IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 5, May 2018

Survey on Automated Dynamic Query formation for Database Queries using Ranking

Yudhishthir D. Chavan[1], Prof. S.M. Shinde[2]

M.E Student, Computer Science and Engineering, SVERI's College of Engineering, Pandharpur, India [1]

Assistant Professor, Computer Science and Engineering, SVERI's College of Engineering, Pandharpur, India [2]

Abstract: A database is only as functional as query interface allows it to be. If a user is not capable to communicate to the database what user wishes from it, even the richest data store provides petite or no value. 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 adhoc way and its applicability 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 distinctive the final candidates.

The essence of DQF is to capture user interests throughout user interactions and to adapt the question type iteratively. Every iteration consists of two 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 satisfying with the question 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.

Keywords: DQF, Query Forms, Query Generation, Query Execution, Ranking.

I. INTRODUCTION

A database is only as functional as query interface allows it to be. If a user is not capable to communicate to the database what he or she wishes from it, even the richest data store provides petite or no value. 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 adhoc way and its applicability 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. The system proposes DQF, a novel database query form interface, which is able to dynamically generate query forms.

The system is proposed to have the following modules.

1. Query Form Enrichment.
   - DQF recommends a ranked list of query form components to the user.
   - The user selects the desired form components into the current query form
2. Query Execution.
   - The user fills out the current query form and submits a query.
   - DQF executes the query and shows the results.
   - The user provides the feedback about the query results
3. Customized Query Form.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 5, May 2018

The system provides visual interfaces for developers to create or customize query forms. problem of those tools is that, this s are provided for the professional developers who are familiar with their databases, not for end-users.

4. Database Query Recommendation.

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

II. LITERATURE SURVEY

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

Modified Query Form:

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.

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.

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 approach 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.

Combining keyword search and forms:

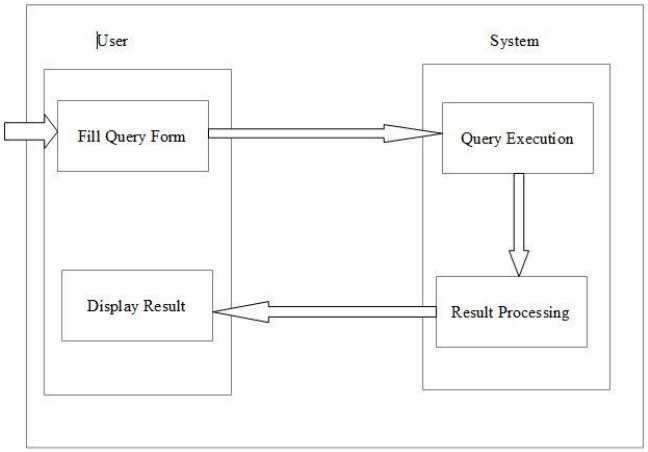
A solution for aforementioned approaches is proposed in. It automatically generates a lot of query forms in advance. The user inputs several keywords to find relevant query forms from a large number of pre-generated query forms but it is not appropriate when the user does not have concrete keywords to 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.

The databases of today have a huge and mixed collection of data in them. These contain a number of relations and attributes also. Earlier the developers or the database administrators created the traditional query forms. But now, with the increase in amount of data, it has become difficult to create such static query forms to satisfy the queries. Some existing database systems also allows user to create customized queries. But creation of such queries depends on the users editing manually. The large number of attributes resent would confuse the user unless user is not familiar with it.

Fig -1: Existing System



IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 5, May 2018

Above system has following limitations :

- Queries are designed and pre-defined by developers in information management systems.
- It is difficult to design a set of static query forms to satisfy various ad-hoc database queries on complex databases. These disadvantages are overcome 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.

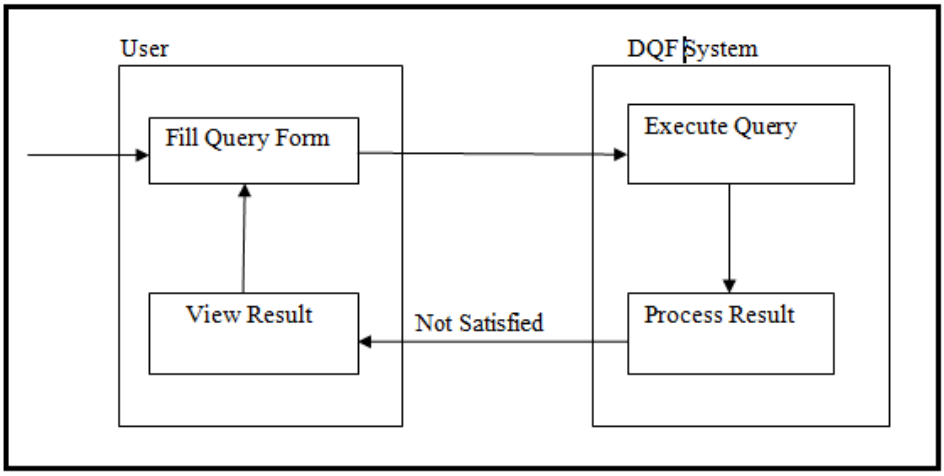


Fig -2: Proposed System

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.

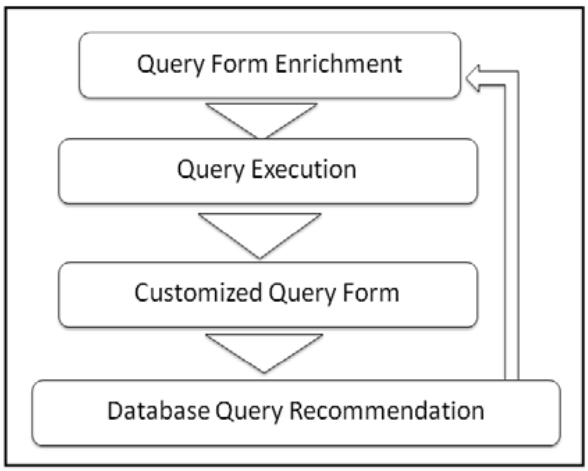


Fig -3: System Module

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 5, May 2018

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.

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

V. CONCLUSIONS

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, we have developed mechanisms to overcome the challenges that limit the usefulness of forms, namely their restrictive nature. 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.

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: pp no: 99 year 2013
[2] M. Jayapandian and H. V. Jagadish. Automated creation of a formsbased database query interface. In Proceedings of the VLDB Endowment, pages 695–709, August 2008.
[3] M. Jayapandian and H. V. Jagadish. Automating the design and construction of query forms. IEEE TKDE, 21(10): 13891402, 2009.
[4] K. Chen, H. Chen, N. Conway, J. M. Hellerstein and T. S. Parikh. Usher: Improving data quality with dynamic forms. In proceedings of ICDE Conference, pages 321-332, Long Beach, California, USA, March 2010.
[5] W. B. Frakes and R. A. Baeza-Yates. "Information Retrieval: Data Structures and Algorithms". Prentice-Hall, 1992. [15] T. Joachims and F. Radlinski. "Search engines that learn from implicit feedback". IEEE Computer (COMPUTER), 40(8):34–40, 2007.
[6] N. Khoussainova, M. Balazinska, W. Gatterbauer, Y. Kwon, and D. Suciu." A case for a collaborative query management system". In Proceedings of CIDR, Asilomar, CA, USA, January 2009.
[7] Q. T. Tran, C.-Y. Chan, and S. Parthasarathy. "Query by output". In Proceedings of SIGMOD, pages 535–548, Providence, Rhode Island, USA, September 2009.
[8] Jayashri M. Jambukar." Survey Paper on Creation Of dynamic query Form for mining highly Optimized transactional databases". International Journal of Computational Engineering Research||Vol, 04||Issue, 3||, March 2014.
[9] Random Query Formulation for Database Queries, Gopi Krishna Lakksani, BalakrishnaNayudari, Department of Computer Science, Mandapalle University, AP. 8 August 2014.