

## Smart Forms for Dynamic Queries over Large Databases

Vishal kumar Patel

PG Student:Computer Dept.  
Shree L.R. Tiwari College Of Eng.  
Thane, India  
*vishalpatel236@gmail.com*

Sharmila Gaikwad

Dr& Professor: Computer Dept.  
Rajiv Gandhi Institute Of  
Technology  
Mumbai, India  
*sharmila\_gaikwad@yahoo.com*

Vinayak shinde

HOD & Professor: Computer Dept.  
Shree L.R. Tiwari College Of Eng.  
Thane, India  
*vdshinde@gmail.com*

**Abstract:** Modern scientific databases and internet databases maintain massive and heterogeneous data. These real-world databases contain lots of or maybe thousands of relations and attributes. Traditional predefined queries forms don't seem to be able to satisfy numerous ad-hoc queries from users on those databases supported the analysis this project suppliers, a completely unique info query form interface, which is in a position to dynamically generate query forms. The essence of DQF (Dynamic Query Form) is to capture a user's preference and rank query form component, helping him/her in making choices. The generation of a query form is an iterative process and is guided by the user. At each and every iteration, the system mechanically generates ranking lists of form component and therefore the user then adds the required form parts into the query form. The ranking of form components is predicated on the captured user preference. A user may also fill the query form and submit queries to look at the query result at every iteration. During this approach, a query form might be dynamically refined till the user is happy with the query results. We have a tendency to utilize the expected F-measure for mensuration the goodness of a query form. A probabilistic model is developed for estimating the goodness of a query form in DQF. Our experimental analysis and user study demonstrate the effectiveness and potency of the system.

**Keywords:** -Query Form, Query Form Generation, User Interaction, Dynamic Query Form(DQF)

\*\*\*\*\*

### I. INTRODUCTION

Database is a group of related data which can access quickly and similarly it can hold the description of data. We can get the data from tables based on the needs of the user. To access the data from the database we can use the queries as basis query forms. Based on user needs the query forms are defined by the designer. Presently there are many static query forms and database management tools such as MS Access, SAP and Easy. By these static query forms and tools the user does not get the accurate results based on the user needs. Dynamic Query Forms is provides an interface to the users to perform the query form actions on the database. Generally dynamic query form has some attributes of the database. Each and every query-form generation is kind of process of selection attributes from the database. In dynamic query form user queries can be refined each and every iteration. It will also generate the rank list for these dynamic query forms. The dynamic query should acts as a interaction between the user and system, till the user get intended results. The main role here is to provide the query forms as per the user needs and quality of the query form.

### II. Literature Survey

#### Automated Creation of Forms:

M. Jayapandian presented a data-driven method. It first finds a set of data attributes, which are most likely queried

based on the database schema and data instances. Then, the query forms are generated based on the selected attributes.

#### Automating the design and construction of query forms:

H.V. Jagadish presented a workload-driven method. It applies clustering algorithm on historical queries to find the representative queries. The query forms are then generated based on those representative queries. One problem of the aforementioned approaches [3],[8] is that, if we generate lots of query forms in advance, there are still user queries that cannot be satisfied by any one of query forms. Another problem is that, when we generate a large number of query forms, how to let users find an appropriate query form would be challenging.

#### Modified Query Form:

Your The tools provided by the database clients make great efforts to help developers generate the query forms, such as Easy Query, Cold Fusion and so on. They provide visual interfaces for developers to create or customize query forms. The problem of those tools is that, they are provided for the professional developers. H.V. Jagadish proposed a system which allows end-users to customize the existing query form at run time. If the database schema is very large, it is difficult for end user to find appropriate database entities and attributes.

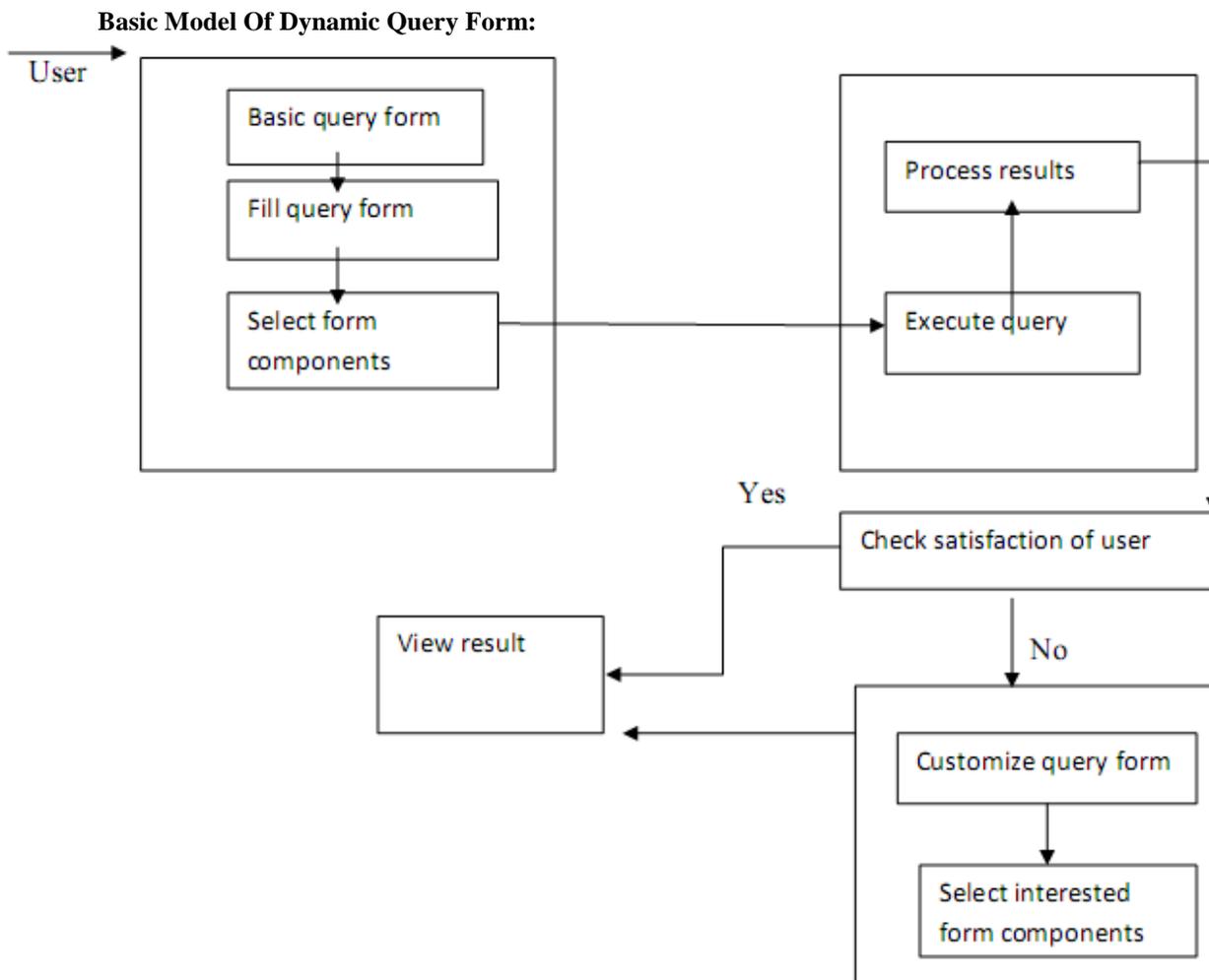


Fig 1. Architecture Of Dynamic Query Form.

### III. PROPOSED SYSTEM

The project proposes a Dynamic Query Form system: DQF, a query interface which is capable of dynamically generating query forms for users. Different from traditional document retrieval, users in database retrieval are often willing to perform many rounds of actions (i.e., refining query conditions) before identifying the final candidates. The essence of DQF is to capture user interests during user interactions and to adapt the query form iteratively. Each iteration consists of two types of user interactions: Query Form Enrichment and Query Execution. It starts with a basic query form which contains very few primary attributes of the database. The basic query form is then enriched iteratively via the interactions between the user and our system until the user is satisfied with the query results. In this paper, we mainly study the ranking of query form components and the dynamic generation of query forms.

#### Advantages of Proposed System:

A dynamic query form system which generates the query forms according to the user's desire at run time. The system provides a solution for the query interface in large and complex databases.

The goodness of a query form is determined by the query results generated from the query form. Based on this, we rank and recommend the potential query form components so that users can refine the query form easily.

### IV. CONCLUSION

The project gives dynamically query form generation for the database queries. Here the main idea is the rank the query form components based on user interest. To capture the user interests based on historical queries and click through feedback. The experimental results show that dynamic query form leads higher success than static query forms. The ranking is also makes easier for user selection in the query form attributes.

Experimental results have shown that the dynamic approach usually generates higher success rates and easier query forms compared with a static approach. The form component rankings make it easier to customize the query forms for users. As development work, the study can be undertaken to extend our approach to non-relational data.

We plan to develop number of methods to capture the user's preferences for the queries instead of click feedback. For example, a text box can be added to input some keyword queries and the query form for the user. At each step, this can be incorporated in the form component rankings.

### REFERENCES

- [1] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen "Dynamic Query Forms for Database Queries" - IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 9, SEPTEMBER 2014.
- [2] S. Cohen-Boulakia, O. Biton, S. Davidson, and C. Froidevaux, "Bioguidesrs: querying multiple sources with a user-centric perspective", Bioinformatics, 2013.
- [3] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, "Automated ranking of database query results", In CIDR, 2003.
- [4] G. Salton and M. McGill, "Introduction to Modern Information Retrieval", McGraw-Hill, 1984
- [5] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum, "Probabilistic information retrieval approach for ranking of database query results", ACM Trans. Database Syst. (TODS), 2006.
- [6] ColdFusion.  
<http://www.adobe.com/products/coldfusion>
- [7] EasyQuery. <http://devtools.korzh.com/eq/dotnet> M. Jayapandian and H. V. Jagadish. Expressive query specification through form customization. In Proceedings of International Conference on Extending Database Technology (EDBT), pages 416–427, Nantes, France, March 2008.