449
select name,stud_id from tblstudents	0.2	0.034	0.217	0.436
select college,branch,course from tblstudents	0.2	0.034	0.289	0.421
select name from tblstudents where course='Engineering'	0.6	0.034	0.034	0.636
select name from tblstudents where course='Engineering' and br...	1	0.034	0.002	0.755
select name from tblstudents where course='Engineering' and br...	0.2	0.034	0.009	0.477
select tblstudents.name from tblstudents where tblstudents.nam...	0.2	0.034	0.009	0.477

Selected MinimumThreshold: 0.50334805

Select Materialized View   Close

Fig 2 : Queries View Selection Information

After finding the view creation cost the next step is to identify the most promising views that need to be created for fast query processing which are shown in fig 3 where the specified materialized view threshold value is 0.5033.
The values above 0.5033 are mostly used queries for which views are created to improve query performance.

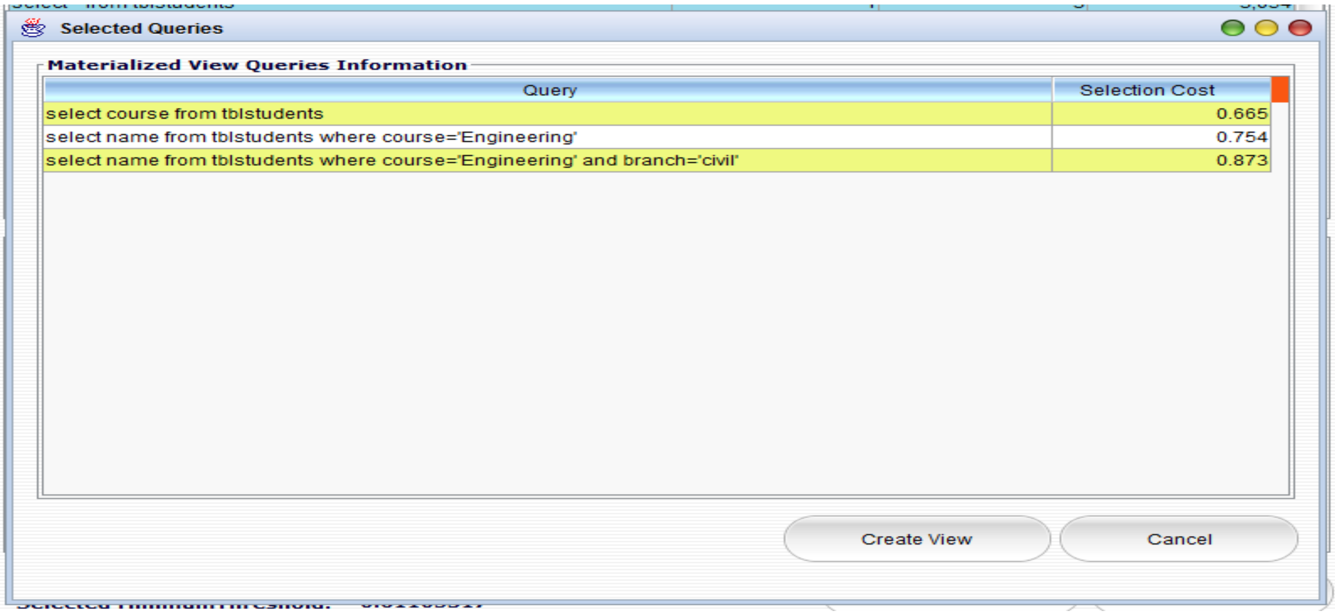


Fig 3: Selected view information for Fast Query processing

Here fig 3 shows only those views which satisfy the multiple purpose so here MVSA selecting only three views having selection cost is greater than the minimum materialized view selection threshold value from the bunch of input queries

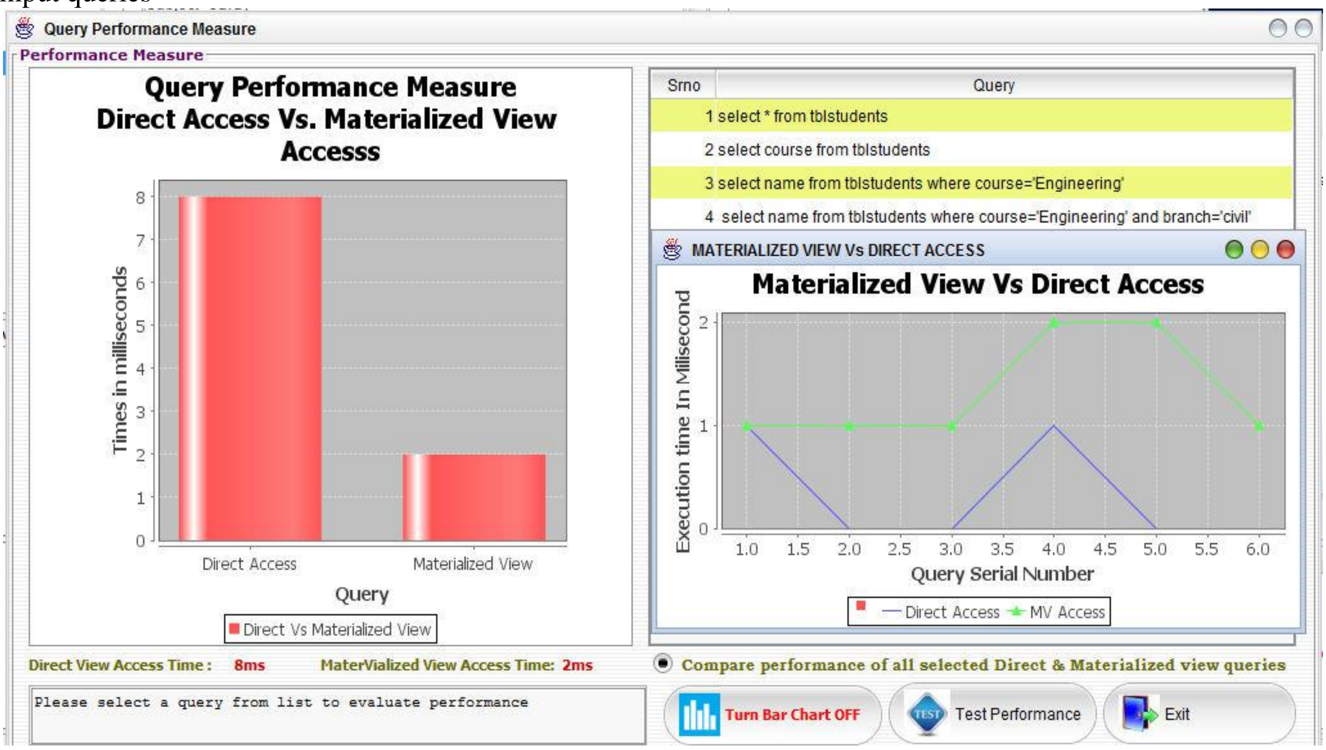


fig 4 shows comparison of view access time and direct query access time

fig 4 shows analysis of execution time of the query using specified materialized view selection framework as well as execution time of the query if it is executed on view of database without materialized view selection framework.

4. CONCLUSION:

Materialized view table store the precomputed result of the query which is used to improve query performance cost by minimizing query processing time. But to create all materialized summary view is next to impossible due to huge materialized view storage cost and duplication of unnecessary base table data. Therefore to select the set of most prominent materialized summary views is essential, so that user query performance increases and storage cost for storing materialized summary view decreases significantly.

This paper gives the idea regarding how to select a most important materialized view with the help of various major parameters like: frequency of summary views, processing cost of summary view and storage space. We have implemented the above design algorithm that determines which views are more valuable for the creation of materialized view so as to achieve the good query performance.

For experimentation, the design framework is executed on the dummy data warehouse model using list of summary views, to find the efficiency of the implemented approach in selection of materialized view. For future research in this area could focus on materialized view maintenance and validating this model against some real-world data warehouse.

REFERENCES:

1. Inmon, W.H., (1996). Building the Data Warehouse. 2nd Edn., John Wiley and Sons, Canada, ISBN: 0471-14161-5.
2. Jin-Hyuk Yang, and In-Jeong Chung, (June 2006), ASVMRT: Materialized View Selection Algorithm in Data Warehouse‘, International Journal of Information Processing Systems, Vol.2, No.2.
3. Gupta, H. & Mumick, I., Selection of Views to Materialize in a Data Warehouse. IEEE Transactions on Knowledge and Data Engineering, 17(1), 24-43, 2005.
4. B.Ashadevi, R.Balasubramanian,‖ Cost Effective Approach for Materialized Views Selection in Data Warehousing Environmental , IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.10, October 2008
5. Elena Baralis, Tania Cerquitelli, Silvia Chiusano,(April 2009) "IMine: Index Support for Item Set Mining", IEEE Transactions on Knowledge & Data Engineering, vol.21, no. 4, pp. 493-506, , doi:10.1109/TKDE.2008.180
6. Qingzhou Zhang , Xia Sun & Ziqiang Wang , ―An Efficient MA-Based Materialized Views Selection Algorithm ‖ IITA International Conference on Control, Automation and Systems Engineering, CASE 2009, pp: 315-318
7. Mr. P. P. Karde, Dr. V. M. Thakare. ―Selection & Maintenance of Materialized View and It‘s Application for Fast Query Processing: A Survey ‖. Proceedings of International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.2, November 2010
8. Ashish Mohod & Manoj Chaudhari.(June 2013) Efficient Algorithms for Materialized View Selection in Data Warehousing Environment‖ International Journal of Computer Science and Network, Volume 2, Issue 3,
9. Ashish Mohod & Manoj Chaudhari (April 2013): Improve Query Performance Using Effective Materialized view Selection and Maintenance: Survey‖ International Journal of Computer Science and mobile Computing, Vol.2, Issue4, pg.485 -490
10. Y.D. Choudhari and Dr. S. K. Shrivastava,(July 2012) Cluster Based Approach for Selection of Materialized Views ‖, International Journal of Advanced Research in Computer Science and Software Engineering ,Volume 2, Issue 7,
11. T.V. Vijay Kumar, M. Haider and S. Kumar,(2010) Proposing Candidate Views for Materialization, Communications in Computer and Information Science (CCIS), Volume 54, Springer Verlag, pages 89- 98,.
12. S. Joshi, C. Jeramine (March 2008) Materialized Sample Views for Database Approximation ‖ IEEE Transaction on Knowledge and Data Engineering, Vol. 20 Issue 3, .
13. Abdulaziz S. Almazyad, Mohammad Khubeb Siddiqui,― Incremental View Maintenance :An Algorithmic Approach ‖, International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 03.
14. C. Zhang, X. Yao, and J. Yang.(2001), An evolutionary Approach to Materialized View Selection in a Data Warehouse Environment. IEEE Transactions on Systems, Man and Cybernetics, vol. 31, no.3, pp. 282– 293,.
15. A.Shukla, P. Deshpande, and J. F. Naughton,(1998) Materialized view selection for the multi -dimensional datasets, in Proc. 24th Int. Conf. Very Large Data Bases, , pp. 488 –499.
16. V.Harinarayan, A. Rajaraman, and J. Ullman (1996) Implementing data cubes efficiently Proceedings of ACM SIGMOD 1996 International Conference on Management of Data, Montreal, Canada, pages 205--216,.
17. Dr.T.Nalini, Dr.A.Kumaravel, Dr.K.Rangarajan, (April 2012) “A Novel Algorithm with IM-LSI Index For Incremental Maintenance of Materialized View” JCS&T Vol. 12 No. 1.