

## Dynamic Query form for Database

Prof. R. R. Yadav  
Assistant Professor, CSE Department  
DESCOET, Dhamnagaon Rly  
rajyadav.engg@gmail.com

Miss. Sarika S. Joshi  
Student, CSE Department,  
DESCOET, Dhamnagaon Rly  
sarikajoshi.engg@gmail.com

Miss. Kalyani L. Jaiswal  
Student, CSE Department  
DESCOET, Dhamnagaon Rly  
kalyani.jaiswal101@gmail.com

**Abstract:-** Database is collection of number of relations and attributes which makes it more complex. Such a large and heterogeneous data is difficult to be maintained. Developers are familiar with the queries but might face problem of writing queries again and again for different situations. And problem is also faced by those who are not familiar with databases but needs to access data from databases. To solve problem of both these parties we are proposing a model called Dynamic Query Form which will be able to dynamically generate query form. It will provide recommendations of components based on user preferences. It will not just generate dynamic query form but new data can be added and deleted by user accordingly. User adds the desired components, specifies the conditions and gets the result. If user is satisfied then task is accomplished otherwise user can add or remove components and gets result for it and this continues until user is satisfied. The ultimate goal of DQF is user's satisfaction.

**Keywords:-** Query form, user preference, query form generation, recommendations, enrichment, components, query execution.

\*\*\*\*\*

### I. Introduction

Query form is the most broadly utilized UI for questioning databases. Ordinary question structures are outlined and predefined by engineers or DBA yet it is not reasonable for end client's having no learning about databases. With the quick improvement, present day databases turn out to be vast and complex. Many web databases, for example, Freebase and DBpedia, commonly have a huge number of organized web substances [4] [2]. In this way, it is hard to plan static inquiry structures to fulfill different specially appointed database questions on those unpredictable databases. It is conceivable that many questions are not characterized in it but rather is required by client.

Many existing database administration and improvement apparatuses, for example, EasyQuery [3], Cold Fusion [1], SAP and Microsoft Access, give a few systems to give clients a chance to make modified questions on databases. Be that as it may, the making of tweaked questions absolutely relies on upon clients' manual altering [5]. It is valuable for engineers or for the individuals who know about databases. On the off chance that a client is not acquainted with the database blueprint ahead of time, those hundreds or a large number of information qualities would confound him/her without a doubt.

Here we propose a Dynamic Query Form framework: DQF, an inquiry interface which is able to do progressively producing question shapes for client's benefit..

Query Form Execution	<ol style="list-style-type: none"><li>1. User fills the query form using query form components.</li><li>2. User specifies the desired conditions.</li><li>3. DQF executes the query and displays the result.</li><li>4. User provides feedback about the result.</li></ol>
Query Form Enrichment	<ol style="list-style-type: none"><li>1. User is provided recommendation ranked list of query form components.</li><li>2. User adds or removes the components according to his desires.</li></ol>

Every emphasis comprises of two sorts of client associations: Query Form Execution and Query Enrichment (see Table 1). Figure 1 demonstrates the work-stream of DQF. It begins with an essential inquiry shape which contains not very many essential traits of the database. The essential question shape is then advanced iteratively by means of the cooperation's between the client and our

framework until the client is happy with the inquiry comes about.

Inside the proposed framework, first client will fill the question shape with required segments and conditions and this inquiry will be executed by the framework and the outcome is shown on the screen. Next, the client give criticism whether the client is happy with the question frame and in the event that he is not fulfilled then there is an alternative of inquiry improvement through which client can include or evacuate parts as he cravings and prescribed positioned rundown will be given to help client to taking choices. In the event that client is fulfilled, our objective is finished generally this procedure proceeds until client is completely fulfilled.

## II. Related Work

DQF is most widely used user interface for querying the database but before DQF there were lots of research performed for finding an interface that makes user capable of querying database. The main motive of DQF is to adapt itself according to user satisfaction. Following researches were carried out before the evolution of DQF:

### 1. Static Query Form

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. It is a workload-driven method. One problem of the aforementioned approaches is that, if the database schema is large and complex, user queries could be quite diverse. In that case, even 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 and desired query form would be challenging.

### 2. Customized Query Form:

Customized Query Form: Existing database clients provide tools which provide visual interfaces to design query forms like Easy Query, Cold Fusion, SAP and Microsoft Access. It helps users to create custom query forms. The draw-back of these aforementioned tools is that they are provided for the professional developers who are familiar with their database and not for the end user who don't have any knowledge of database.

## III. Work flow of System

For an implementation of proposed system is as following:

1. Insertion And Deletion of Components into System
  2. Adding Desired Components in Form
  3. Applying Desired Conditions
  4. Query execution
  5. Query Enrichment
1. *Insertion And Deletion of Components*  
Components can be added or deleted by user in this module. This module will ask for table name and its attributes but as user is not familiar with database system will ask like about what you want to save information and as user will type the name, the table of that name will be created then what kind of information you want to save the user will fill attributes and at last it will ask for values of those attributes.
  2. *Adding Desired Components in Form*  
User first enters name of database and gets a list of tables that are present in that particular database. Each table is consist of components present within it and user selects various components which user needs.
  3. *Applying Desired Conditions*  
After choosing desired components user needs to specify conditions. User selects the component on which the condition is to be applied. Next the user selects the condition to be applied and at last user puts the value of condition.
  4. *Query Execution*  
After specifying the condition the DQF generates query for given data and presents the result. The user will either be satisfied with the result and will end up with that result otherwise if not satisfied he will perform query enrichment.
  5. *Query Enrichment*  
If user is not satisfied with the result he can choose query enrichment through which the user can choose or remove components as per requirement and can alter the specified condition and will execute the query again. At each iteration the user will get a recommended list of components which will help user in making decisions.

Recommendations are presented on the base of user preferences. The components selected maximum by user will be ranked accordingly.

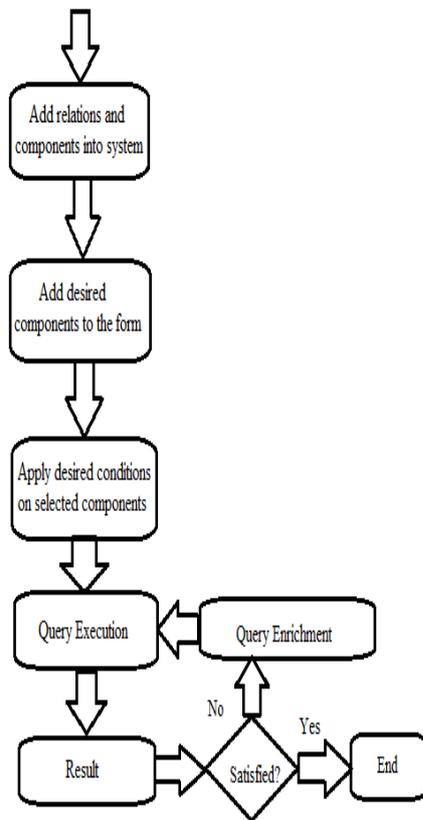


Fig.1. Workflow of System

#### IV. Advantages

- DQF provides user an interface for accessing data from database without having any knowledge about database querying languages.
- DQF saves the time of developers as he doesn't need to write the query again and again.
- The dynamic approach often leads to higher success rate and simpler query forms compared with a static approach.
- The ranking of form components also makes it easier for users to customize query forms.

#### Conclusion

Query interfaces play a most important role in determining the usefulness of a database. User-friendly querying method is provided by form-based interface that are widely used in large and complex database. In the proposed system user interact with the system and form the dynamic query as per requirement. The proposed system provides higher success rate and simple query for user. We capture user feedback at run-time by click through method. As future work, In future proposed system works with non-relational data. In the future we plan to develop multiple

methods to capture the users interested for the queries besides the click feedback. For instance, we can add a text-box for users to write some input keywords queries.

#### References

- [1]. C. C. Aggarwal, 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.
- [2]. R. Agrawal, S. Gollapudi, A. Halverson, and S. Jeong. Diversifying search results. In *Proceedings of WSDM*, pages 5–14, Barcelona, Spain, February 2009.
- [3]. S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In *CIDR*, 2003.
- [4]. S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of SIAM International Conference on Data Mining (SDM 2008)*, pages 243–254, Atlanta, Georgia, USA, April 2008.
- [5]. A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda. Usability measurement and metrics: A consolidated model. *Software Quality Journal*, 14(2):159–178, 2006.