

Dynamic Query Form for Database Queries

Subhash V. Pingale* Anandrao Deshmukh**

Abstract : With highest growth, all databases like scientific, internet information databases and web databases maintain enormous amount of data. Some traditional query forms are not able to satisfy user's ad-hoc queries. In This paper we developed a DQF, Dynamic Query Form, which generates query forms dynamically. Main purpose to use DQF is for reaching user's satisfaction and guiding him for making decisions about the query form through ranking. Ranking lists of form components helps user for selecting desired form component and get query results. The system used the expected F-measure for measuring the virtue of query and evaluating the ranking list. Ranking list make query easy to use, therefore efficiency increases.

Keywords : Dynamic Query Form, F-measure, query forms,rank, fscore.

1. INTRODUCTION

A database is only a collection of large amount of data. We can persist, update, and delete the data to or from the database. To perform such a manipulation, we require a query interface. A user can communicate with the database. Problem here is traditional databases can't give output as user's satisfaction. *i.e.* only professional developers, programmers can understand the database query languages. Some normal users can't understand such programming languages. Even SQL, Oracle, etc databases are well-structured query languages, it can be challenging due to a number of reasons, including the user's lack of familiarity with the query language and the user's ignorance of the underlying schema. A form based query interface, which only requires filling blanks to for query details, is easy to understand for users even if they have no knowledge of official query languages or the database schema. In practice, form-based interfaces are used frequently, but usually each form is designed in necessary way and its use is limited to a small set of fixed queries.

Query form is a user interfaces for querying databases. Traditional query forms are designed by developers or DBA in all database management systems. As growth of internet andscience goes on increasing, requirement of modern databases is increased and it becomes more complex. To recover such a problem, we used Dynamic question type system: DQF, a question interface that is capable of dynamically generating question forms for users, different from document retrieval system. In information retrieval area, the users always perform several rounds of actions (refinement of question conditions) until they will get satisfaction.

The main purpose of DQF is to understand the user interests about query form and to adapt the question type repeatedly. The question type iteration consists of two sorts of user interactions: it contains attributes of the information. The interaction continues between the user and the system till the user satisfies with query results. Goal of this Project is to show that the advantages of using dynamic query forms for database over the existing static forms for database. We reach to user satisfaction using ranking attribute.

Ranking attribute stores users query results. Whenever user revisits the site it will display all related query results in ranking list. If user is satisfied with these results then he will use the results otherwise he will go with new form.

* Department of Computer Science Engineering SKN Singhgad College of Engineering Korti ,Pandharpur, Solapur, India,

** Department of Computer Science Engineering SKN Singhgad College of Engineering Korti ,Pandharpur , Solapur, India anandgdeshmukh@gmail.com

2. PROPOSED ALGORITHM

Methodology

DQF will actually work in two phases one is Query Execution and other is Query Enhancement.

A. Query Execution:

In Query execution user will fill the query form with desired form components, query form will be executed by the system and result will be displayed. If user gets desired output of that query form then user will rank the query form for future use. Depending on the ranking, query form will be restored for further use. And user can exit from the systems else user can get another query form by selecting other form components of his desire to get expected query result. After that user will execute the query form and the process will repeat until user will get desired output.

B. Query Enhancement

As said in above query execution method, if user is not satisfied with query results then again he will select the query component, fill the form and he will get the results. If user is satisfied then use the query from ranking list, otherwise he will select another form component. By this query component enhanced based on user choice and query form improvement and efficiency is increased.

C. Ranking Attribute

Ranking attribute stores query component of user and restore it whenever he revisits the system. If his query is related to ranking list then he will select and use the rank or he go with new query form.

D. Ranking Matrices

There are two measures for evaluating the quality of the query results: precision and recall. *Expected precision* and *expected recall* to evaluate the expected performance of the query form.

Expected precision is the expected proportion of the query results which are interested by the current user.

$$\text{Precision } E(F) = \frac{\sum_{d \in XAf} P_C(dAf) P(dAf) P(\alpha f | d)N}{\sum_{d \in XAf} P(dAf) P(\alpha f | d)N} \tag{1}$$

Expected recall is the expected proportion of user interested data instances which are returned by the current query form.

$$\text{Recall } E(F) = \frac{\sum_{d \in XAF} P_C(dAf) P(dAf) P(\alpha f | d)N}{\alpha N} \tag{2}$$

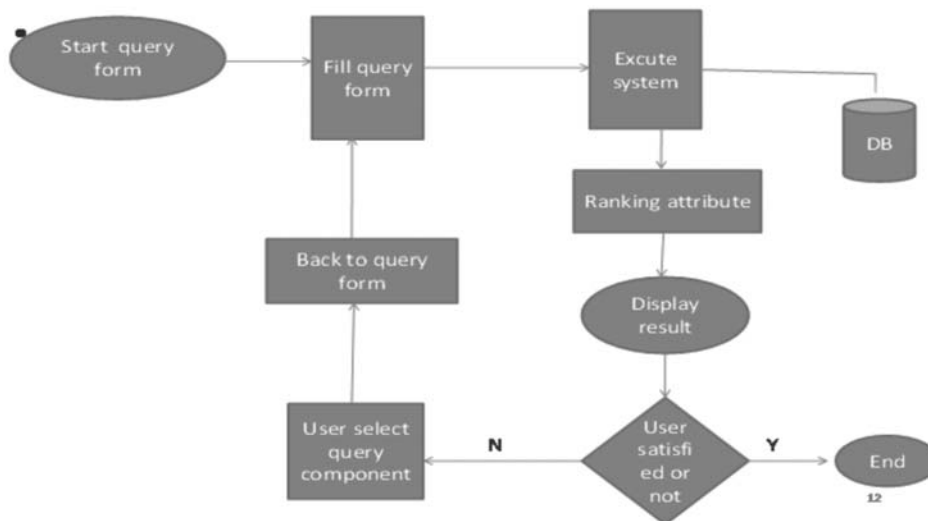


Fig. 1. System architecture

The calculation of ranking score is an entity is by the averaged F-Score. Rank score is used to calculate the accuracy. Initially the user select the components from more no of components and the ranking are calculated for selected components has the highest rank.

3. EXPERIMENT AND RESULT

Step 1: Generate query

The system provides a form window. User needs to select the generate query tab and he need to select the tables, attributes of the table and operation to be performed (*i.e.* Insert, delete and update method).

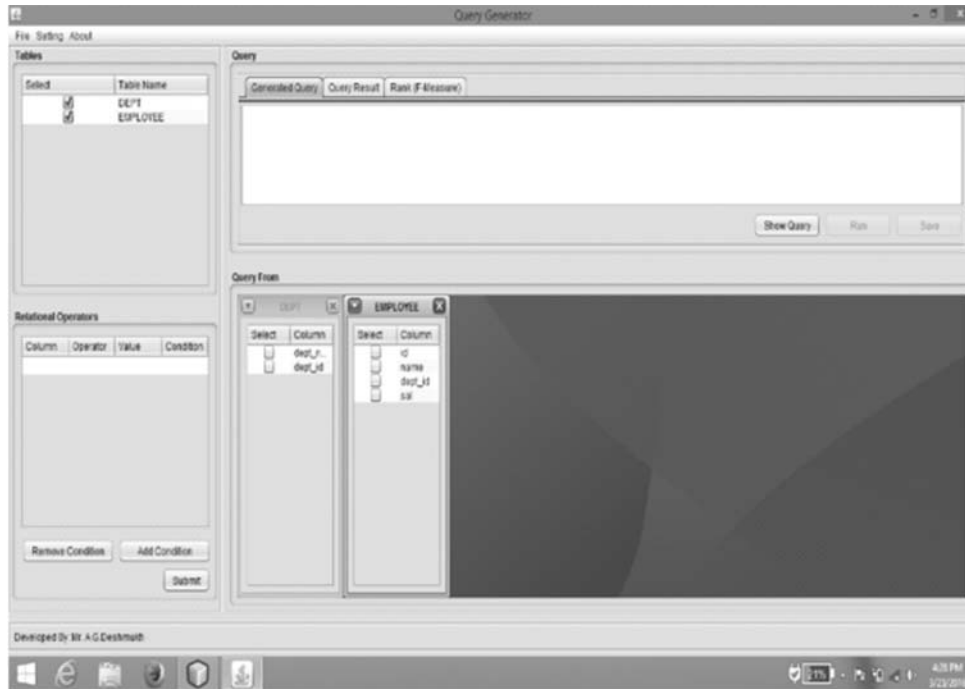


Fig. 2. Generate query

Step 2: Query results

Once user selects all the details in generate query tab, he will click on show query. Based on the given values the system will display query results.

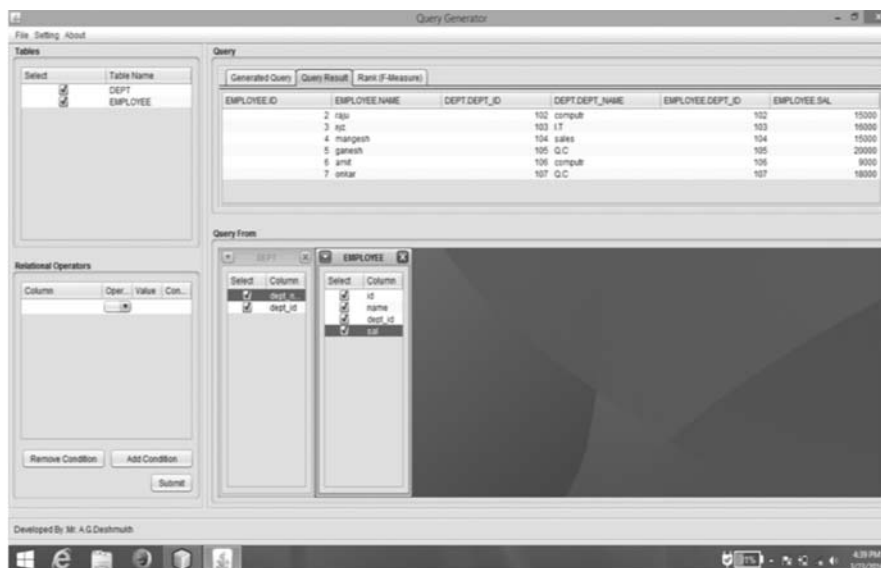


Fig. 3. QueryResults

Step 3: Rank Attribute

Rank stores the query results for future use and restores when user executes another query in future. Using rank it is possible to know the user interest and we make query enhancement easily.

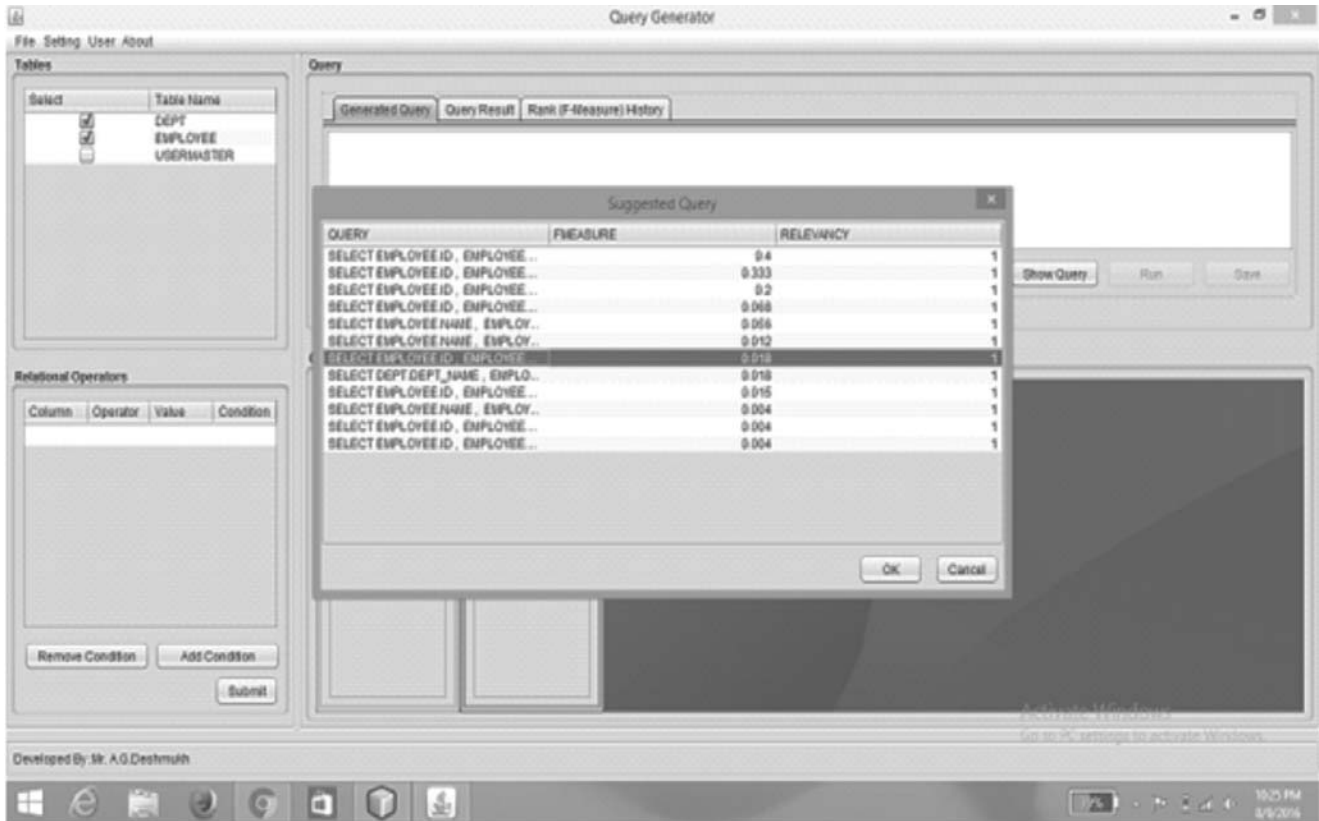


Fig. 4. Rank Attribute

V(A) RESULT

2. RESULTS

The Concept of this paper has implemented and results are shows as follows..

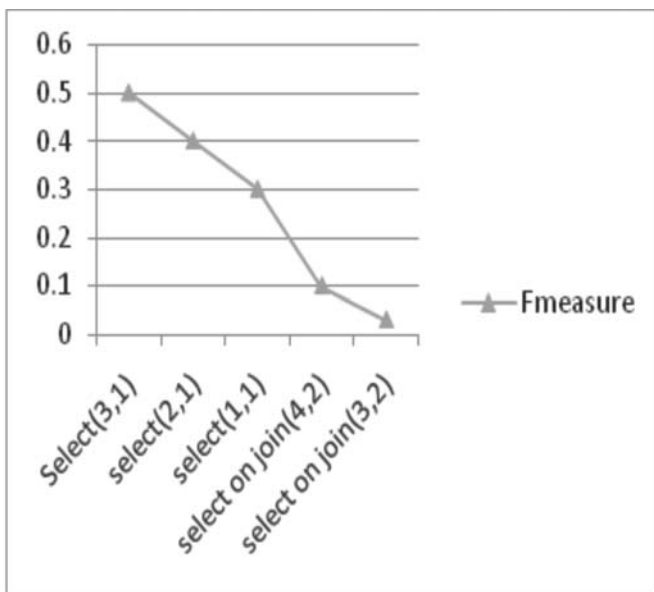


Fig. 4. Fmeasure

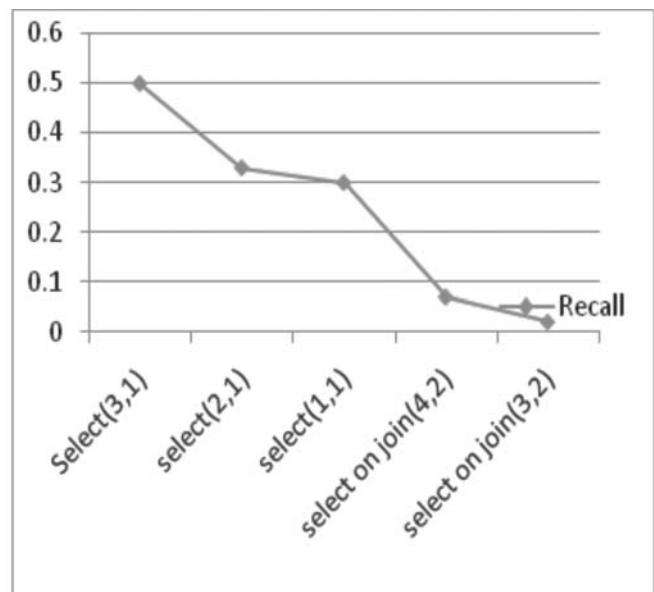


Fig. 6. Recall

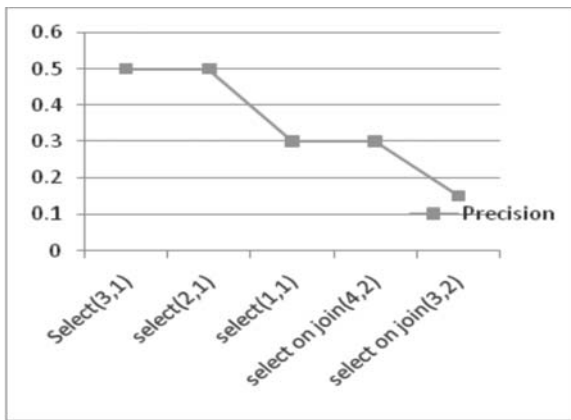


Fig. 6. Precision

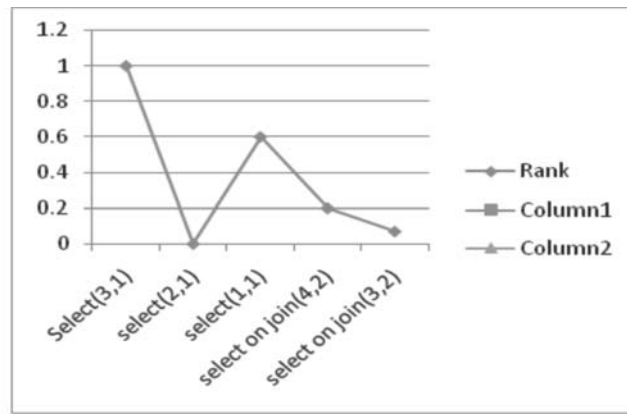


Fig. 7. Rank

3. CONCLUSION

Using traditional query form system, users didn't get results as expected. Therefore we moved to dynamic query form which dynamically generates the query forms for evaluating the database queries. We have implemented Dynamic Query Form. In this system, user just needs to fill the query form and submit it to get results. We added one feature i.e. ranking system. In ranking system, the system rank the query form components based on user interest and restore the historical query results whenever necessary. It is possible to capture the user interests based on historical queries. The experimental results show that dynamic query form leads higher success than static query forms.

4. FUTURE WORK

We extend our method to non relational data. And try to add keyword matching concept, so that user can just search any previous query results in search window. And the system matches the keywords with existing data and displays the results.

5. REFERENCES

1. Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen "Dynamic Query Forms for Database Queries", IEEE Trans. On Knowledge and Data Engg. Vol:PPNo:99 Year 2014
2. H.V. Jagadish "Automating the design and construction of Query form", IEEE trans. Knowledge data eng, Vol. 21 no.10 oct.2009 pp 1389-1402.
3. M. Jayapandian proposed system "Automating the design and construction of Query form", IEEE trans. Knowledge data eng, Vol. 21 no.10 oct.2009 pp 1389-1402.
4. S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. "Probabilistic information retrieval approach for ranking of database query results". ACM Trans. Database Syst. (TODS), 31(3):1134-1168, 2006.
5. Vinayak Jadhav, Amrit Priyadarshi "A Survey on Database Queries by using Dynamic Query Forms" 1, 2 Department of Information Technology, Dattakala Group of Institute of Technology, University of Pune, Pune, Maharashtra, India, International Journal of Science and Research (IJSR) 2012.
6. Meenu Joy Bhuruguram T M Adi Shankara "Dynamic Query Form with query Refinement and Database encryption" Institute of Engineering and Technology, Kalady, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. IV (Sep - Oct. 2014), PP 154-159 www.iosrjournals.org.
7. S. Naresh & B. Gurunadha Rao "Ranking based Dynamic Query Forms for Database Queries". Sree Vidyanikethan Engineering College India. Global Journal of Computer Science and Technology: C Software & Data Engineering Volume 15 Issue 4 Version 1.0 Year 2015 Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc. (USA) Online ISSN: 0975-4172 & Print ISSN: 0975-4350.
8. Priyanka P. Nikam, "A Review on Dynamic Query Forms for Database Queries" Student, Computer Engg. Department, MCOERC Nashik Pune University, India.
9. C. C. Agawam, J. Han, J. Wang, and P. S. Yu. "A framework for clustering evolving data streams". In Proceedings of VLDB, pages 81-92, Berlin, Germany, September 2003. S. Agrawal, S. Chaudhuri, G. Das, and A. "Gionis. Au-tomated ranking database query results." In CIDR 2003.