IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

An Automated Approach for Database Queries by using DQF.

Prof. Jagruti A. Dandge[1], Mr. Subodh Patil[2], Miss. Shital Patil[3], Mr. Bhushan Kelhe[4],Mr. Shrikant More[5]

Assistant Professor, Department of Information Technology, P. V. G.'s College of Engineering, Nashik, India [1]

UG Student, Department of Information Technology, P. V. G.'s College of Engineering, Nashik, India [2,3,4,5]

Abstract: With rapid development of scientific databases and internet information databases are becoming very huge in size and complex in nature. These databases maintain large and heterogeneous data, with large number of relations and attributes. Thus there is need of such system which generate Query Forms dynamically according to the user's need at run time. The proposed system Dynamic Query Form i.e. DQF system going to provide a solution by the query interface in large and complex databases. In proposed system, the core concept is to capture user interests throughout user interactions and to adapt the question type iteratively. Every iteration consists of 2 sorts of user interactions: Query Form Enrichment and Query Execution. In Query Form Enrichment DQF would recommend a ranked list of query form component to user so user can select desired form components into current query form. In Query Execution user fills current query form and submit query, DQF is going to show result and take feedback from user on provided query results. A user would have facility to fill the query form and submit queries to view the query result at each process. So that a query form could be dynamically run till the user satisfies with the query results.

Keywords: Database, Query Form, Query form Generation, Query Execution, User Interaction.

I. INTRODUCTION

A database is only as functional thing that allows to be as it is. If a user is not capable to communicate to the database what user wishes from it, even the richest data store is complicated and ambiguous. Writing well-structured queries, in languages such as SQL and XQuery, 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 identify query parameters, is precious since it helps make data users with 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 an ambiguous way and its useability is restricted to a small set of fixed queries. Query form is one of the majority used user interfaces for querying databases. Traditional query forms are designed and predefined by developers or DBA in various information management systems. With the rapid development of web information and scientific databases, modern databases become very large and complex.

Dynamic question type system: DQF, a question interface that is capable of dynamically generating question forms for users. Different from ancient document retrieval, users in information retrieval area unit usually willing to perform several rounds of actions (i.e., refinement question conditions) before displaying final result to the candidates. The essence of DQF is to capture user interests throughout user interactions and to adapt the question type iteratively. Every iteration consists of 2 sorts of user interactions: it contains only a few primary attributes of the information. The essential question type is then enriched iteratively via the interactions between the user and our system till the user is satisfy with the question results. Goal of this paper is to show that the advantages of using dynamic query forms for database over the existing static query forms.

II. L ITERATURE SURVEY

A lot of research works focus on database interfaces which assist users to query the relational database without SQL. QBE (Query-By-Example) [6] and Query Form are two most widely used database querying interfaces. Current studies and works mainly focus on how to generate the query forms.

A. 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 [2], Cold Fusion [1] 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 [3].H.V. Jagadish proposed a system which allows end-users to customize the existing query form at run time [7]. If the database schema is very large, it is difficult for end user to find appropriate database entities and attributes.

B. Automated Creation of Forms:
M. Jayapandian presented a data-driven method [3]. 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.

C. Automating the design and construction of query forms: H.V. Jagadish presented a workload-driven method

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

[8].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.

D. Combining keyword search and forms:
A solution for aforementioned approaches [3], [8] is proposed in [9].It automatically generates a lot of query forms in advance. The user inputs several keywords to find relevant query forms from a large number of pre-generated query forms but it is not appropriate when the user does not have concrete keywords to describe the queries

III.EXISTING SYSTEM

Traditional query forms are designed and pre-defined by developers or DBA in various information management systems. With the rapid development of web information and scientific databases, modern databases become very large and complex. Therefore, it is difficult to design a set of static query forms to satisfy various ad-hoc database queries on those complex databases.

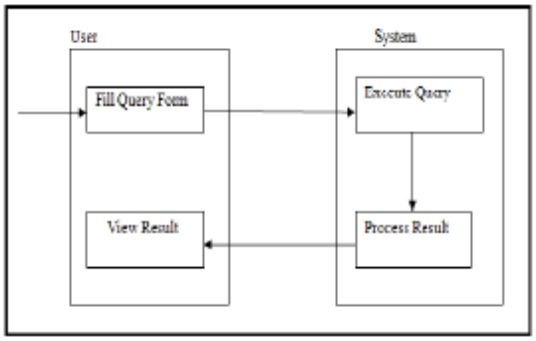

Fig -1: Existing System

Above system has following limitations :

- Query forms are designed and pre-defined by developers in information management systems.
- Difficult to design a set of static query forms to satisfy various ad-hoc database queries on complex databases.

The above disadvantages are motivated to design such DQF system which can solve mentioned limitations.

IV.PROPOSED SYSTEM

The proposed 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. This paper proposes DQF, a novel database query form interface, which is able to dynamically generate query forms. The essence of DQF is to capture a user's preference and rank query form components, assisting him/her to make decisions. The generation of a query form is an iterative process and is guided by the user. At each iteration, the system automatically generates ranking lists of form components and the user then adds the desired form components into the query form. The ranking of form components is based on the captured user preference. A user can also fill the query form and submit queries to view the query result at each iteration. In this way, a query form could be dynamically refined till the user satisfies with the query results.

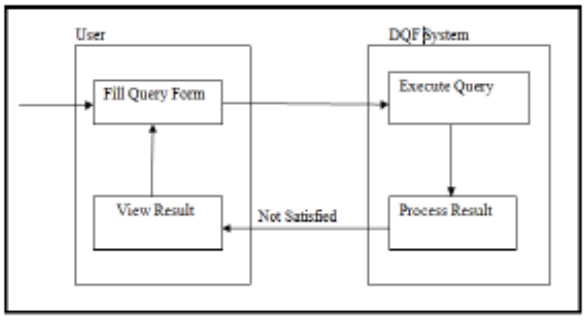

Fig -2: Proposed System

The above proposed system has following advantages:

- The proposed a dynamic query form generation approach which helps users dynamically generate query forms.
- 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.

V. SYSTEM MODULE

Fig 3 shows the flow of the system modules. There are four basic modules in DQF system are :

A. Query Form Enhancement :
In this module DQF recommends a ranked list of query form components to the user. So that the user selects the desired form components into the current query form.

B. Query Execution :
Firstly the user fills out current query form and submit a query. Then DQF executes the query and shows results

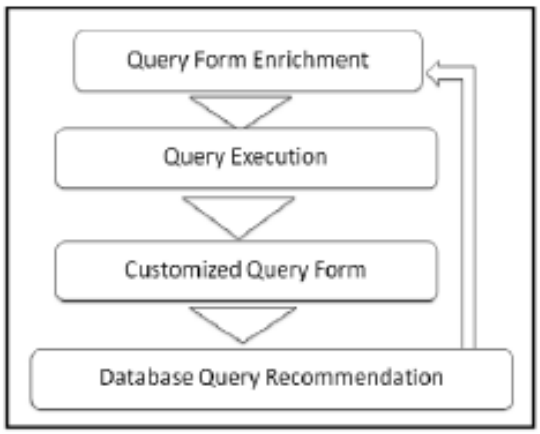

Fig -3: System Modules

C. Customized Query Form :
i. Visual interfaces are provided for developers to create or customize query forms. The problem of those tools is

DOI 10.17148/IJARCCE.2016.53105

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

that, they are provided for the professional developers who are familiar with their databases, not for end-users. In proposed a system which allows end-users to customize the existing query form at run time. However, an end-user may not be familiar with the database. If the database schema is very large, it is difficult for them to find appropriate database entities and attributes and to create desired query forms.

D. Database Query Recommendation :

Recent studies introduce collaborative approaches to recommend database query components for database exploration. They treat SQL queries as items in the collaborative filtering approach, and recommend similar queries to related users.

VI. PARTIAL RESULTS

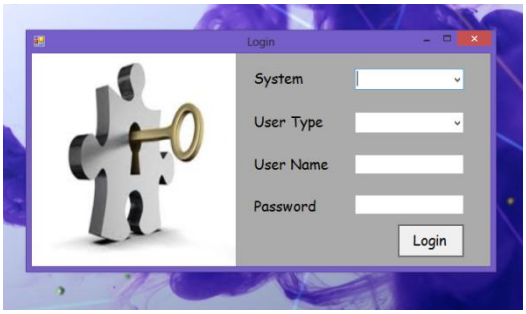

Fig -4: Login Page

When the Proposed System gets Connected to the Database then a login page is displayed for security purpose for providing authentication and authorization. First the system will select the appropriate database, after selecting it will provide some authentication by giving user type which should be selected, after selection of that username and password have to be entered for the security purpose the username and password may vary according to the usertype that have been selected once all the credential have been put into proper format and if the username and password are correct then user get's login into the system

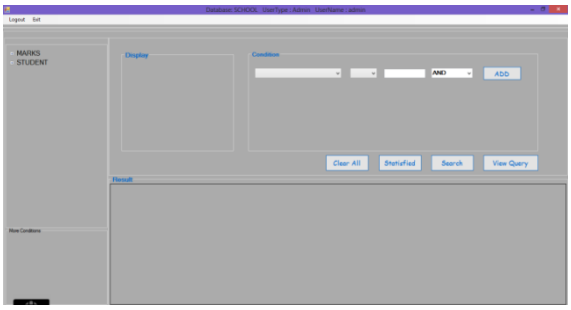

Fig -5:Partial result of proposed system.

VII. CONCLUSION

Dynamic Query Form plays 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, we have developed mechanisms to overcome the challenges that limit the usefulness of forms, namely their Structure Wellness. In this paper we propose an interactive query form generation approach which helps users to dynamically generate query forms. As future work, we will study how our approach can be extended to non- relational data. As for the future work, we plan to develop multiple methods to capture the user's interest for the queries besides the click feedback. For instance, we can add a text-box for users to input some keywords queries and can gain more user's interest in database.

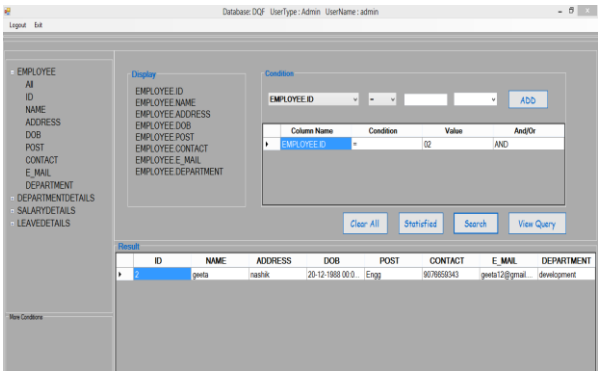

Fig -6: Dynamic Query Result.

The Result is obtained by Selecting the database and selecting the proper attributes. once the attributes are selected, select the particular property or attribute of the table and by providing conditions then the query is displayed user have more then one conditions to be satisfied. the user obtains it results by just clicking the conditions present on the buttons provided.

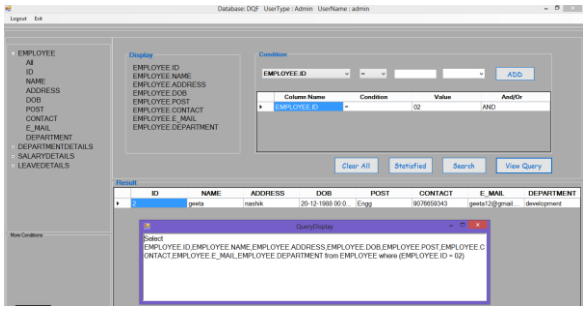

Fig -7:Query Generation

REFERENCES

1. ColdFusion.http://www.adobe.com/products/coldfusion/.
2. EasyQuery.http://devtools.korzh.com/eq/dotnet/.
3. M. Jayapandian and H. V. Jagadish. Automated creation of a forms- based database query interface. In Proceedings of the VLDB Endowment, pages 695–709, August 2008.
4. S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In CIDR, 2003.
5. G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1984.
6. M. M. Zloof. Query-by-example: the invocation and definition of tables and forms. In Proceedings of VLDB, pages 1–14, Framingham, Massachusetts, USA, September 1975.
7. 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.
8. M. Jayapandian and H. V. Jagadish. Automating the design and construction of query forms. IEEE TKDE, 21(10):1389– 1402, 2009.
9. 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:99 Year 2013

DOI 10.17148/IJARCCE.2016.53105