International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 5 (Sept - Oct, 2016), PP 88-92

Dynamic Query Forms for Database Queries

[1] Miss. Minal A. Deshmukh,[2]Prof. A. B. Deshmukh
[1]Computer Science and Engineering and perceiving masters in Information Technology,
[2]Dept. of Computer Science and Engineering Sipna College of Engineering and Technology, Amravati
[1]minal2392.d@gmail.com

***Abstract*— Modern scientific databases and web databases maintain large and heterogeneous data. These real-world databases contain over hundreds or even thousands of relations and attributes. Traditional predefined query forms are not able to satisfy various ad-hoc queries from users on those databases. This paper proposes, 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.**

Keywords: **DQF, Heterogenous data, Query Form, User Interaction, Query Form Generation.**

I. INTRODUCTION

Query form is one of the most widely 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. 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 ad-hoc way and its applicability is restricted to a small set of fixed queries.

Many existing database management and development tools, such as EasyQuery [1], Cold Fusion [2], SAP provide several mechanisms to let users create customized queries on databases. Here, the creation of customized queries totally depends on users' manual editing [3] which leads to users' confusion because they being non- technical are not familiar with the database schema. 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 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 satisfying with the question results.

II. LITERATURE REVIEW

How to let non-expert users make use of the relational database is a challenging topic. A lot of research works focus on database interfaces which assist users to query the relational database without SQL. Current studies and works mainly focus on how to generate the query forms.

1. Customized Query Form: M. Jayapandian and H. V. Jagadish [4] proposed a system which allows end-users to customize the existing query form at run time. 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 who are familiar with their databases, not for end-users.

2. Automatic Static Query Form: Recently, [5] [7] proposed automatic approaches to generate the database query forms without user participation. [5] 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.

3. Auto completion for Database Queries: In [8], [9], novel user interfaces have been developed to assist the user to type the database queries based on the query workload, the data distribution and the database schema. Different from our

work which focuses on query forms, the queries in their work are in the forms of SQL and keywords.

4. Query Refinement: Query refinement is a common practical technique used by most information retrieval systems. It recommends new terms related to the query or modifies the terms according to the navigation path of the user in the search engine. But for the database query form, a database query is a structured relational query, not just a set of terms.

5. Database Query Recommendation: In this approach, [10] [11] treat SQL queries as items in the collaborative filtering approach, and recommend similar queries to related users.

However, they do not consider the goodness of the query results.

6. Dynamic Data Entry Form: K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh. Usher [12] develops an adaptive forms system for data entry, which can be dynamically changed according to the previous data input by the user. Our work is different as we are dealing with database query forms instead of data-entry forms.

7. Active Feature Probing: Zhu et al. [13] develop the active featuring probing technique for automatically generating clarification questions to provide appropriate recommendations to users in database search. Different from their work which focuses on finding the appropriate questions to ask the user, DQF aims to select appropriate query components.

III. ANALYSIS OF PROBLEM

A. Existing system:

With the rapid development of web information and scientific databases, modern databases become very large and complex. In natural sciences, such as genomics and diseases, the databases have over hundreds of entities for chemical and biological data resources. Many web databases, such as Freebase and DBPedia, typically have thousands of structured web entities. Therefore, it is difficult to design a set of static query forms to satisfy various ad-hoc database queries on those complex databases.

B. Disadvantages of existing system:

- However, the creation of customized queries totally depends on users' 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.

C. Proposed system:

In this paper, we propose 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 related to the database.

Fig 1: Home page

Modules:

1. DBA
2. User
3. Databases

The working of each of the above modules is as follow:

1. DBA (Database Administrator):

The DBA is responsible for executing all the complex queries that non-technical people or users are unable to perform. His other main functnality is to approve the new user request who wants to create an account on this site to store his data. Along with this, he has the authority to give some certain different permission to the new user or he can also update the permissions he has given to any of the users on this site.

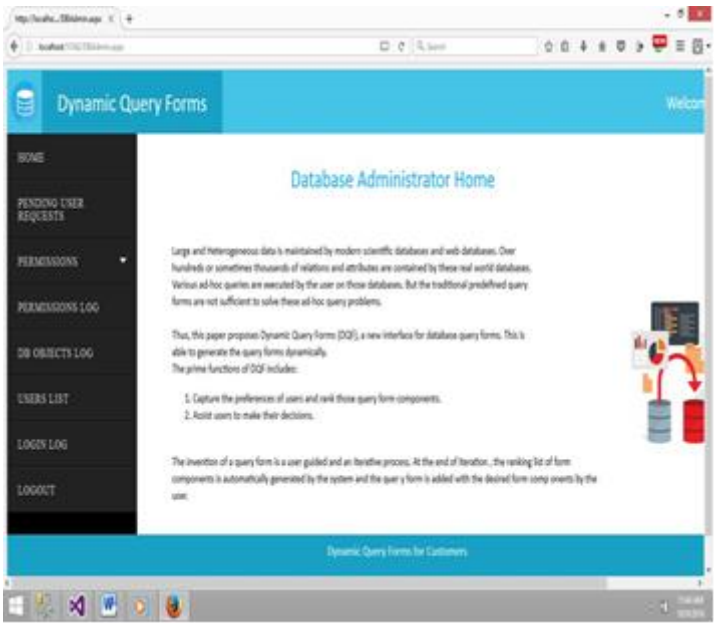

Fig 2: DBA's Account

2 User:

User can either be a technical person or a non-technical Person as well. For a technical person it is very easy to deal with the database and all of its queries but a non-technical one finds it quite difficult to deal with the database and its queries. So, we have provided an interactive and very user friendly platform for users to deal with database and to store or retrieve their data from database. User can login in to our site through the new user request button. Once a new user created his account on the site and it is get approved by the DBA, new user will get a mail of his user d and password of his main account and also n other mail of his database's user id and password in his mailbox. After getting all the above details, he can then deal with his account as well as the database in which he can store his data.

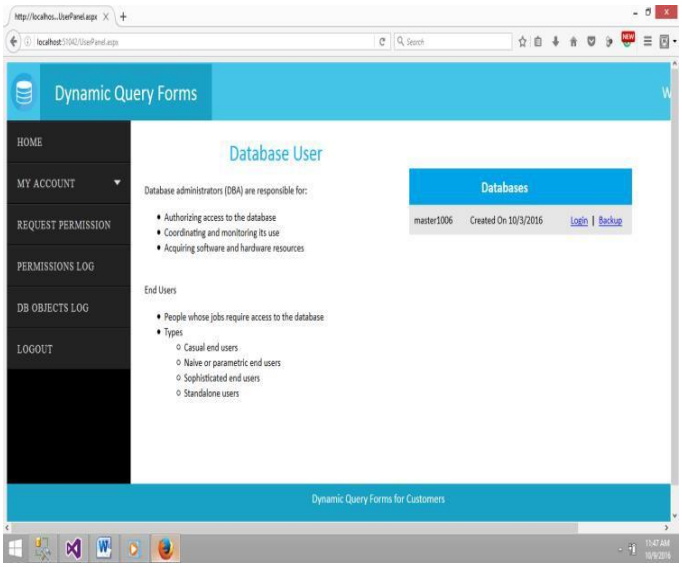

Fig 3: User Account

3. Databases:

Database is a place where any user can store his data or information into it. Database is one of the most used storage mediums . Different database queries are available to store, retrieve, update or delete data from the database. Fig 4 shows the login page of the database of a particular user from which he can login into his database and can store his data in it.

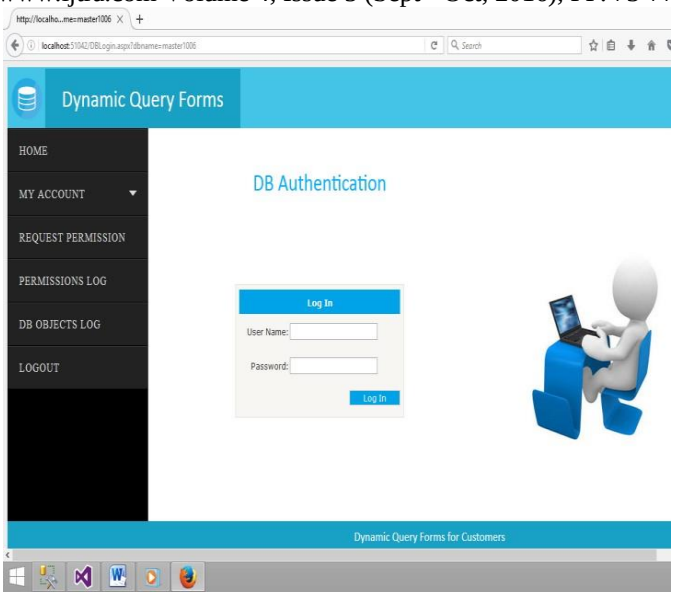

Fig 4: Database login page

Once the user login into his database, he can perform some certain operations with database. As we can see in the fig 5, that the DBA has provided the user some permissions like to create a table and a view as well. So, now, in this database, the user can perform all the database operations he wants to perform with or without the help of the DBA.

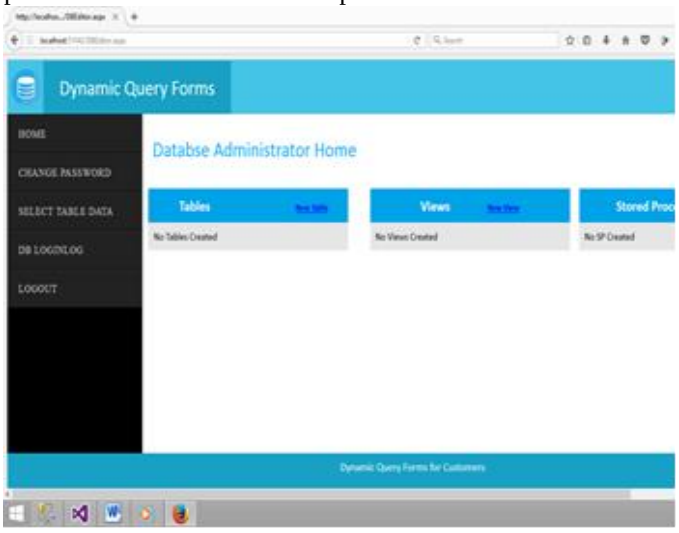

Fig 5: User's Database

Password recovery is a must needed concept in any of the systems. In case. A user forgets his password, he can recover it back with the help of this password recovery function.

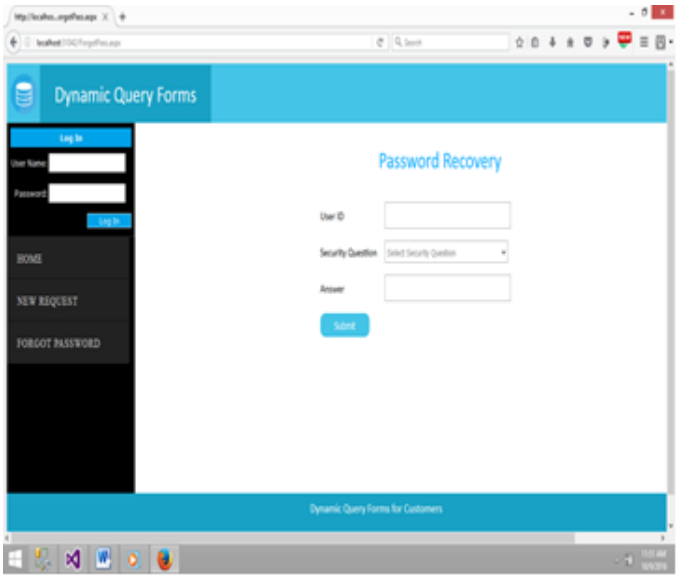

Fig 6: Password Recovery

Advantages of proposed system:

1) 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.

2) 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. RESULTS AND DISSCUSION

The comparison gives us different systems with their disadvantages which is shown in the following table.

Title	Features	Problems
Query-By Example	Provide a simple interface for a user to enter queries.	1.Ralational Completeness 2.Ordering problem
Automated Ranking of Database Query	To build a generic automated ranking infrastructure for SQL databases.	1) Ranking function might fail to perform.
Instant Response Interfaces	Interface developed to assist the user to type the database queries.	1) The users information need is explicit 2) Attempt to apply efficient index structures for basic keyword querying.
Forms-based Database Query Interface	Automatic approaches to generate the database query forms without user participation.	1) Not appropriate when the user does not have concrete keywords to describe the queries.
Form Customization	A system which allows end-users to customize the existing query format run time.	1) Database schema is very large so it is difficult to create desired query forms.
Forms for Ad Hoc Querying of Databases	Form Based Interfaces and Keyword Search.	1) Hard for users, uncomfor table with a formal query language.

CONCLUSION

This paper proposes a dynamic query form generation approach which helps users dynamically generate query forms. The key idea is to use a probabilistic model to rank form components based on user preferences. We capture user preference using both historical queries and run-time feedback such as click through. Experimental results show that 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.

ACKNOWLEDGEMENT

I would like to thank my guide Prof. A. B. Deshmukh for fulfilling my research work on DQF. Moreover I thank for the facilities provided by Sipna College of Engineering and Technology Amravati for providing me necessary article for completing my study on this topic.

References

[1] EasyQuery. http://devtools.korzh.com/eq/dotnet/.

[2] ColdFusion.http://www.adobe.com/products/coldfusion/.

[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] 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.

[5] 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.

[6] 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.

[7] M. Jayapandian and H. V. Jagadish. Automating the design and construction of query forms. IEEE TKDE, 21(10):1389–1402, 2009.

[8] N. Khoussainova, Y. Kwon, M. Balazinska, and D. Suciu. Snipsuggest: Context-aware autocompletion for sql. PVLDB, 4(1):22–33, 2010.

[9] A. Nandi and H. V. Jagadish. Assisted querying using instant response interfaces. In Proceedings of ACM SIGMOD, pages 1156–1158, 2007.

[10] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis. Query recommendations for interactive database exploration. In Proceedings of SSDBM, pages 3–18, New Orleans, LA, USA, June 2009.

[11] 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.

[12] 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.

[13] S. Zhu, T. Li, Z. Chen, D. Wang, and Y. Gong. Dynamic active probing of helpdesk databases. Proc. VLDB Endow., 1(1):748–760, Aug. 2008.

[14] G. Soundararajan, D. Lupei, S. Ghanbari, A. D. Popescu, J. Chen, and C. Amza. Dynamic resource allocation for database servers running on virtual storage. In FAST, 2009.

[15] S. Aulbach, T. Grust, D. Jacobs, A. Kemper, and J. Rittinger. Multi-tenant databases for software as a service: Schema-mapping techniques. In SIGMOD, 2008.

[16] M. Hui, D. Jiang, G. Li, and Y. Zhou. Supporting database applications as a service. In ICDE, 2009.

[17] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In CIDR, 2003.