

A Design and Implementation of Efficient Algorithm for Materialized View Selection with its Preservation in a Data Warehouse Environment

Sapna Choudhary¹, Roshna Mahajan²

¹Assistant Professor at Shri Ram Group of Institutions, Jabalpur, Affiliated to RGPV University, Bhopal, India

²Student of M.Tech- 4th semester at Shri Ram Group of Institutions, Jabalpur, Affiliated to RGPV University, Bhopal, India

Abstract: *Though, different approaches designed by the researchers to select views in order to materialized it in a data warehouse (DW) with aim of speed up to access the requisite information, most of having still a sort of limitations/constraints. In order to remove constraints/limitations, we have designed an efficient algorithm of MV selections with its preservation by considering the three operating parameters viz. processing time, query frequency and query area required. The performance of selected MV of any query with respect to those query selected directly from data warehouse was presented in term of access time required. The result shows that MVs access time of any queries was found to be very less as compared to that queries access directly from data warehouse. The results also presented that the performance of selected MV for any query in terms of access time should be outstanding if our data warehouse contains huge data. The preservation of selected MVs depended upon the area required and frequency of the query. Assigning the same frequency to all queries of selected MVs, then the preservation depended upon only area required to be stored by any query. The result shows the preservation of query also.*

Keywords: Data warehouse (DW), Materialized View (MV), Processing time, Query Frequency, Area, Performance and Preservation

1. Introduction

As we all aware, there are many factors that leads to competition in present market. Every business man faces the challenges in this present era in the market to run their business with maximum profits along with consumer satisfaction by their product. Any kind of business either fails or even faster leads as per the complexity or, with their ability to analyze the problem if any in the business and synthesize the information by using management information systems or simply information or data base management system they adopted. A data warehouse is a subject-oriented, integrated, time-varying, nonvolatile collection of data that is used primarily in organization decision making [1]. By adopting an update-driven approach (new emerging network services), a data warehouse system collects data from various data sources through communication networks both locally and internationally. By using DW system we can consolidate historical data for analysis and access such analysis by users to use locally and remotely.

Data warehousing enables easiness to any organization and maintenance of large data in addition to fast recovery and analysis in the desired manner and depth required from time to time. As the data size increases continuously, the speed requirements for processing and then analyzing the data by economic tools so as to understand the meaning of this data are also required to be increase significantly. Also the speed requirement of lane/ internet services/ network services are also needs faster in order to storing the new data in data base of data warehouse. Already stated, data warehouse contains large amount of data but not all data are required at a time by any firm or any user for specific purpose. The firm or any user access required data directly from data warehouse is unnecessarily time consuming. Knowing the time is the dominating factor for profit maximization of any business,

we needs to search such an alternative method to access data that required much less time. This alternative method is through selection of MV that require much less time for accessing any data as per required.

Before seen MV, let us define a view; a view is defined, the value of a given view at a given time is whatever results from evaluating certain relational expression at that time. i.e the definition of the view is store in the database system itself. Hence, in terms of base relations, a view is defined as derived relation. A view thus defines a function of derived tables (child table) from a set of base tables (parent table); this function is typically recomputed each & every time the view is mentioned.

Now let define MV; According to the definition of materialized views, the contents of the data warehouse are treated as a set of materialized views defined over the data base sources. One can designed the materialized views based on the user's requirements. The advantage by means of materialized views is significant since database access to the materialized view is just a cache, which is copy of the data that can be accessed quickly. Although there are number of benefits getting from materialized views creation, some important are its Integrity checking and optimization of query

In short, database store the result of query defining the view. Contrast this, a materialized view is a view, whose information is pre-calculated, with physically storing it and thus it is beneficial in many cases to access the information directly from materialized view rather than to compute the information of the view by running the query defining that view.

The main problem lies in front of the DW designer/administrator that which views is to be select first, in order to materialize it in the DW. Maintaining the materialized views for every query is impractical as the MVs is the physical table so disk-space requirements and hence consumption is quite large and/or large update cost having. Thus, the issue, constraint and challenges lies in-front of designer(those work in this field) is how to select such a bunch of query to materialized views created with optimal condition of parameters of processing time, area/space, and processing frequency. In this connection, constraint of the disk-space identify the availability of the disk-space in a data warehouse, whereas constraint of the maintenance-cost represents how long all views must be updated, because changes to the source data result in re-computing the materialized views accordingly, which will be periodically done in a time window. The materialized view creation and selection is based on the parameters defined combination of which depended upon the selection cost and threshold value. The optimal condition of parameters of low processing time, low area/space, and high frequency are desirable in order to create MVs.

2. Research Methodology

Research methodology is the essential part of research from which we achieved our predefined objectives with highlighted on research problem.

2.1 Problem Formulation

The main problem lies in front of the DW designer/administrator that which views is to be select first, in order to materialize it in the DW. Maintaining the materialized views for every query is impractical as the MVs is the physical table so disk-space requirements and hence consumption is quite large and/or large update cost having. Thus, the issue, constraint and challenges lies in-front of designer(those work in this field) is how to select such a bunch of query to materialized views created with optimal condition of parameters of processing time, area/space, and processing frequency. In this connection, constraint of the disk-space identify the availability of the disk-space in a data warehouse, whereas constraint of the maintenance-cost represents how long all views must be updated, because changes to the source data result in re-computing the materialized views accordingly, which will be periodically done in a time window. The materialized view creation and selection is based on the parameters defined combination of which depended upon the selection cost and threshold value. The optimal condition of parameters of low processing time, low area/space, and high frequency are desirable in order to create MVs.

2.2 Pre-requisite for proposed work

As the steps for carrying work are random access record generator, creation of data warehousing, selection cost of queries, selection of MVs, and at last preservation of selected MVs. And the work carried on Java (JDK 1.6), my sql 5.1 and Net beam 6.0, we needs to know the knowledge for the same and needs to install on our system.

A. NetBeans

NetBeans written in Java is a software development platform. It also supports other languages, like PHP, C/C++ and HTML5. Netbeans is an open source integrated development environment. NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM. NetBeans IDE supports development of all Java application types. Netbeans profiler is a tool for the monitoring of java application

B. Java

In software development, the Java a programming language called as object oriented programming language derived from C++

Java is often Just-in-time compiled at runtime by the Java Virtual Machine, but may also be compiled ahead-of-time, just like C++

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

C. MySQL

MySQL(My Structured Query Language), an open-source relational database management system (RDBMS) is a popular choice of database for use in web applications, and written in C and C++. Its SQL parser is written in yacc, MySQL can be built and installed manually from source code.

2.3 Random Access Record Generator

Programmings were done on the front end java and back end my sql in order to create random record. Our data base contains the four tables naming orderinfo, supplierinfo, supplier productinfo, tblcustomer. The orderinfo contains four attributes including order id, cust id, product name, date. The supplierinfo contains seven attributes including supplier id, first name, middle name, last name, address, tel number. The supplierinfo contains eight including attributes supplier id, product name, date, nos. of items, unit price, total, amount paid, amount balance. The tblcustomer contains eight attributes including srno., cust:-id, middle name, last name, company name, address, tel no., time stamp. The first phase of materialized view selection is generation of huge random set of record for the above given database tables using random data insertion record generation. After that all possible set of complex queries are generated on above created records.

2.4 Creation of data warehousing

The data base of any information are created and stored in data warehousing. The data warehousing contains the large number of data that needs to be access as per needs of user. Thus, the data warehousing are created by creation of number of data. These programmes run on front end java and back end mysql.

2.5 Selection cost of queries

Keeping in the views up-to-date in response to the changes in the source data, several algorithms have been proposed by researchers in recent times. The presented algorithms have a sort of limitation/constraints. Our strategy for arriving at selection cost of queries are based on certain operating parameters viz. processing frequencies, processing time and area required by each query. Each parameter is multiply with the suitable weighted factor to arrive at parameter's cost. Then after adding all parameter's cost we arrived at selection cost. Designed formula to be used in order to find the selection cost (S.C) is given by:

$$SQ = \text{Weight1} * QFC + \text{Weight2} * (1 - \text{Area Cost}) + \text{Weight3} * \{1 - QPC\}$$

Where QFC is the frequency cost that can be calculated as frequency of particular query/ max frequency from all queries

Area is calculated as multiplication of row and column of query and A.C is area cost that can be calculated as area to be selected of particular query/ maximum area from all queries.

QPC is the processing cost that can be calculated as processing time of particular query/maximum processing time from all queries

Where, Weight1, Weight 2 and Weight 3 are Weights given to QFC , Area Cost, and QPC such that their sum is equals to 1

2.5.1 Selection of MVs from different selection cost of queries

Queries directly fired to DW are critical with regards of their complexity and length. Generations of materialized views are the alternative solution to provide the better performance for DW queries. However, these views have constraints of maintenance cost, so it is not possible to materialize all views. An important challenges lies in environment is view selection cost to be materialized because we have to realize the trade-off between performance and view maintenance. This method can be resolved by adopting clustering method. In this method similar queries will be clustered based on their query frequency in order to select the views to be materialize. By doing this way, the execution time and space can be reduced. The processing is done on the way that when the query is fired, it will be firstly compared with already available clustered or existing query, and then pre-computed MV will be returned as a result of it, it will reduce the execution time of the query.

In order to remove the problems of preservation and maintenance of created MV, only those created MVs are selected for which, S.C is greater or equal to threshold value.

The threshold value T.M. is calculated by the sum of S.C of individual query/total number of query. Thus, the required condition is S.C is greater than or equal to T.M, for the view to be materialized

2.5.2 Algorithm for materialized view selection

The design algorithms are presented here for materialized view selection

Algorithm 1: This algorithm is design to processing frequencies, processing time, and area as required by different queries.

The assumptions involved are

Assumptions:

GQB → Given Bunch of queries
 QF → Query frequency
 QAV → Query Area value
 QPT → Query processing time
 QIDL → Query information detail list
 QDD → Query detail DTO
 begin:
 Repeat for I ← 1 to GQB
 QDD ← find QF ;
 QDD ← find QAV ;
 QDD ← find QPT ;
 QIDL ← QDD ;
 end repeat
 end

Candidate frequency, processing time and space of queries are found out by given algorithm1. For different queries present in query set (GQB), we have to find out the frequency, processing time and space. This will be stored in query information detail list (QIDL) which in the form of query detail DTO (QDD).

Algorithm 2: This algorithm is design to find out processing frequencies, processing time, and area cost (selection cost) required by queries and based on selection cost and threshold value, MVs are build

Assumptions:

QMFreq → Maximum query frequency
 QS → Query
 QMPT → Maximum query processing time
 QMA → Maximum Query Area Value
 QPC → Query processing cost
 QSC → Query area cost
 QFC → Query frequency cost
 QCT → Query cost table
 SQ → Query selection cost
 MT → Minimum threshold
 N → Number of rows in query cost table
 NQIC → Number of rows in query information detail list
 Weight1, Weight2 & Weight3 → Weighted constant values
 Where
 Weight1+Weight2+Weight3=1;
 begin:
 Repeat for I ← 0 to NQIC -1
 QDD ← QIDL [i]
 QF ← QDD
 QPT ← QDD
 QAV ← QDD
 QFC ← QF/QMFreq ;
 QPC ← QPT / QMPT ;
 QSC ← QAV/ QMA ;
 QCT ← QFC ;

```

QCT ← QPC ;
QCT ← QSC ;
end repeat
[Find selection cost]
Repeat for I ← 1 to QCT
SQ = Weight1 * QFC + Weight2 *(1-Area Cost)+Weight3 *
{ 1- QPC}
QCT ← SQ ;
end repeat
[Select MV Selection Threshold]
MT = ∑Ki=1 SQ / N
[Select materialized view having good query response,
High frequency, low processing, and area cost]
Repeat for i ← 0 to N-1
SQ ← QCT [i]
If (SQ >= MT ) then
Build the materialized view for the selected query
else
    Discard the query
end repeat
    
```

The output of algorithm1 i.e. QIDL is used as input in algorithm 2 in order to find out frequency, query processing and query area cost of each queries and hence by assigning suitable weighted factor and combining all three factor we arrived at selection cost of query.

The next step is using summation of all the selection cost divided by number of selected queries to find the appropriate materialized view selection threshold value. If the materialized view selection threshold value is greater than selection cost in that case query is discarded otherwise build the materialized view for the selected.

2.6 Preservation of existing MVs

As due to huge area requirements, not all created MVs are preserved. The preservation of selected MVs is based on some logistic approach based on processing frequencies and area required and hence cost of the same of any queries but not depended upon processing time cost.

2.6.1 Selection cost of MVs queries and its preservation

Before selecting new views for materialization, the existing materialized views are sustained based on their access frequency and space. In designing the formulae for calculating the selection cost of MV queries, no weighted is given to time. And hence selection cost of MV queries is depending upon two operating parameter viz. access frequencies and space required and hence depending upon the cost of the same parameters.

So, the design formulae are presented here are
 $S.C = MVPQ = \alpha * MVFC + \beta (1 - MVSC)$;
 $\alpha + \beta = 1$, where α, β are the weighted factor given to MVFC (MV frequency cost) and MVSC (MV space cost)
 $S.C = MVPQ$ are the selection cost of MV query
 Only those created MVs are preserved for which, S.C is less than threshold value.

The threshold value T.M. is calculated by the sum of S.C of individual query/total number of query of MVs.
 Thus, the required condition is S.C is less than to T.M, for the MV to be preserved

2.6.2 Algorithm for preservation of existing MVs

The steps for the preservation process are given in preservation algorithm.

Assumptions

ALMV → Array List of Materialized views

MVAF → Materialized view access frequency

MVSR → Materialized view area requirement

MVPT → Materialized view preservation threshold

MVPCS → Summation of materialized view preservation cost

MVPQ → Materialized views preservation cost

MVSC → Materialized view area cost

MVFC → Materialized view access frequency cost

α, β → Weighted constant values

For each Materialized View in ALMV

$MVFC = MVAF / \text{Maxi } MVAF_i$

$MVSC = MVSC / \text{Maxi } MVSC_i$

$MVPQ = \alpha * MVFC + \beta (1 - MVSC)$;

$MVPCS = MVPCS + MVPQ$;

end for

$MVPT = MVPCS / \text{Size of ALMV}$;

For each Materialized View in ALMV

if ($MVPQ < MVPT$) then

Remove current Materialized view;

end if

end for

3. Implementation and Results

3.1 Tables created in mvdb

The snap shot of table created in mvdb 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 particular table.

Table 1: Tables stored in mvdb

```

mysql> SHOW TABLES;
+-----+
| Tables_in_mvdb |
+-----+
| orderinfo      |
| selectedquery  |
| supplierinfo   |
| supplierproductinfo |
| tblcustomer    |
+-----+
5 rows in set (0.01 sec)
mysql>
    
```

3.2 Record insertion

The snap shot of input record insertion are presented in figure 1. It showing four insertions, viz. insert customer, insert supplierinfo, insert supplierproductinfo, insert orderinfo. One of the example for record insertion in orderinfo are presented in table 2

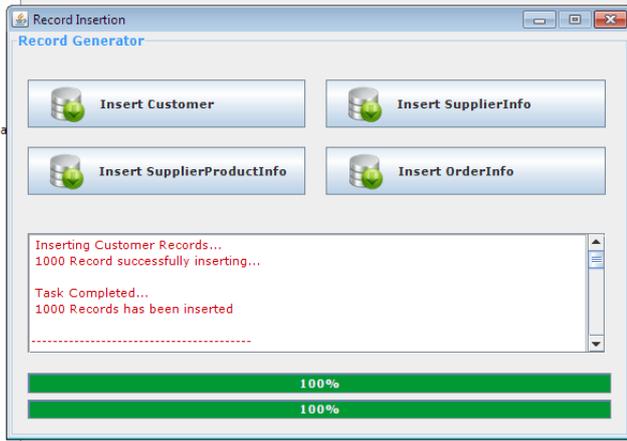


Figure1: Record insertion in different tables stored in mvdb

```
5 rows in set (0.00 sec)
mysql> select * from orderinfo;
```

order_id	cust_id	productName	orderdate
101	101	Keyboard	2013-02-23 00:00:00
102	102	Keyboard	2014-02-11 00:00:00
103	103	Harddisk	2014-05-16 00:00:00
104	104	Keyboard	2013-05-01 00:00:00
105	105	Sony	2013-02-07 00:00:00
106	106	CD-Rom	2011-01-23 00:00:00
107	107	Sony	2013-06-19 00:00:00
108	108	Mouse	2013-09-16 00:00:00
109	109	Harddisk	2012-10-07 00:00:00
110	110	Chip	2014-09-11 00:00:00
111	111	Chip	2013-11-21 00:00:00
112	112	Sony	2011-08-12 00:00:00
113	113	Mouse	2012-02-04 00:00:00
114	114	Mouse	2014-10-21 00:00:00
115	115	Keyboard	2014-11-04 00:00:00
116	116	Sony	2013-04-23 00:00:00
117	117	CD-Rom	2011-10-11 00:00:00
118	118	Mouse	2011-09-04 00:00:00
119	119	Chip	2013-09-22 00:00:00
120	120	Mouse	2010-08-25 00:00:00
121	121	Mouse	2013-01-13 00:00:00
122	122	CD-Rom	2011-06-02 00:00:00
123	123	Harddisk	2010-04-02 00:00:00
124	124	Chip	2013-09-20 00:00:00
125	125	Harddisk	2014-02-09 00:00:00
126	126	Keyboard	2011-04-09 00:00:00
127	127	Mouse	2011-10-23 00:00:00
128	128	Chip	2011-10-24 00:00:00
129	129	Harddisk	2013-04-07 00:00:00
130	130	CD-Rom	2011-02-26 00:00:00
131	131	CD-Rom	2012-09-12 00:00:00
132	132	CD-Rom	2014-09-12 00:00:00
133	133	Sony	2010-10-18 00:00:00
134	134	Harddisk	2012-09-05 00:00:00
135	135	Keyboard	2012-11-01 00:00:00
136	136	Mouse	2012-02-20 00:00:00
137	137	Mouse	2012-10-02 00:00:00

Table 2: Inserting result of orderinfo records in orderinfo table

3.3 Input query for materialized view creation process

Snap shot of some input query earlier recorded and presently required in different insertion tables in materialized creation process are presented in figure 2. It access the selection cost of different query required.

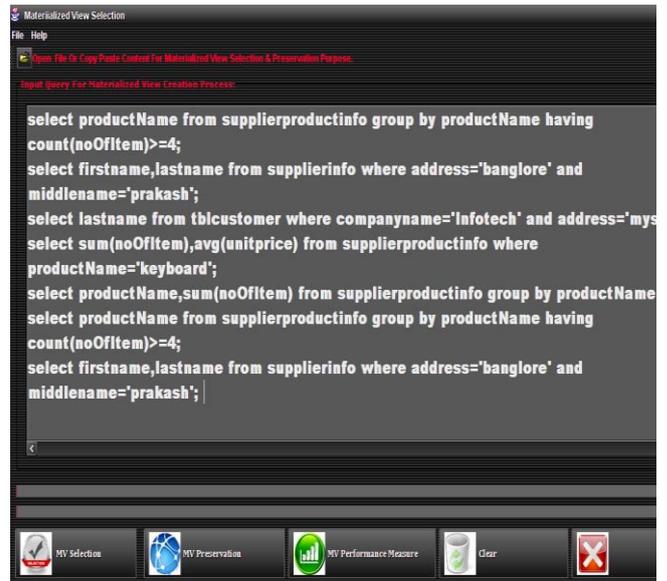


Figure 2: Input query fired for Materialized view creation process

3.4 Input query information (selection cost) of queries and its analysis for materialized view creation

The parameters along with its cost including frequency and frequency cost, query area and query area cost, processing time and processing time cost are presented in snap shot shown in figure 3 for different queries. And hence the selection costs of bunch of queries are presented. It also present the threshold value required above the value of which M.V created.

S/no	Query	Frequency	Frequency Cost	Query Area	QueryArea Cost	Processing Time	Processing Cost	MV Cost
1	select productName from supplierproductinfo group by...	2	1	6	0.033	16	0.516	0.885
2	select firstname,lastname from supplierinfo where addre...	2	1	182	1	16	0.516	0.597
3	select lastname from tblcustomer where companyname...	1	0.5	57	0.313	15	0.484	0.559
4	select sum(noOfItem),avg(unitprice) from supplierprod...	1	0.5	2	0.011	16	0.516	0.643
5	select productName,sum(noOfItem) from supplierprodu...	1	0.5	12	0.066	31	1	0.652

Default Threshold Value: 0.6433250620237835

Figure 3: Selected query information and its MV cost

Entering the calculated threshold value and comparing with materialized cost of each query and then the queries having the selection cost greater than or equal to threshold value are materialized and presented in snap shot given in figure 4(a) & 4(b).

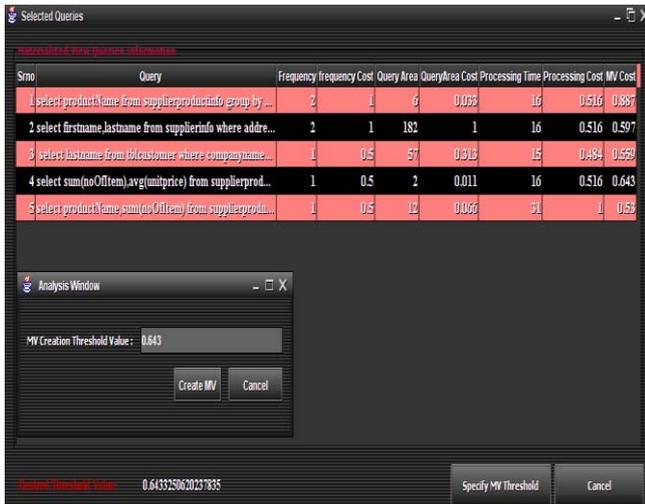


Figure 4(a): Selected query information and its MV cost along with screen of MV creation Threshold value

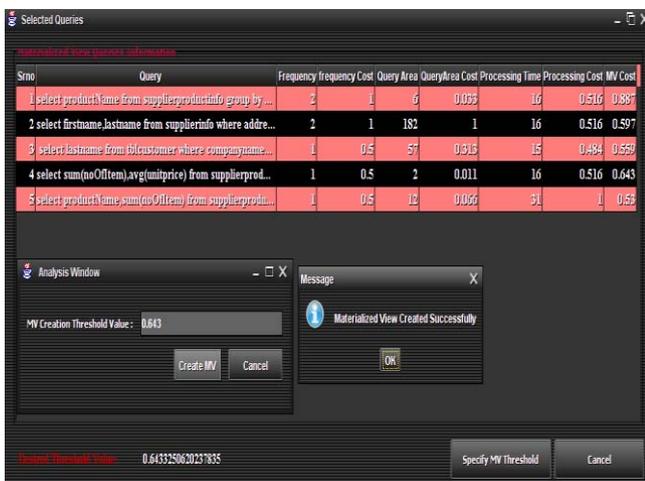


Figure 4(b): Selected query information and its MV cost along with screen of MV creation Threshold value & MV created successfully

The performance of query by direct access and access through MV are presented in snap shot given in figure 5



Figure (5): The performance of query by direct access and access through MV

From the performance any one judge the efficiency of MV queries over the queries directly access through data warehouse. The response time of queries through MVs access were found 15 milli-seconds while through direct access queries were found 32 milli-seconds. And hence efficiency of queries through MVs access over direct access

$$\text{Efficiency} = (\text{Direct access of queries} - \text{MV's access of queries}) / \text{MV's access of queries} \times 100$$

$$= (32-15)/15 \times 100$$

$$= 113.33\%$$

i.e. 113.33 % more efficient than access over direct access

3.5 Preservation of selected MVs

As, already discussed the preservation of all selected MV are not feasible due to area constraints. So, those MV views having high frequencies and required low area are preserved for long run. The snap shot for initial screen of selected MV view information with initial zero frequencies are presented in figure 6

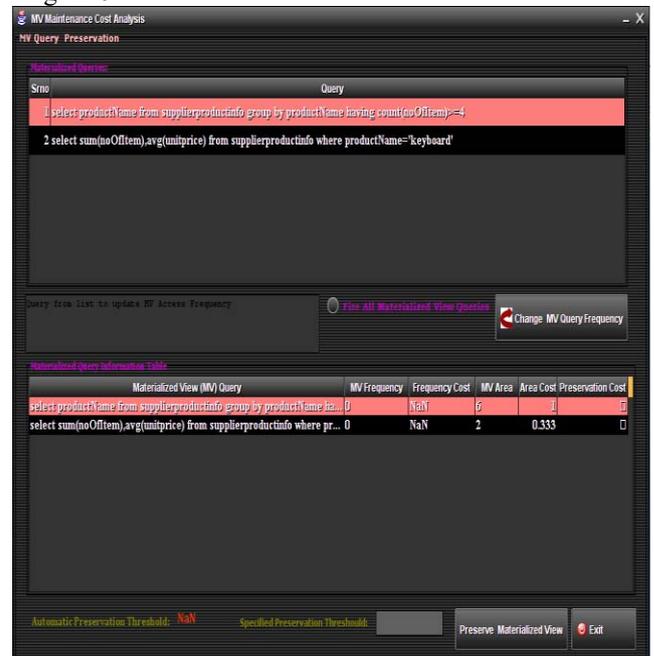


Figure 6: Initial screen of selected MV view information with initial zero frequencies

The frequencies and areas are the main domain in order to preserve it. We know the frequency of any query. Change the frequency as per known data. In our case we have assign uniform frequency for all queries of MVs and then preservation is based on the area only. The snap shot of screen of selected MV view with variable frequencies along with preservation cost are presented in figure 7(a)

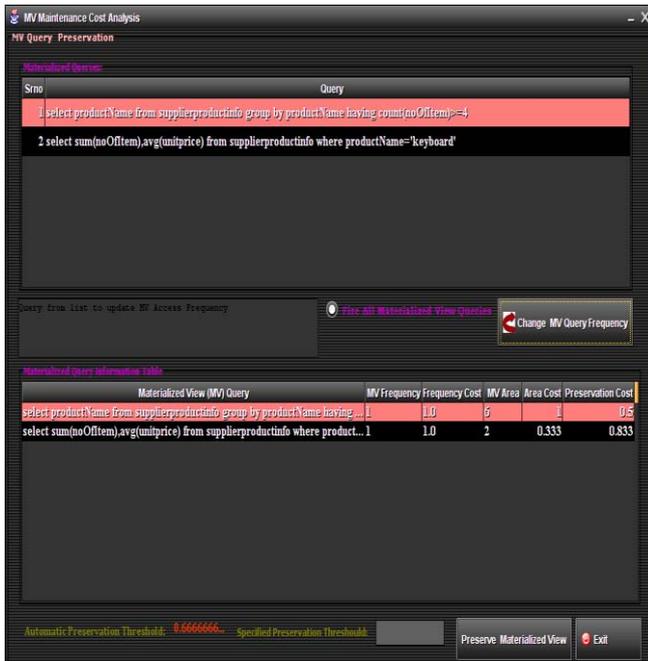


Figure 7(a): Initial screen of selected MV view with variable frequencies along with preservation cost

Specify the threshold value, and then the preservation of those queries having the preservation cost are less than threshold value are deleted. The remaining all selected MVs are preserved for future use. The snap shot of screen of selected MV view along with preservation cost and threshold value are given in figure 7(b) and 7(c)

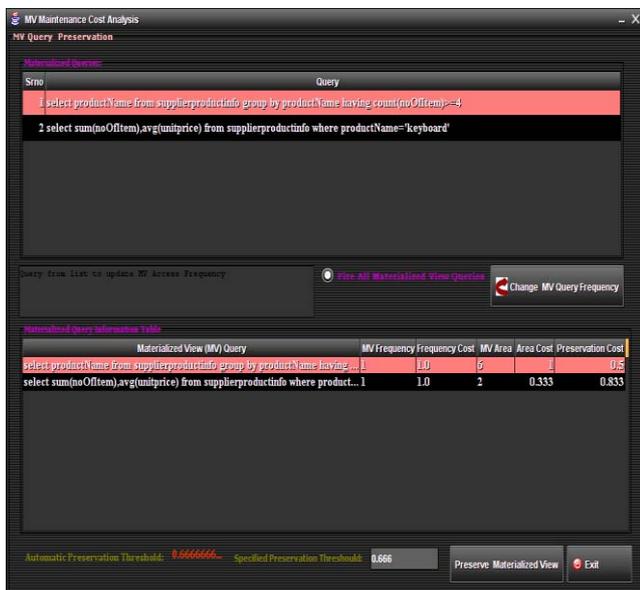


Figure 7(b): Screen of selected MV view along with preservation cost and threshold value

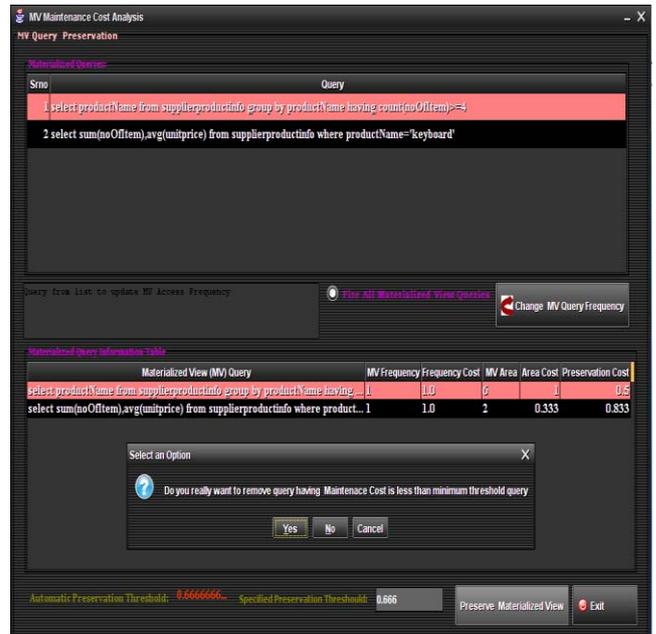


Figure 7(c): Screen of selected MV view along with preservation cost and threshold value with message select an option

Finally, the preserved MVs query having cost greater than and equal to threshold value are presented in snap shot given in figure 8. In this regards, the preservation cost of preserved queries are recalculated and based on only preserved queries

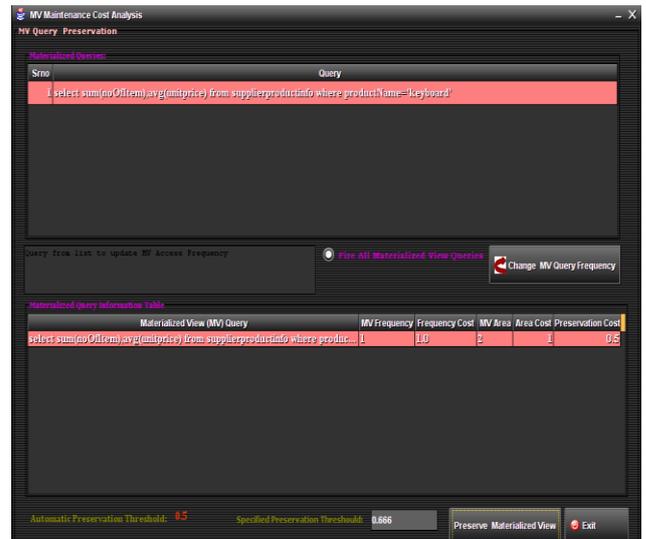


Figure 8: The preserved MVs query having cost greater than and equal to threshold value

4. Conclusions and Future Work

4.1 Conclusion

Carrying literature review predicts us the focus of all researchers' lies on how to optimize the materialized view selection. Some papers also focus on the maintenance/preservation of selected materialized views. As the materialized view store the pre-computed data it is used to improve query performance, by minimizing query processing time. As due to space, time constraints, this is not possible to select the MV for all fired queries, this needs to select the optimized condition in order to select the

materialized view from the bunch of fired queries. The optimized condition arrived only with have high frequency and hence high frequency cost, low processing time and hence low processing cost and low space- area and hence low area cost required by queries storing the data by selected view to materialized it.

In this work we have presented the methodology that determines which queries are more beneficial for the creation of materialized view so as to achieve the high query performance. No doubt, the selection of query depends upon the user need bases that changes the weighted factor given to different parameters. The framework in order to preserve the selected MV for long last usage is also given in this work.

For experimentation, the designed framework is executed on the customer data warehouse model using list of query, to find the efficiency of the designed approach. The performance of selected MV can be measures in term of access time required by each query with respect to query directly fired from DW. Beyond doubt, the time required of query by selected MV as compared to that query directly selected by DW is very much less. As, the population of DW increases, the time require of queries by selected MV is decrease very much. One of the result shows that the performance of queries in terms of access time from materialized view selection was found to 15 milliseconds while those queries directly selected from data base/data warehouse was found to 32 milli-seconds. This show the performance of query through materialized views access is 113.33 % better than those directly access through data base/data warehouse

The implemented framework will able to develop a highly efficient multi objective approach for materialized view selection and then preservation using a mathematical model.

4.2 Future Scope

The maintenance works of data warehouse are not done here which provides the future scope of work for those working in this field. Also, these designed algorithms can be adopted for real life world by utilizing the data of any company/organization/firm. i.e for future research in this area could focus on validating this model against some real-world data warehouse systems.

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.Balalabramanian, " Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment", IJCSNS International

- Journal of Computer Science and Network Security, Vol.8 No.10, October 2008
- [5] Elena Baralis, Tania Cerquitelli, Silvia Chiusano, "IMine: Index Support for Item Set Mining", IEEE Transactions on Knowledge & Data Engineering, vol.21, no. 4, pp. 493-506, April 2009, 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 (IJCSSES) Vol.1, No.2, November 2010
- [8] Ashish Mohod & Manoj Chaudhari. —"Efficient Algorithms for Materialized View Selection in Data Warehousing Environment" International Journal of Computer Science and Network, Volume 2, Issue 3, June 2013
- [9] Ashish Mohod & Manoj Chaudhari: —"Improve Query Performance Using Effective Materialized view Selection and Maintenance: Survey" International Journal of Computer Science and mobile Computing, Vol.2, Issue4, April 2013 pg.485-490
- [10] Y.D. Choudhari and Dr. S. K. Shrivastava, —"Gister Based Approach for Selection of Materialized Views", International Journal of Advanced Research in Computer Science and Software Engineering , Volume 2, Issue 7, July 2012
- [11] T.V. Vijay Kumar, M. Haider and S. Kumar, —"Proposing Candidate Views for Materialization, Communications in Computer and Information Science (CCIS), Volume 54, Springer Verlag, pages 89- 98, 2010.
- [12] S. Joshi, C. Jeramine —"Materialized Sample Views for Database Approximation" IEEE Transaction on Knowledge and Data Engineering, Vol. 20 Issue 3, March 2008.
- [13] Abdulaziz S. Almazayad, 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,," 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, 2001.
- [15] A. Shukla, P. Deshpande, and J. F. Naughton, —"Materialized view selection for the multi-dimensional datasets," in Proc. 24th Int. Conf. Very Large Data Bases, 1998, pp. 488–499.
- [16] V.Harinarayan, A. Rajaraman, and J. Ullman.—"Implementing data cubes efficiently". Proceedings of ACM SIGMOD 1996 International Conference on Management of Data, Montreal, Canada, pages 205--216, 1996.
- [17] Yang, J., Karlapalem. K., and Li. Q. (1997). A framework for designing materialized views in a data warehousing environment. Proceedings of the

Seventieth IEEE International Conference on Distributed Computing systems, USA, pp:458.

- [18] Samir ANTER Ahmed ZELLOU Ali IDRI. —Personalization of a hybrid integration system” creation of views to materialize based on the distribution of user queries 2012 IEEE.
- [19] T.V. Vijay kumar and kalyani devi . —frequent queries identification for constructing materialize views” 2011 IEEE.
- [20] Jeffrey Xu Yu, Xin Yao, *Fellow, IEEE*, Chi-Hon Choi, and Gang Gou. —Materialized View Selection as Constrained Evolutionary Optimization”. IEEE transactions on systems, man and cybernetics—Part C: applications and reviews, Vol. 33, no. 4, November 2003
- [21] Gang Gou, Jeffrey Xu Yu, and Hongjun Lu. —A Search: An Efficient and Flexible Approach to Materialized View Selection.” IEEE transactions on systems, man, and cybernetics—Part C: applications and reviews, Vol. 36, no. 3, may 2006 411
- [22] Chandrashekhar A. Dhote, Dr. M. S. Ali. —Materialized View Selection in Data Warehousing” International Conference on Information Technology (ITNG'07)
- [23] Sanjay Agrawal, Surajit Chaudhuri Vivek Narasayya. —Automated Selection of Materialized Views and Indexes for SQL Databases”

Author Profile

Mrs. Sapna Choudhary received the Bachelor degree and Master degree in Computer Science and Engineering department in 2004, & 2008 respectively from RGPV University, Bhopal, M.P. India. She is currently working as Assistant Professor in department of computer Science and Engineering at Shri Ram Group of Institutions, Jabalpur, affiliated to RGPV University, Bhopal, India



Ms Roshna Mahajan received the Bachelor degree in Computer Science and Engineering department from H.C.E.T, college, Jabalpur, RGPV University, Bhopal, M.P. India in 2006. I am presently doing my master degree in the same Engineering department from S.R.G.I, college, Jabalpur in the same university.

During the period of 2006-2011, I have a teaching experience in different polytechnic college of M.P. state