IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-2, Issue-9 E-ISSN: 2347-2693

Giving Future Vision to IR: A Query Clustering Approach

Gaurav Dubey[1], Romina Nayak[2*], Neha Wadhwa[3], Dr. Ajay Rana[4]
[1,3,4]Amity University, Noida, India
[2*]Kellton Tech Solutions Ltd., Gurgaon, India

www.ijcaonline.org

Received: Aug/21/2014 Revised: Sep/04/2014 Accepted: Sep/17/2014 Published: Sep/30/2014

Abstract— Information Retrieval (IR) has become very tedious given the amount of data handled these days. Search engines are posed with an ever increasing responsibility of giving precise responses to user queries in minimal time. In this paper, we present a query clustering approach which identifies Frequently Asked Questions (FAQs) for answering future queries. The proposed approach is based on identification of distinct subjects from queries enquired& logged in the past. The queries falling under each of the subject category are then reduced to a group which represents the frequently asked queries. In the past, these queries have been asked frequently & thus have an inclination of being repeated in the future. This will give the interface (e.g. search engines) an ability to predict future queries and respond in a time efficient manner. We extend this approach on a Real Estate data warehouse which proves its viability and efficiency in Real Estate domain as well.

Keywords**—** Data Warehouse, Information Retrieval, Query Clustering, Apriori, Subject Area Identification

I. INTRODUCTION

The IR for providing an answer to a simple user query might sometimes be very complex. This may also require multiple searches on the information storage. The time required for this IR increases with increasing complexity of the query. This has made the problem of IR a prospective research subject for people interested in data mining.

There are two basic methods of data mining: Query-driven (lazy, on-demand) approach and Warehouse (in-advance) approach. [31] The former is a traditional research approach in which a piece of information is extracted only if the user query demands for it. Disadvantages of this approach are:

- High response time due to slow or unavailable information sources and complex filtering and integration.
- Inefficient and potentially expensive for frequent queries.
- Competes with local processing at sources.
- Hasn't gained popularity in industry.

The latter approach is based on data warehousing. [31] In this, information is collected and combined in a central repository known as a data warehouse. [12] Data warehouse stores the information in advance according to previously posted queries. [12] This helps in a better decision making for answering similar questions in the future. This query processing takes a lot of time when a huge amount of information is being handled by the data warehouse. As a result, response time is high. Many organisations are working towards decreasing this response time to a minimal and helping in better decision making.

Many solutions have been devised to invent a better research approach. Each of them has their advantages and disadvantages. The constant need for improvisations in previous research approaches has led us to give way to this research paper. [18] In this work, our approach is to cluster the queries using user logs i.e. identifying similarity in user queries from previously posed queries.[16] The important task of discovering this similarity is done by identifying a common interest (subject) amongst the queries in the user query log.[10,20] This is known as Subject Area Identification. The clusters so obtained facilitate the decision making for identifying user's FAQs which are accumulated and cached. This helps the search engines or other question answering systems to respond to similar queries accurately and bring down the response time effectively.

Corresponding Author: *Romina Nayak*

Our work is based on two main principles:

- *Queries retrieving the same data belong to the same subject.*
- *Queries belonging to a subject help in answering similar future queries.*

This work elaborates a technique for grouping similar queries from user logs based on a common subject, identifying frequently asked queries and an example that demonstrates the viability of this work. Our paper is organized as. Section 2 discusses related approach. Section 3 describes dataset used in the study and results. Section 4 discusses business implication of current work and conclusion and future work.

II. APPROACH

In our approach, we have devised an automatic method of selecting queries from the user log that are relevant for answering future queries. This selection is based on identifying common subjects amongst the previously posted queries using a similarity function and grouping the queries under the identified subjects. A frequent query selection technique is then applied to each of these subject area clusters to obtain the frequent query set. [25, 26, 29] This frequent set contains relevant data that has likelihood to answer future user queries of similar structure. These two techniques have been explained in detail below.

A. Subject Area Identification based on Nearest Neighbor

Our approach is based on the fact that most of the queries posted in the past are subject-specific. Only a minority of these queries belong to more than one subject or domain. As a result, it is suitable to group previously posed queries into subject domains and storing the groups obtained. We do this subject-specific grouping of past clusters by applying Nearest Neighbour Clustering Technique.[13] The similarity between previously posted queries is calculated using the DICE Coefficient.[7] According to DICE Coefficient,[7] the similarity between a pair of queries Qi and Qj i.e. Sim(Qi, Qj), based on DICE Coefficient measure, is given by

$$\mathrm{Sim}(Q_i, Q_j) = \frac{2|R(Q_i) \cap R(Q_j)|}{|R(Q_i)| + |R(Q_j)|}$$

WhereR(Qi) and R(Qj) are the relations accessed by queries Qi and Qj respectively.

Using the DICE coefficients obtained for all the past queries, a query similarity matrix is built.

The Nearest Neighbor clustering technique uses the query similarity matrix so obtained to group the past queries into subject-specific clusters. Each of these query cluster obtained signifies a subject area. The algorithm SubjectAreaIdentification, based on nearest neighbor clustering technique,[13] used to identify subject areas is given in Fig. 1 below. This algorithm takes 3 inputs: user query log, the query similarity matrix and a minimum query similarity threshold and produces the subject-specific query clusters as output.

The algorithm can be described as follows. Firstly the queries count QC and the cluster count CC is initialised to 1. Then, the first query QQC, from the previously posed queries Qp, is assigned to cluster CCC. The next query in Qp is then picked and its nearest neighbor, i.e. in terms of having maximum similarity, is identified from the queries that are already assigned to clusters. If this similarity is greater than or equal to the minimum similarity threshold ε, then the query is assigned to the corresponding cluster. Otherwise, a new cluster is created and the query is assigned to it. This continues till all queries have been considered. The identified clusters specify the various subject areas.

ALGORITHM SubjectAreaIdentification

Inputs: Q_P : Previously posed Queries queries,
□ : Minimum query similarity threshold,
SimMat : Similarity Matrix showing similarity between queries

Output: Cluster of Queries C_Q

Method:

STEP 1 Set query count QC = 1 and cluster count CC = 1.
STEP 2 Assign query Q_{QC} in Q_P to cluster C_{CC}
STEP 3 Increment QC by 1.
STEP 4 Find nearest neighbour of Q_{QC} among the queries in Q_P already assigned to clusters.
STEP 5 Using the SimMat, let MaxSim denote the similarity between $Q_{QC}$and it's nearest
Neighbor query in the existing clusters. Suppose the nearest is in cluster K
STEP 6 If MaxSim is greater than or equal to □, then assign Q_{QC} to C_K otherwise increment CC by one and assign Q_{QC} to C_{CC}
STEP 7 If every query has been considered then STOP else go to STEP 3.

Fig. 1 Algorithm Subject Area Identification based on Nearest Neighbor

Each of the clusters might contain a large number of queries. Some of these queries might contain information likely to be accessed more than the others. So identification of such queries is necessary in order to efficiently answer similar queries in the future. A technique for this selection of frequent queries is discussed next.

B. Frequent Query Selection

As mentioned above, identifying frequent queries from past queries reduces the response time for answering a user query in the future. So it's important to identify such queries accurately so that they contain only relevant information capable of answering future queries and not any random information. This technique identifies such relevant and required information by selecting queries that access frequently accessed information. These queries, referred to as frequent queries, provide information that have high a high likelihood of answering future queries and therefore can appropriately be used for consolidating relevant information for the corresponding subject area. The frequent queries selection algorithm, based on Apriori Algorithm given in Fig.2. This algorithm uses prior knowledge of frequent query properties.

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore(k+1)-itemsets. First, the setof frequent 1-itemsets is foundby scanning the database to accumulate the count for each item, and collecting those itemsthat satisfy minimum support. The resulting set is denoted *L*1.Next, *L*1 is used to find *L*2, the set of frequent 2-itemsets, which is used to find *L*3, and so on, until no more frequent*k*-itemsets can be found. The finding of each *Lk*requires one full scan of the database.

• Join Step: Ck is generated by joining Lk-1with itself
•Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
• Pseudo-code:
Ck: Candidate itemset of size k
Lk: frequent itemset of size k
L1= {frequent items};
for(k= 1; Lk!=∅; k++) do begin
Ck+1= candidates generated from Lk; for each transaction t in database do
increment the count of all candidates in Ck+1 that are contained in t
Lk+1= candidates in Ck+1 with min_support
end
return∪k Lk;

Fig. 2Algorithm FrequentQuerySelection based on Apriori

III.EXAMPLE

Below is a table which presents the previously posed queries on our Real Estate database:

Queries	Tables
Q1	State, City, Address
Q2	Current Project, Availability Status, City
Q3	Current project, Availablity Status, Address
Q4	Current Project,AvailablityStatus,Porperty type
Q5	State, Developer, Address
Q6	CurrentProject, Project, Developer
Q7	Current Project, Availablity Status, Budget
Q8	City, Developer, Address
Q9	State, Developer, City
Q10	Current Project, Availablity Status, Address
Q11	State, City, Address
Q12	City, Address, Developer
Q13	Address, Property type, Current Project
Q14	State, Current Project,Budget
Q15	Budget, Property type, Facilities
Q16	Budget, Property type, Organisation
Q17	Property Type, Facilities,Organisation
Q18	Current Project,Availablity Status, Property type
Q19	State, Devloper, City
Q20	Budget, Facilities, Organisation

Table 1: Previous Posted queries Relation Q1………..Q20

Below is a table which describes the table structure of our database:

S.NO.	Table Name	Columns
1	State	State ID, State Name
2	City	City ID, City Name, State ID
3	Address	Address ID, Address1, Address2, Landmark, City ID
4	Developer	Developer ID, Developer Name, Is Active, City ID
5	Property Type	Property Type ID, Property Type Name, Is Active, City ID, State ID
6	Budget	Budget ID, Budget Amount, State ID
7	Current Projects	Project ID, Project Name, Developer ID, Address ID, Property Type ID, Budget ID, Is Active
8	Availability Status	Project ID, Project Name, Developer ID, Address ID, Property Type ID, Budget ID, Status
9	Facilities	Facility ID, Budget ID, Property Type ID, Organization Name, Availability facilities,
10	Organization	Organization ID, Organization Name, Budget ID, Facility ID, Property Type Name

Table 2 Relations accessed by the Queries Q1...Q20

The similarity between the queries in Table 1 is computed using the DICE Coefficient. These similarities are then used to construct a similarity matrix, given below(Table 3):

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
Q1	1	O.333	0.333	0	0.666	0	0	0.666	0.666	0.333	1	0.666	0.333	0.333	0	0	0	0	0.666	0
Q2	0.333	1	0.666	0.666	0	0.333	0.666	0.333	0.333	0.666	0.333	0.333	0.333	0.333	0	0	0	0.666	0.333	0
Q3	0.333	0.666	1	0.666	0.333	0.333	0.666	0.333	0	1	0.333	0.333	0.666	0.333	0	0	0	0.666	0	0
Q4	0	0.666	0.666	1	0	0.666	0.666	0	0	0.666	0	0	0.666	0.333	0.333	0.333	0.333	1	0	0
Q5	0.666	0	0.333	0	1	0.333	0	0.666	0.666	0.333	0.666	0.666	0.333	0.333	0	0	0	0	0.666	0
Q6	0	0.333	0.333	0.667	0.333	1	0.333	0.333	0.333	0.333	0	0.333	0.666	0.333	0.333	0.333	0.333	0.666	0.333	0
Q7	0	0.666	0.666	0.666	0	0.333	1	0	0	0.666	0	0	0.333	0.666	0.333	0.333	0	0.666	0	0.333
Q8	0.666	0.333	0.333	0	0.666	0.333	0	1	0.666	0.333	0.666	1	0.333	0	0	0	0	0	0.666	0
Q9	0.666	0.333	0	0	0.666	0.333	0	0.666	1	0	0.666	0.666	0	0.333	0	0	0	0	1	0
Q10	0.333	0.666	1	0.666	0.333	0.333	0.666	0.333	0	1	0.333	0.333	0.666	0.333	0	0	0	0.666	0	0
Q11	1	0.333	0.333	0	0.666	0	0	0.666	0.666	0.333	1	0.666	0.333	0.333	0	0	0	0	0.666	0
Q12	0.666	0.333	0.333	0	0.666	0.333	0	1	0.666	0.333	0.666	1	0.333	0	0	0	0	0	0.666	0
Q13	0.333	0.333	0.666	0.666	0.333	0.666	0.333	0.333	0	0.666	0.333	0.333	1	O.333	0.333	0.333	0.333	0.666	0	0
Q14	0.333	0.333	0.333	0.333	0.333	0.333	0.666	0	0.333	0.333	0.333	0	0.333	1	0.333	0.333	0	0.333	0.333	0.333
Q15	0	0	0	0.333	0	0.333	0.333	0	0	0	0	0	0.333	0.333	1	0.666	0.666	0.333	0	0.666
Q16	0	0	0	0.333	0	0.333	0.333	0	0	0	0	0	0.333	0.333	0.666	1	0.666	0.333	0	0.666
Q17	0	0	0	0.333	0	0.333	0	0	0	0	0	0	0.333	0	0.666	0.667	1	0.333	0	0.666
Q18	0	0.666	0.666	1	0	0.666	0.666	0	0	0.666	0	0	0.666	0.333	0.333	0.333	0.333	1	0	0
Q19	0.666	0.333	0	0	0.666	0.333	0	0.666	1	0	0.666	0.666	0	0.333	0	0	0	0	1	0
Q20	0	0	0	0	0	0	0.333	0	0	0	0	0	0	0.333	0.666	0.666	0.666	0	0	1

Table 3 Query Similarity Matrix representing DICE coefficients for Q1….Q20

Using the similarity matrix in Table 3, the previously posed queries in Table 1, minimum query similarity threshold ε=0.5, the subject areas are identified as given below:

Initialize QC=1 and CC=1

Assign QQC i.e. Q1 to cluster CCC i.e. C1

Now C1= {Q1}

Set QC=QC+1 i.e. QC=2

MaxSim of QQC i.e. Q2 is 0 with nearest neighbor query Q1

Since MaxSim<□, set CC=CC+1 i.e. CC=2 and CCC i.e. C2={Q2}

Set QC=QC+1 i.e QC=3

MaxSim of QQC i.e Q3 is 0.666 with its nearest neighbor query Q2 in C2

Since MaxSim>□, Assign Q3 to C2

C2={Q2, Q3}

Set QC=QC+1 i.e. QC=4

MaxSim of QQC i.e Q4 is 0.666 with its nearest neighbor query Q2 and Q3

Since MaxSim>□, Assign Q4 to C2

C2={Q2, Q3, Q4}
Set QC=QC+1 i.e. QC=5
MaxSim of QQC i.e. Q5 is 0.666 with its nearest neighbor Q1
Since MaxSim>□, Assign Q5 to C1
C1={Q1, Q5}
Set QC=QC+1 i.e. QC=6
MaxSim of QQC i.e. Q6 is 0.666 with its nearest neighbor Q2, Q3 and Q4
Since MaxSim>□, Assign Q6 to C2
C2={Q2, Q3, Q4, Q6}

The above steps are carried out in the similar manner to identify cluster of queries. The cluster of queries C1, C2, C3, C4 and C5 identified represent the five subject areas S1, S2, S3, S4 and S5 respectively as given below:

S1={Q1,Q5, Q8, Q9, Q11, Q12, Q19}
S2= {Q2, Q3, Q4, Q6, Q7, Q10, Q13, Q14, Q18}
S3= {Q15, Q16, Q17, Q20}

Next, the frequent queries are selected in eachsubject area using the FrequentQuerySelectionusing Apriori algorithmgiven in Fig. 2. Considersubject area S2. The queriesalongwith the relation accessed by them are given in Table 2.

Input :
Minimum Support Count =5
Transactions = Q2,Q3,Q4,Q6,Q7,Q10,Q13,Q14,Q18 (Subject Area S2)

Output :
Frequent itemset

Instructions:
Step 1 : Consider cluster S2 (subject area – buying and selling).[Table 4]
Step 2 : Assign a symbol to each of the item sets(tables) involved.
So we get the following table corresponding to the cluster S2.
Step 3 : A Candidate Key Ck is obtained by counting all the occurrences of each table(denoted by symbols I1...I8) in the transactions(queries) and summarising it as follows :
Step 4 : The table obtained in Step 3 is then pruned i.e. less used tables are removed on the basis of Minimum Support Count and following result is obtained, known as Lk :
The above table gives us the most frequent itemsets from previously posed queries for our example i.e. I1(CurrentProjects) and I2(AvailabilityStatus).

Q2	Current Project, Availability Status, City
Q3	Current project, Availability Status, Address
Q4	Current Project, Availability Status, Property type
Q6	Current Project, Property Type, Developer
Q7	Current Project, Availability Status, Budget
Q10	Current Project, Availability Status, Address
Q13	Address, Property type, Current Project
Q14	State, Current Project, Budget
Q18	Current Project, Availability Status, Property type

Table 4 Itemsets in Subject Area S2–‘Buying and Selling’

Items	View
Current Project	I1
Availability Status,	I2
City	I3
Address	I4
Property Type	I5
Developer	I6
Budget	I7
State	I8

Table 4.1Symbols corresponding to each itemset

TID	ITEMSET
Q2	I1, I2,I3
Q3	I1,I2,I4
Q4	I1,I2,I5
Q6	I1,I5,I6
Q7	I1,I2,I7
Q10	I1,I2,I4
Q13	I4,I5,I1
Q14	I8,I1,I7
Q18	I1,I2,I5

Table 4.2 Queries in terms of the symbols assigned in the previous figure.

Item	Count
I1	9
I2	6
I3	1
I4	2
I5	4
I6	1
I7	2
I8	1

item	Count
I1	9
I2	6

Table 4.3Candidate Key ‘Ck’ Table 4.4Lk

Hence, we have analysed from the user query logs that the Real Estate users are more interested in current projects and their status like its availability for buying or selling.

IV. CONCLUSION AND FUTURE WORK

The amount of information handled today is enormous and would keep increasing with each passing day. So we need to keep inventing and improvising on the techniques used to search through this enormous amount of information. Through this paper we present a methodology which gives the search engines or any question answering system a future vision so that they can identify search patterns from user's past queries and exploit that information to answer users future queries. We are doing so by grouping the users past queries subject-wise in the first step. Each of the groups or clusters so obtained describes a subject or domain area pertaining to the type of database being used. In the next step, we identify frequent queries from each of the subject cluster and name them as FAQs(Frequently Asked Queries). These FAQs contain relevant information that might help in answering future queries which are similar in nature. This leads to a substantial reduction in the response times of the question answering systems since they already have answers prepared, if any user query matches an FAQ stored. Hence what we obtain are question answering systems that are better and efficient in responding to user queries which is a peak demand today.

Meanwhile, we are also working on another query clustering technique in which 'hot topics' will be predicted through thorough analysis of actual data of real estate. It will be very dynamic in nature and will brilliantly cater to the demand of response time reduction of question answering systems. Also, we are working towards implementing our clustering techniques onto mobile industries.

REFERENCES

[1]. Agrawal, S., Chaudhari, S. and Narasayya, V. ‘Automated Selection of Materialized Views and Indexes in SQL databases’, In 26th International Conference on Very Large Data Bases (VLDB 2000), Cairo, Egypt, pp. 495-505, 2000.

[2]. Aouiche, K. and Darmont, J. ‘Data mining-based materialized view and index selection in data warehouse’, In Journal of Intelligent Information Systems, Pages 65 – 93, 2009

[3]. Baralis, E., Paraboschi, S. and Teniente, E. ‘Materialized View Selection in a Multidimansional Database’, In 23rd International Conference on Very Large Data Bases (VLDB 1997), Athens, Greece, pp. 156-165, 1997

[4]. Brin, S., Motwani, R., Ullman, J.D., Tsur, S. "Dynamic Itemset Counting and Implication Rules for Market Basket Data", SIGMOD Record, Volume 6, Number 2: New York, June 1997, pp. 255 - 264.

[5]. Chaudhuri, S. and Shim, K. ‘Including Groupby in Query Optimization’, In proceedings of the International Conference on Very Large Database Systems, 1994

[6]. Chirkova R., Halevy A. Y., and Suciu D. ‘A Formal Perspective on the View Selection Problem’, In Proceedings of VLDB, pp 59-68, 2001

[7]. Frakes, W. B. and Baeza-Yates, R. Information Retrieval, Data Structure and Algorithms. Prentice Hall, 1992

[8]. Gupta H. and MumickI. S. ‘Selection of Views to Materialize in a Data warehouse’, IEEE Transactions on Knowledge & Data Engineering, 17(1), pp. 24-43, 2005

[9]. Gupta, A., Harinarayan, V. and Quass, D. ‘Generalized Projections: A Powerful Approach to Aggregation’, In proceedings of the International Conference of Very Large Database Systems, 1995

[10]. Harinarayan V., Rajaraman A. and Ullman J. D. ‘Implementing Data Cubes Efficiently’, ACM SIGMOD, Montreal, Canada, pp.205-216, 1996

[11]. Horng J. T., Chang Y. J., Liu B. j., Kao C. Y. ‘Materialized View Selection Using Genetic Algorithms in a Data warehouse System’, In Proceedings of the 1999 congress on Evolutionary Computation, Washington D. C., USA, Vol. 3, 1999

[12]. Inmon W. H. ‘Building the Data Warehouse’, 3rd Edition, Wiley Dreamtech India Pvt. Ltd, 2003

[13]. Jain, A.K. and Dubes, R.C. “Algorithms for Clustering Data”. Englewood Cliffs NJ: Prentice Hall, 1988

[14]. Lawrence, M. ‘Multiobjective Genetic Algorithms for Materialized View Selection in OLAP Data Warehouses’, GECCO’06, July 8-12, SeattleWashington, USA, 2006

[15]. Lehner, W., Ruf, T. and Teschke, M. ‘Improving Query Response Time in Scientific Databases Using Data Aggregation’, In proceedings of 7th International Conference and Workshop on Database and Expert Systems Applications, DEXA 96, Zurich, 1996

[16]. Mohania M., Samtani S., Roddick J. and Kambayashi Y. ‘Advances and Research Directions in Data Warehousing Technology’, Australian Journal of Information Systems, 1998

[17]. O’Neil, P. and Graefe, G. ‘Multi-Table joins through Bitmapped Join Indices’, SIGMOD Record, Vol. 24, No. 3, pp. 8-11, 1995

[18]. Shah, B., Ramachandran, K. and Raghavan, V. ‘A Hybrid Approach for Data Warehouse View Selection’, International Journal of Data Warehousing and Mining, Vol. 2, Issue 2, pp. 1 – 37, 2006

[19]. Teschke, M. and Ulbrich, A. ‘Using Materialized Views to Speed Up Data Warehousing’, Technical Report, IMMD 6, Universität Erlangen-Nümberg, 1997

[20]. Theodoratos, D. and Sellis, T. ‘Data Warehouse Configuration’. In proceeding of VLDB pp. 126-135, Athens, Greece, 1997

[21]. Theodoratos, D. and Xu, W. ‘Constructing Search Spaces for Materialized View Selection’, In 7th ACM Internatioanl Workshop on Data Warehousing and OLAP (DOLAP 2004), Washington, USA, 2004

[22]. Vijay Kumar, T.V., Ghoshal, A.: A Reduced Lattice Greedy Algorithm for Selecting Materialized Views, Communications in Computer and Information Science (CCIS), Volume 31, Springer Verlag, pp. 6-18, 2009

[23]. Vijay Kumar, T.V., Haider, M., Kumar, S.: Proposing Candidate Views for Materialization, Communications in Computer and Information Science (CCIS), Volume 54, Springer Verlag, pp. 89-98, 2010

[24]. Vijay Kumar, T.V., Haider, M.: A Query Answering Greedy Algorithm for Selecting Materialized Views, Lecture Notes in Artificial Intelligence (LNAI), Volume 6422, Springer Verlag, pp. 153-162, 2010

[25]. Vijay Kumar, T.V. and Jain, N.: Selection of Frequent Queries for Constructing Materialized Views in Data Warehouse, The IUP Journal of Systems Management, Vol. 8, No. 2, pp. 46-64, May 2010

[26]. Vijay Kumar, T.V., Goel, A. and Jain, N.: Mining Information for Constructing Materialised Views, International Journal of Information and Communication Technology, Inderscience Publishers, Volume 2, Number 4, pp. 386-405, 2010

[27]. Vijay Kumar, T.V., Haider, M.: Greedy Views Selection using Size and Query Frequency, Communications in Computer and Information Science (CCIS), Volume 125, Springer Verlag, pp. 11-17, 2011

[28]. Vijay Kumar, T.V., Haider, M., Kumar, S.: A View Recommendation Greedy Algorithm for Materialized Views Selection, Communications in Computer and Information Science (CCIS), Volume 141, Springer Verlag, pp. 61-70 , 2011

[29]. Vijay Kumar, T.V. and Devi, K. ‘Frequent Queries Identification for Constructing Materialized Views’, In the proceedings of the International Conference on Electronics Computer Technology(ICECT-2011), April 8-10, 2011, Kanyakumari, Tamil Nadu, Published by IEEE, Volume 6, pp. 177-181, 2011

[30]. Vijay Kumar, T.V., Haider, M.: Selection of Views for Materialization using Size and Query Frequency, Communications in Computer and Information Science (CCIS), Volume 147, Springer Verlag, pp. 150-155, 2011

[31]. Widom, J. 'Research Problems in Data Warehousing', 4th International Conference on Information and Knowledge Management, Baltimore, Maryland, pp. 25-30, 1995

[32]. Yang, J., Karlapalem, K. and Li, Q. 'Algorithms for Materialized View Design in Data Warehousing Environment', The Very Large databases (VLDB) Journal, pp. 136-145, 1997

AUTHORS PROFILE

Gaurav Dubeyis working as an assistant professor at Amity University, Noida, India. He has completed his B.Tech & M.Tech. in computer science & is currently pursuing Ph.D. His areas of research include data mining and DBMS.

Romina Nayak is working as a Software Engineer at Kellton Tech Solutions Ltd., Gurgaon, India. She has completed Bachelor of Technology in June 2012 from B.S. Anangpuria Institute of Technology and Management, Faridabad, India.

Neha Wadhwa is a M.Tech. scholar in computer science and engineering at Amity University, Noida, India. She is also working as a Software Engineer at Kellton Tech Solutions Ltd., Gurgaon, India.

Dr. Ajay Rana is working as Director at Amity University, Noida, India. He has completed his B.Tech., M.Tech. & Ph.D. in computer science. (ajay_rana@amity.edu)