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

Dynamic Query Form for Relational Database

Priyanka Gupta[1], Rakesh Kumar Singh[2], Rambha Agrahari[3]

M.Tech Scholar, CSE, Kamla Nehru Institute of Technology, Sultanpur, U.P., India[1,3]

Associate Professor, CSE, Kamla Nehru Institute of Technology, Sultanpur, U.P. , India [2]

Abstract: There is a rapid development of the morden scientific database and internet database. These databases have thousand numbers of relations and attributes. So static query form does not work with a large number of relations and attributes, as they do not satisfy many ad-hoc database queries on the real world databases. Thus there is a requirement of a system which will satisfy the user requirement and generate the query form dynamically at the run time. In this paper we propose the Dynamic Query Form (DQF), which is going to provide the solution of the large and heterogeneous database. DQF captures the user's interest and provide the rank query form component, which is usefull for the user to make his/her decision. DQF is the iterative process and each iteration is guided the user. In each iteration system generates the ranking list of form component automatically and user can add any component according to his/her requirement. The component ranking is generated by capturing the user preference. Now user can fill the enrich query form and execute the query and the result will be shown to the user. This process is repeated until the user is satisfied.

Keywords: Query form, Query execution, User interaction, Query form generation.

I. INTRODUCTION

A database is not usefull until a query interface does not provide to handle the database. If the user can not communicate with the database, there is no use of that database. Structured query can be written in SQL and XQuery but it is not good for that type of user who does not know anything about SQL. For this reason, Query form is the easiest way for that type of user because they just fill the blanks in the form. That is the reason Query form is one of the most widely used technique for providing the interface to the database. Traditional Query form are predefined by the developers or DBA which does not satisfy the user ad-hoc query as the database become large and complex.

There are many existing database development tools such as Easy Query [1], Cold Fusion [2], Microsoft access, which will provide mechanism to customize the query form by the user but if the user is not familiar with these database tools then it is not easy to work with large database. DQF will generate the query form dynamically and each iteration in DQF consist Query form enrichment and Query execution. The basic query forms are provided to the user and this form will be enriched iteratively using the user and system interactions till the user is satisfied with the query result.

II.QUERY FORM TECHNIQUE

How non expert user makes use of the relational database is a challenging topic. Non expert user can interact with the database using Query By Example (QBE) [3] and Query Form.

A. Query By Example:

It is a database query language for relational database. It is the first query language used to create visual tables where the user can enter commands, example elements and condition.

Example form contacts-

.....Name: X
.....Address:
.....City:
.....State: Y
.....Pin code:

Resulting SQL :

```
SELECT *FROM CONTACTS
WHERE NAME='X'
AND STATE='Y';
```

B. Customized Query Form:

There are many existing tools for the developers to create or customize the query form such as SAP, Microsoft access and so on. The problem with these tools is that, they are for the professional developers who are familiar with their database. These tools are not for end user who does not know anything about the database.

C. Automatic Static Query Form:

It is the data driven technique, which first finds a set of data attributes which are mostly queried. Then based on these attributes query form will be generated. So it will applied clustering algorithms on the historical queries to find the new queries. Then query form generated based on these new queries. The problem with this technique is that, if the database is very large and complex then user query could be diverse. In such cases, if we generate lots of forms in advance still it does not satisfies the user requirements. Another problem is that if we generate lots of query forms them how user will know which form will satisfies his/her requirement.

III.EXISTING SYSTEM

Traditional query forms are predefined by the developers or DBA. As the database become large and complex, static query forms does not satisfy the user requirements.

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

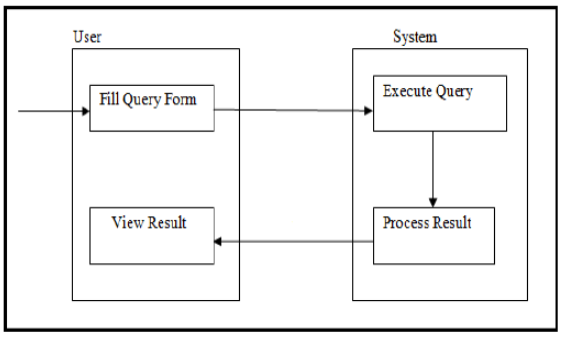

Fig .1. Existing System [12]

IV.PROPOSED SYSTEM

The proposed system is Dynamic Query Form which is the iterative process until the user is not satisfied. In DQF, a basic query form is provided to the user then user can fill the query form and execute the query. If user does not satisfy with the result then he will enrich the query form and execute again.

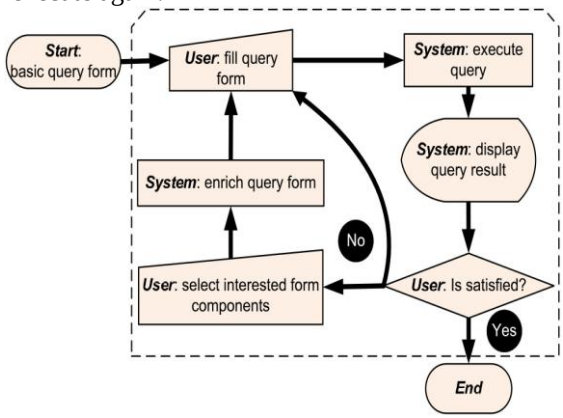

Fig .2. Flow diagram of DQF [4]

V.LITERATURE REVIEW

Dynamic query form includes the query form, query result and ranking metrics of the form component.

A. Query Form Interface:
This includes the query form and query result
(1) Query Form:
Each of the query form represents a SQL query template.
A query form F is define as $(A_f, R_f.\sigma_f, \triangleright\triangleleft(R_f))$
Which represent SQL query template as

$$F = (\text{ Select } A_1, A_2, A_3 \ldots\ldots\ldots A_f$$

$$\text{from } \triangleright\triangleleft(R_f)$$

$$\text{where } \sigma_f).$$

Where $(A_1, A_2, A_3 \ldots\ldots\ldots A_f)$ are the k attributes which are projected to the user and $\triangleright\triangleleft(R_f)$ is the join of the relations and σ_f is the condition which is provided by the end user. $\triangleright\triangleleft(R_f)$ is automatically constructed according to the foreign keys in R_f. R_f is determine by A_f and σ_f. R_f is the union set of relation which contains at least one attribute of A_f or σ_f. A_f and σ_f are visible to the user. Ad-hoc join does not handle by the dynamic query form because join is not the part of the query form and invisible to the user.

(2) Query Result:
As we mention that now a day's database become so large and complex so the result of the dynamic query form will give huge amount of data instances. To avoid this problem, we output a compressed result to show a high level view of the query result.

In the compressed table, each instance represents a cluster of actual data instances. Now user can click on any instance in the compressed table to view the detailed view. There are many one-pass clustering algorithms to generate the compressed view for example incremental data clustering [5].

Another important use of query result is that, End user is not interested to provide their feedback explicitly. So using the compressed view, we can collect the user feedback so that we can estimate the goodness of the query form.

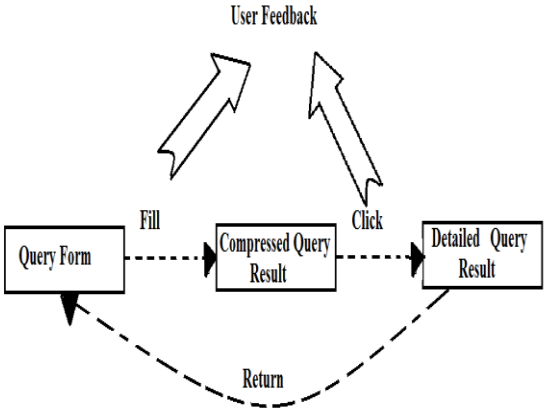

Fig .3. User action [9]

B. Ranking Metrics:
Query forms are designed so that it will return the user's required result. To estimate the quality of the query form result there are two traditional measure, Recall and Precision [6].

According to query form, different query will be executed and will give the different output. So we get different Recall and Precision value. Because of this we are using Expected Precision and Expected Recall to find the performance of the query result. Expected Precision is the expected part of the query result, which the current user wants.

Expected Recall is the expected part of the user interested query result instance, which is returned by the query form. User interest is calculated based on the click through on the query result which is displayed to the user.

DOI 10.17148/IJARCCE.2016.5303

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

There are many symbol and notations which are used to calculate the precision and recall and F-measure.

Table 1: Notations and Symbols [10]

F	Query form
R_f	Set of relations involved in F
A	Set of all attributes in $\triangleright\triangleleft(R_f)$
A_f	Set of projection attributes of query form F
$A_r(F)$	Set of relevant attributes of query form F
σ_f	Set of selection expressions of query form F
$o\rho$	Set of relational operators in selection
d	Data instance in $\triangleright\triangleleft(R_f)$
D	The collection of data instances in $\triangleright\triangleleft(R_f)$
N	Number of data instances in D
d_{A_1}	Data instance d projected on attribute set A_1
D_{A_1}	Set of unique values D projected on attribute set A_1
Q	Database query
D_Q	Result of Q
D_{uf}	User feedback as clicked instances in D_Q
α	Fraction of instances desired by user

C. Estimation of ranking score:

The ranking score estimation consist two phases. Ranking projection form component and Ranking selection form component.

(1) Ranking projection form component:

In dynamic query form, the projection component has two level ranking. The first level ranking is to rank the entity. In this level, we provide the procedure how to rank the entity. The second level of ranking provides a ranking list of attributes of that entity. In this process, first we define how to rank the attributes of entity and then how to rank the entity.

(2) Ranking selection form component:

The selection of attributes will be meaningless if the selection attributes are not relevant to the current projected entities. Therefore, the relevant attributes for creating the selection components should find out by the system first.

This procedure is divided into three phases-

- Relevant attribute selection-

In this phase, the relevant or similar type of attributes are selected and then grouped.

- Ranking selection component-

In this phase, components grouped which are selected in the first phase are acquired and these components are then ranked according to their usage.

- Diversity of selection components-

There may be number of redundancies or overlays in two selection components. Therefore, in order to select the recommended components, a high diversity should be provided. Diversity is the major topic in web searches and recommendation system which are proposed in [7] and [8]. Simultaneously, maximizing the diversity and precision is an NP-hard problem [6].

In dynamic query form, it is observed that the most redundant components are constructed by the same attribute. So for each attribute, only the best selection components must be recommended.

VI.FUTURE WORK

In the future work, we can develop the multiple methods for capturing the user's interest for the query besides the click feedback. We can also add a text box for the user to enter the some keywords queries. The relevance score between the query form and the keywords can be incorporated into the ranking of form components in each step.

VII.CONCLUSION

Dynamic query forms helps users to generate the query form dynamically. The main idea is to use a probabilistic model for providing the rank form components based on user preferences. Here we are capturing the user preferences using historical queries and run-time feedback both. Experimental results show that the success rate of dynamic query form is high as compared to static query form. Form component ranking makes easier for the user to customize query form.

REFERENCES

[1]. Korzh.com. (2005)," EasyQuery [Online],"Available: http://devtools.korzh.com/eq/dotnet/
[2]. Adobe. (1995)," Cold Fusion [Online],"Available: http://www.adobe.com/products/coldfusion/
[3]. 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.
[4]. Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, "Dynamic Query Form for Database Queries," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 9, pp. 2167, sept 2014.
[5]. C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in Proc. VLDB, Berlin, Germany, Sept. 2003, pp. 81–92.
[6]. G. Salton and M. McGill, "Introduction to Modern Information Retrieval", New York, NY, USA: McGraw-Hill, 1984.
[7]. R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in Proc. WSDM, Barcelona, Spain, Feb. 2009, pp. 5–14.
[8]. D. Rafiei, K. Bharat, and A. Shukla, "Diversifying web search results," in Proc. WWW, Raleigh, NC, USA, Apr. 2010, pp. 781–790.
[9]. Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, "Dynamic Query Form for Database Queries," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 9, pp. 2168, sept 2014.
[10]. Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, "Dynamic Query Form for Database Queries," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 9, pp. 2169, sept 2014.
[11]. Priyanka P. Nikam, "A Review on Dynamic Query Forms for Database Queries," Priyanka P. Nikam / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6), pp. 8080, 2014.