ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 7, July 2019

Dynamic Query Forms for Database Queriesby using Users Feedback

Rajan S. Jamgekar[1], Ranjeet B. Parihar[2], Archana R. Sawant[3]

Assistant Professor, Department of Computer Science Engineering, NBN Sinhgad College of Engineering kegaon, Solapur, Maharashtra, India[1]

Assistant Professor, Department of Computer Science Engineering, SKN Sinhgad College of Engineering, Korti, Pandharpur, Maharashtra, India[2]

P.G. Student, Department of Computer Science Engineering, SKN Sinhgad College of Engineering, Korti, Pandharpur, Maharashtra, India[3]

ABSTRACT: Recent scientific and web databases maintain huge and different types of data. Due to the enhancement of web databases and scientific databases, user is not able to get required results with the predefined static query forms. To avoid such a problem, Dynamic Query Forms (DQF)are developed in the present paper. In this newly defined Dynamic Query Form, user can execute the query by selecting the desired form component and then he/she should submit the query for execution. At the time when user is searching for the required result, the system is able to catch the user interest by using the user's feedback. According to this feedback, system then recommendsa ranked query to the user which helps in finding the desired result as early as possible. A user can perform the form component selection and query submission operation iteratively until he/she is satisfied with the query results.Precision and recall are the parameters used in measuring the goodness of the query result.

KEYWORDS:Query Form; Query Execution; User's feedback; Dynamic Query Forms(DQF)

I. INTRODUCTION

Usefulness of database depends on its query interface. If a user is unable to express properly to the database what he/she desire from it, even the richest data store will provide information with little or no value. Hence, user needs to write queries in a structured manner in languages such as SQL and XQuery for retrieving the required information from the database. This is not an easy task since the users may not have the technical knowledge and the underlying schema of the database. Formbased query interface presents a better solution for extracting relevant information from large and composite data structure.

In form based searching, the end user is only requiredto fill the text boxes without the need of having any knowledge about official query language or the database schema. However, the applicability of those predefined query forms is restricted to a small set of fixed queries. Recentdevelopments in web informatics and scientific databases have led to generation of modern databases which are huge (housing thousands of entities and attributes) and complex. Retrieving information from this heterogeneous database with static query forms is quite a difficult and inefficient task to perform. Many database management tools provide mechanisms to design predefined query forms. The process is complex because user must manually edit to design predefined query forms. If a user is unaware of the database schema, then handling attributes in the process of designing predefined query forms becomes too complex to perform. This paper proposes a Dynamic Query Form (DQF) system, an interface which is capable for generating query forms for the user at runtime. Before identifying the final candidate, the users need to execute several rounds of action for database retrieval. In this aspect, the proposed approach is different from the traditional document retrieval. Capturing the user's

DOI: 10.15680/IJIRCCE.2019. 0707002

interest during the interaction session and systematically adapting the query forms iteratively are some of the important features of dynamic query form.

II. Related work

Currently, the challenge is that the novice users without sufficient technical knowledge should be able to perform operations on the relational databases with ease. Hence,recent research is heavily focussed on database interfaces whichassist the users for querying on relational databases without the use of SQL or any other query language. Normally, thereare twointerfaces for querying on databases: 1] Query–By-Example(QBE)[9]: This is a high-level data base management language that provides a convenient and unified style to query, update, define, and control a relational data base. 2] Query Form:This interface is used for interacting with scientific and web databases.The database tools such as Easy Query [4], Cold Fusion, and Microsoft Access and so on helps the user to design and generate the query form. Only professional users can use these tools so the users who are unaware with the database schema are unable to find the required entities and attribute for generation of the needed query form. JayapandianandJagadish [5] proposed an automatic generation of query forms without the participation of the user. It initially finds the data attributes which are queried based on the database schema and data instances. Based on the selected attributes, the Query form is generated. In the work ofJayapandian andJagadish [2], the workload driven method is proposed where the clustering algorithm is used on historical queries for finding the representative queries. These representative queries are then used for form generation. One problem with this method is that when database is large and complex, then user is unable to create the required query. In order to solve this, a large number of query forms are generated previously and user will select the required query form at runtime. However, this may lead to confusion on the user part to search the required query form. To solve such a problem, Chu et al.[6] proposed the keyword based search where user can enter the keyword and system will suggest the query form to user according to the keyword.The best examples of this approach are search engines wherein user inputs certain keywords to obtain relevant information. An efficient technique such as query refinement technique is used normally for query retrieval systems [7].In this technique, when the user is searching for information, the system suggests something relevant to the query or modifies the suggested terms when there is change in the users search. This mechanism gives the relevant facets to user according to their navigation path. Dynamic faceted are like DQF when considering only the selection operation of query form. In Dynamic Query Form, along with the selection component, projection component also plays the main role for the query results [8].

III. PROPOSED METHODOLOGY

The methodology adopted in the present work can be broken down into three important steps as:

A. *Query Execution*

- Initially, the basic Query Form is provided to the user.
- User will select the required database from given database list.
- Forthe selected database, system will provide the table list.
- User is required to select the table and attributes to run the query.
- After selecting the required component, user can run the query to obtain the result.

B. *Query Enhancement*

After running the query, in case, if user is not satisfied with the result, then he/she can select different components and conditions to enhance the existing query form. This process is repeated till the user is satisfied with the results.

C. *Query History*

The previously executed queries are stored in the table. There is a choice for the user to select the required queries and obtain the result. Additionally, the user can also use the textbox search for queries.

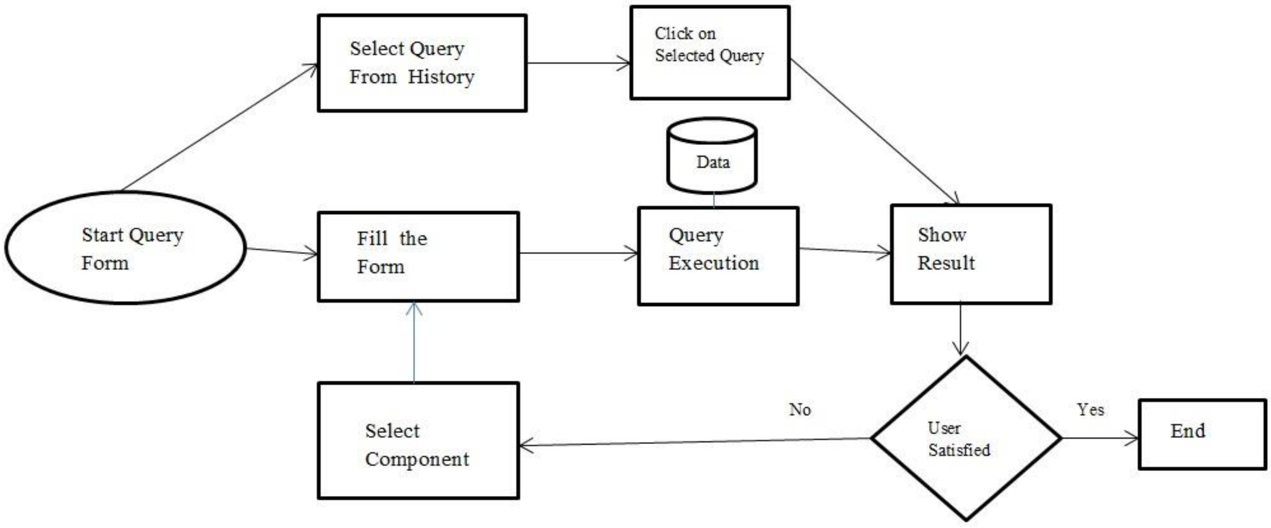


Figure 1.System Block Diagram showing flow of data

The block diagram of the proposed system is shown in Fig.1.The entire process is repeated until the user is satisfied with the end result..

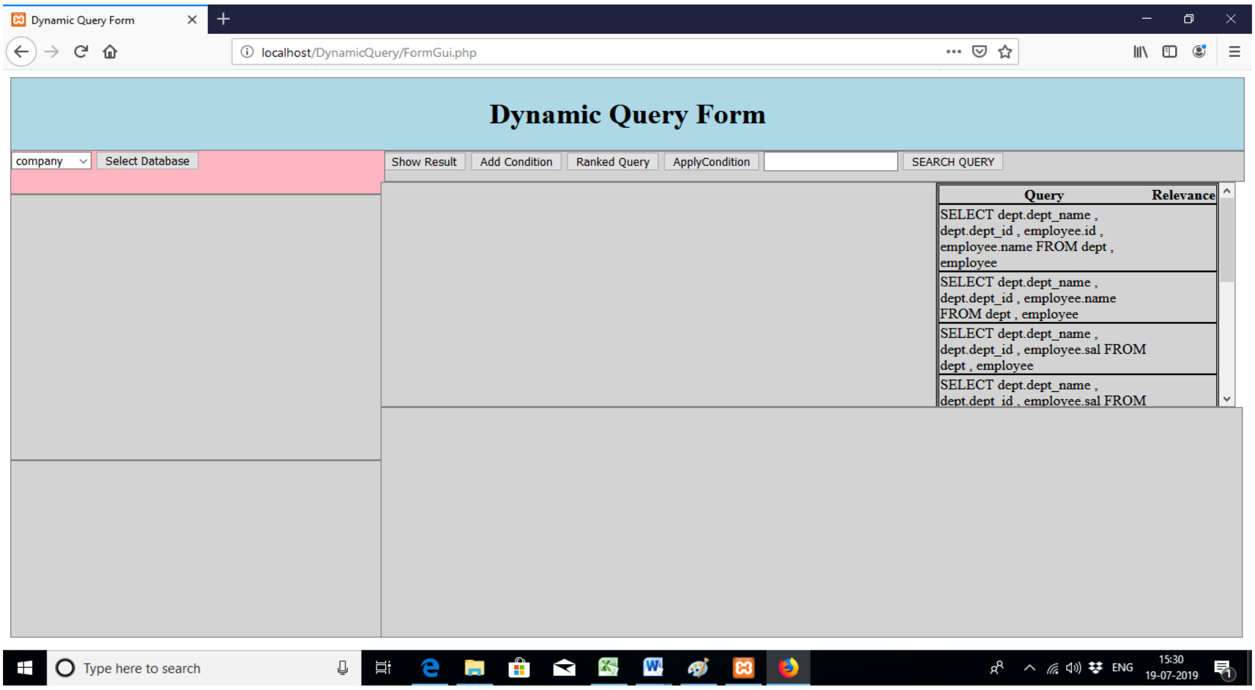
IV. EXPERIMENTAL RESULTS

In the present work, an approach of dynamic generation of query form is developed. In this approach, a user can select the required database, the required table and attributes and then can run the query for execution. The submitted queries are stored in history. To enhance the ease of use, the textbox is provided to user for inputting some text. According to the user input, the ranked queries from history are given to user. For analyzing the performance, different types of queries at different times in history are stored. The parameters such as precision and recall are evaluated to determine the measure of the goodness of the query form. These parameters are evaluated using following formulae:

$$\text{Precision} = \frac{\text{True Predicted Queries}}{\text{True Predicted Queries} + \text{False predicted Queries}}$$

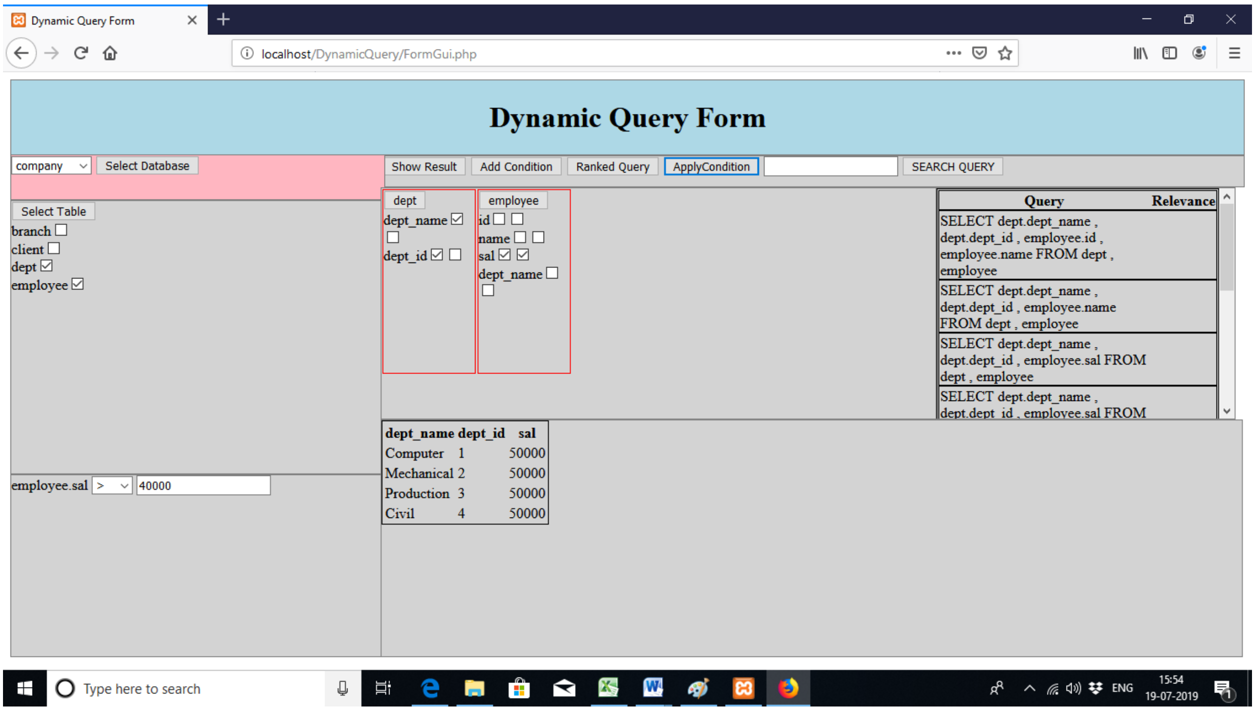
$$\text{Recall} = \frac{\text{True Predicted Queries}}{\text{Total matched Queries}}$$

The snapshots of the basic query form and query execution are shown in Figs. 2 and 3, respectively.



Query	Relevance
SELECT dept.dept_name , dept.dept_id , employee.id , employee.name FROM dept , employee	
SELECT dept.dept_name , dept.dept_id , employee.name FROM dept , employee	
SELECT dept.dept_name , dept.dept_id , employee.sal FROM dept , employee	
SELECT dept.dept_name , dept.dept_id , employee.sal FROM	

Figure 2. Screenshot of basic query form



branch □
client □
dept ☑
employee ☑

dept: dept_name ☑ □ dept_id ☑ □

employee: id □ □ name □ □ sal ☑ ☑ dept_name □ □

employee.sal > 40000

Query	Relevance
SELECT dept.dept_name , dept.dept_id , employee.id , employee.name FROM dept , employee	
SELECT dept.dept_name , dept.dept_id , employee.name FROM dept , employee	
SELECT dept.dept_name , dept.dept_id , employee.sal FROM dept , employee	
SELECT dept.dept_name , dept.dept_id , employee.sal FROM	

dept_name	dept_id	sal
Computer	1	50000
Mechanical	2	50000
Production	3	50000
Civil	4	50000

Figure 3. Screenshot of query execution

The results of the proposed approach in terms of precision and recall as performance parameters are shown in Figs. 4 and 5 respectively

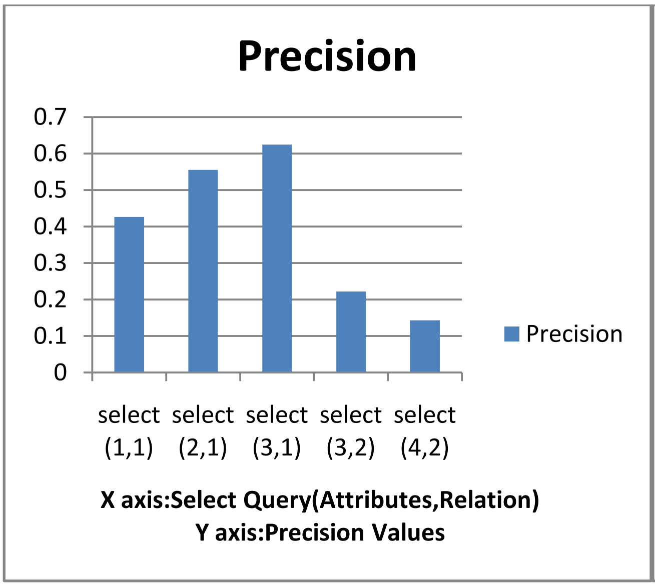


Figure 4.Performance of precision parameter

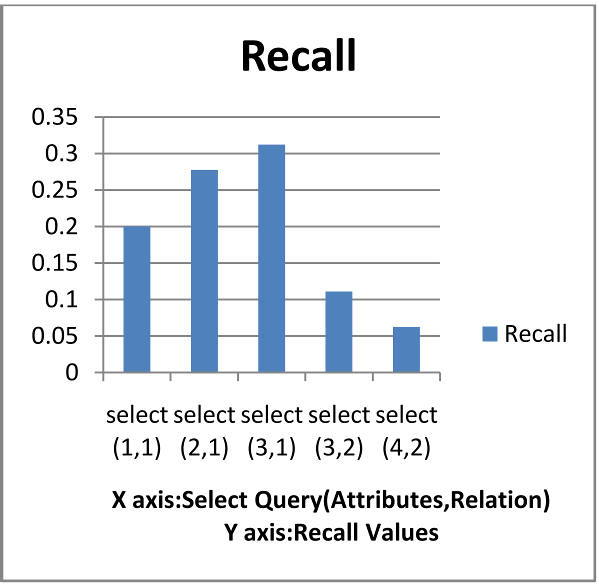


Figure 5. Performance of recall parameter

V. CONCLUSION

In the present work, a dynamic query form generation approach is developed with an aim to help novice users while interacting with complex databases.The developed approach seems to be better in performance when compared with the static approach. This iterative approach takes into account the user's feedback. The user can perform the form component selection and query submission operation iteratively until he/she is satisfied with the query results. Precision and recall are the parameters used in measuring the goodness of the query result. The ranking of form components will make it easier for users to customize the query forms.

REFERENCES

1. Liang Tang, Tao Li, and Zhiyuan Chen,' Dynamic Query Forms for Database Queries', IEEE Transaction On Knowedge and data Engineering, vol.26, no.9, September 2014.
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. Korzh.com. (2005) Easy Query [online]. Available: http://devtools.kortzh.com/eq/dotnet/
4. M. Jayapandian and H. V. Jagadish , ' Automated creation of a forms-based database query interface', Proc. VLDB, vol. 1, no. 1 , pp. 695-709, Aug. 2008.
5. 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, pp. 349360, Jun.2009.
6. W. B. Frakes and R. A. Baeza-Yates, Information Retrieval: DataStructures and Algorithms. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.
7. S. B. Roy, H. Wang, U. Nambiar, G. Das, and M. K. Mohania,'Dynacet: Building dynamic faceted search systems over databases', in Proc. ICDE, Shanghai, China, Mar. 2009,pp. 1463–1466.
8. M. M. Zloof, 'Query-by-example: The invocation and definition of tables and forms', in Proc. VLDB, Framingham, MA, USA, Sept.1975, pp. 1–14.