

A Survey on Database Queries Using Dynamic Query Forms

Miss. Priyanka G. Varkad¹, Prof. P. N. Kalavadekar²

¹PG student, Department of Computer Engineering

² Asst Prof., Department of Computer Engineering

SRES's College of Engineering, Kopargaon
Maharashtra, India.

ABSTRACT

With the rapid enhancement of web services and their popularity, web users are increasing day by day. The modern databases are relational and include large number of relations and attributes. By this enhancement of web information and scientific databases it is not able to get user require results with the static query forms. The solution to this problem is dynamic queries. This paper provides a Dynamic Query Form (DQF), a curious database query form interface which is able to tackle the large and complex relational databases. A system captures the user's preference during the user communication and guides user to make decisions. Query form generation is a iterative process. The ranking of form components is based on the captured user preferences. A user can also fill up the query form and deliver queries to view the query output at each step. In this way, a query form could be dynamically refined till the user gets the query result.

Keyword: - Query Form, User Interaction, Query Form Generation, Query Form Enrichment, and Dynamic Query Forms (DQF).

1. Introduction

'Query Form' is one of the most extensively used user interface for querying databases to access information. The traditional query forms are configured and predefined by the Developers and Database Administrators in various information management systems. With the rapid development of web informatics and scientific databases, the modern databases have become very huge and complex. There are over hundreds of entities for biological and chemical data resources in the databases in natural sciences, such as diseases and genomics. The web databases, like DBPedia and Freebase, usually have over thousands of structured web entities. So, it is difficult to design a set of static query forms which better response the different ad-hoc database queries on those complex databases. Many modern database management and development tools, such as SAP and MS Access, allows user to develop customized queries on databases, by providing several mechanisms. The development of these customized queries totally based upon manual editing's of user. If user is not familiar with the database schema in advance, he/she will be confused by the hundreds and thousands of data attributes. Tackling with the relational database is a challenging task for non-technical user. Considering this view, in recent years many researchers are focusing on database interfaces so that user can query the relational databases with no SQL easily.

This paper proposes a Dynamic Query Form (DQF) system, an interface which is capable of generating query forms for user at runtime. Different from traditional document retrieval, prior to identify the final candidate, the users in database retrieval need to execute several rounds of action [6]. The important features of DQF includes: a) Capture the user interest during the user interaction and b) Iteratively adapt the query forms.

The rest of the paper is arranged as follows: Section 2 presents the literature survey over the related work. Section 3 gives in brief idea about the existing system and proposed system. Finally, the section 4 concludes the review paper.

2. Literature Survey

Researchers focus is on database interfaces which assist users to query the relational database with no SQL. There are two most widely used database querying interfaces: QBE (Query by Example) and Query Form. Current studies and works mainly focus on how to create the query forms.

1. M. Jayapandian and H. V. Jagadish, in their paper “Automated creation of a forms based database query interface,” stated that various existing database management and development tools, such as EasyQuery, Cold Fusion, SAP and Microsoft Access, provide several mechanisms to let users create customized queries on databases. However, the creation of customized queries totally depends on user’s manual editing. If a user is not familiar with the database schema in advance, those hundreds or thousands of data attributes would confuse him/her. It first finds a set of data attributes, which are most likely queried based on the database schema and data instances. After that, the query forms are generated based on the chosen attributes [1].

2. M. Jayapandian and H. V. Jagadish, “Automating the design and construction of query forms,” proposed a workload driven method. It aims to find the representative queries by applying the clustering algorithm. Based on those representative queries the query forms are generated. One problem of the aforementioned approaches is that, in case of lots of query forms generation in advance, there are still user queries which cannot be satisfactorily solved by any one of query forms. Another problem is that, when we create a huge number of query forms, it’s a difficult task to let users find an appropriate query form would be difficult. The Query interfaces play a vital role in determining the usefulness of a database. A form-based interface is widely regarded as the most user-friendly querying method. In this paper, they developed mechanisms to defeat the challenges that limit the usefulness of forms, namely their restrictive nature and the tedious manual effort required to build them. Specifically, they introduce an algorithm to generate a set of forms automatically given the expected query workload [2].

3. E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, “Combining keyword search and forms for ad hoc querying of databases,” provides solution that combines keyword search with query form generation . The solution is, in advance to generate a lot of query forms automatically. User can find relevant query forms from a large number of pre-generated query forms by giving it several keywords as an input. The user inputs several keywords to find related query forms from a huge number of previously generated query forms but it is not suitable when the user does not have real keywords to describe the queries [3].

4. S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, “Automated ranking of database query results,” state that the results of a query is a popular aspect of the query model in Information Retrieval (IR) that we have grown to depend on. In contrast, database systems support only a Boolean query model. For instance, a selection query on a SQL database returns all tuples that satisfy the conditions in the query. Hence, the following two situations are not gracefully handled by a SQL system: Empty answer and many answer [4].

5. G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, “Query recommendations for interactive database exploration,” stated that now days there are numerous ways to explore the database in order to recommend the query forms. SQL queries play a vital role to recommend the user related queries as per their intendment. However they are not considering the quality of query forms much. Here is an additional method to recommend based on query results. The differences between these two strategies are each and every loop will provide the query component but in the other hand of previous recommendation is providing complete query [5].

6. C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, “Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia,” Dynamic faceted search is a type of search engines where relevant facets are presented for the users according to their navigation paths. Dynamic faceted search engines are similar to our dynamic query forms if we only consider Selection components in a query. However, besides Selections, a database query form has other important components, such as Projection components. Projection components control the output of the query form and cannot be ignored. Moreover, designs of Selection and Projection have inherent influences to each other [6].

7. L. Tang, T. Li, Y. Jiang, Z. Chen, “Dynamic Query Forms for Database Queries,” provides a solution that nontechnical users make usage of relational database which is a challenging task. Therefore, in modern years lots of

researches were focused on database interfaces to help users to query the relational databases without use of SQL. This paper provides a Dynamic Query Form system (DQF), is a query interface able to dynamically produce query forms for the users. Unlike conventional document retrieval, before distinguishing the final candidate, the users in database retrieval are mostly willing to execute several rounds of action [7].

3. System Architecture

3.1 Existing System

With the increase in web information and scientific databases, traditional query forms are unable to satisfy the needs of large and complex modern databases. Therefore, it is not easy to design a set of static query forms to answer various extemporary database queries on those complex databases. The flow of existing system architecture is shown in Fig. 1. When user fills static query form, the system executes the query, processes the query for result and further user is able to see the result. In this process, user is not provided with the feedback or requerying facility i.e. user statically interacts with the system. Recently proposed automatic approaches to generate the database query forms without user participation 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. One problem of the aforementioned approach is that, if the database schema is large and complex, user queries could be quite diverse. In that case, even if the system generates lots of query forms in advance, there are still user queries that cannot be satisfied by anyone of query forms. Another problem is that, when the system generates a huge number of query forms, how to let users find an appropriate and desired query form would be challenging. A solution that combines keyword search with query form generation is proposed. It automatically generates a query forms in advance. The user inputs several keywords to find relevant query forms from a large number of pre-generated query forms. It works well in the databases which have rich textual information in data tuples and schemas. However, it is not appropriate when the user doesn't have actual key words to describe the queries at the beginning, especially for the numeric attributes.

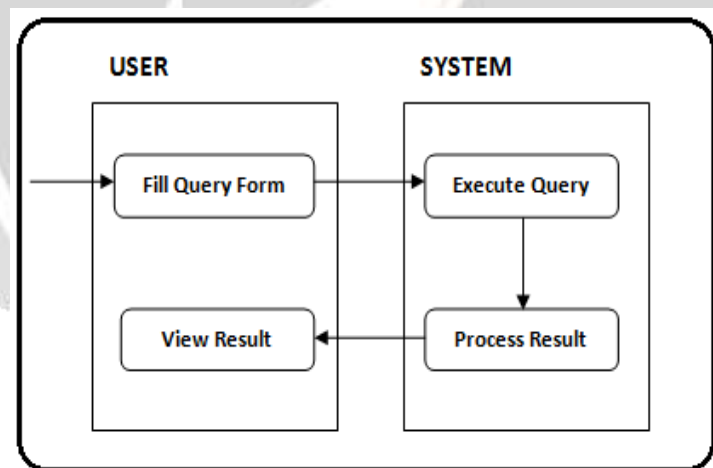


Fig-1: Existing System Architecture

3.2 Proposed 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 catch 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. Dynamic Query Form (DQF) is the proposed system which dynamically generates the query form as per the user's desire. The system is a solution for the query

interface towards the large and complex databases. DQF is a novel database query form approach, which provides simplicity to users in customizing the query forms. The generation of query form is an iterative process which is guided by the user. To help users for making decisions DQF captures user's preferences and ranks query form components. System auto generates ranking lists of form components and the user then adds the desired form components into the query form. Ranking is based on the captured user preferences. User can also fill the query form and fire queries to see the query result at each iteration. This could be continued till the user satisfies with the query results. The flow of proposed system architecture is shown in Fig. 2. It shows that there is a provision of querying/feedback, if the user is not satisfied with the previous query result.

The advantages of the above proposed system are as follows:

- The system helps user to dynamically generate query forms.
- As the system uses dynamic approach it provides higher success rate and simpler query forms compared with a static approach.
- It becomes easy for users to customize the query forms using the ranking based on user preferences.

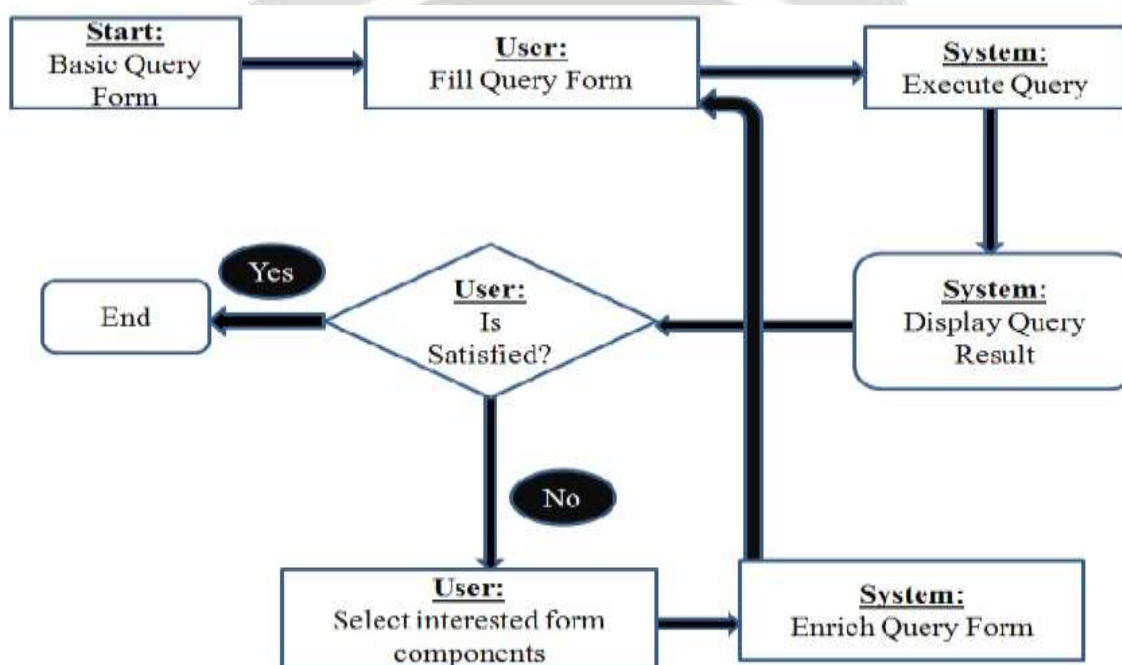


Fig -2: Flowchart of Dynamic Query Form

4. CONCLUSIONS

This paper review various existing methods used for the generation of query forms and their limitations. Query interface play an important role in determining usefulness of a database. It seems that, form based interface is a widely regarded as the most user friendly method for querying. In this paper, we propose a dynamic query form generation approach which is very effective in nature. It will help user to dynamically generate query forms i.e. at run time. The idea behind the proposed system is to use a probabilistic model which will rank form components based on the user choice/preferences. System will capture user preferences based on the historical queries and user runtime feedback such as click through. Compared with the traditional static approach for querying, surely dynamic approach will not only lead to higher success rate but also simpler query forms. It is also easier for users to customize query form. In future, the system can be extended to the use of non relational data. Also, multiple methods can be used to capture the user's preferences besides the click through method. For example, to use a textbox through which user inputs some keywords for querying.

5. ACKNOWLEDGEMENT

“Database Queries using Dynamic Query Forms” had been a wonderful subject to research upon, which leads one’s mind to explore new heights in the field of database management system. I would like to take this opportunity to express my heartfelt thanks to my guide Prof. P. N. Kalavadekar, Computer Engineering Department, for his esteemed guidance and encouragement, especially through difficult times. I am also very much thankful to Head of Department Prof. D. B. Kshirsagar, Computer Engineering Department, for helping us while selecting and preparing for dissertation work. I would like to thank all the faculties who have helped me during my work. Lastly, I am thankful to my friends who shared their knowledge in this field with me.

6. REFERENCES

- [1] M. Jayapandian and H. V. Jagadish, “Automated creation of a forms-based database query interface,” Proc. VLDB, vol. 1, no. 1, pp. 695709, Aug. 2008.
- [2] M. Jayapandian and H. V. Jagadish, “Automating the design and construction of query forms,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 10, pp. 13891402, Oct. 2009.
- [3] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, “Combining keyword search and forms for ad hoc querying of databases,” in Proc. ACM SIGMOD, Providence, RI, USA, Jun. 2009, pp. 349360.
- [4] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, “Automated ranking of database query results,” in Proc. CIDR, 2003 pp. 888899
- [5] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, “Query recommendations for interactive database exploration,” in Proc. SSDBM, New Orleans, LA, USA, Jun. 2009, pp. 318.
- [6] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, “Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia,” in Proc. WWW, Raleigh, NC, USA, Apr. 2010, pp. 651660.
- [7] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, “Dynamic Query Forms for Database Queries”, IEEE Trans. on Knowledge and Data Engg. Vol:PP No:9 sept. 2014.