ISSN: 2321-7782 (Online)
Volume 2, Issue 9, September 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

A survey paper on dynamic query forms for database queries

Revathy P S
P G Scholar
Department of Computer science and Engineering
Mohandas College of Engineering and Technology
Trivandrum – India

Abstract: Scientific databases and web databases maintain large and heterogeneous data. The databases contain over hundreds or even thousands of relations and attributes. Predefined old query forms are not able to satisfy various ad-hoc queries from users on those databases. Dynamic query form, a new database query form interface used to dynamically generate query forms. The use of DQF is to capture a user's preference and rank query form components, assisting user to make decisions. The query form is an iterative process and is guided by the user. The system automatically generates ranking lists of form components and the user then adds the desired form components into the query form. Based on the captured user preference ranking of form components can be done. User can also fill the query form and submit queries to see the query result. A query form could be dynamically refined till the user satisfies with the query results.

Keywords: Query Forms, User Interaction, Query Form Generation.

I. INTRODUCTION

Query form is one of the most widely used user interfaces for querying databases. Old query forms are designed and predefined by developers or database administrator in various information management systems. The development of web information and traditional databases, modern databases become very large and complex. Databases have over hundreds of entities. Many web databases, such as Freebase and DBPedia [7] 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. The database management and development tools, such as EasyQuery , Cold Fusion, SAP and Microsoft Access, provide several mechanisms to let users create customized queries on databases. The creation of customized queries totally depends on users' manual editing. If a user is not familiar with the database schema those hundreds or thousands of data attributes would confuse the user. Dynamic Query Form system a query interface which is capable of dynamically generating query forms for users. The use of DQF is to capture user interests during user interactions. A basic query form which contains very few primaries attributes of the database. The query form is enriched iteratively via the interactions between the user and our system until the user is satisfied with the query results.

II. QUERY FORM TECHNIQUES

The non-expert users make use of the relational database is a challenging topic. Many 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 .The query forms have been utilized in most real-world business or scientific information systems.

a) **Query by Example** (QBE)

It is a database query language for relational databases. It is the first query language used to create visual tables where the user can enter commands. A lot of graphical front-ends for databases use the ideas from query by example today. QBE limited only for the purpose of retrieving data, Query by example was later extended to allow other operations, such as insert, select, deletes, forms, updates, create tables.

The motivation behind QBE is that a parser can convert the user's actions into statements expressed in a database manipulation language, such as SQL. A front-end can minimize the burden on the user to remember the importance of SQL, and it is easier and more productive for end-users (and even programmers) to select tables and columns by selecting them rather than typing in their names. The problem of query by example is relational completeness and ordering problem.

EXAMPLE FORM

.....Name: Bob

..Address:

.....City:

....State: TX

..Zip code:

Resulting SQL: SELECT * FROM Contacts WHERE Name='Bob' AND State='TX'

b) Automated Ranking of Database Query

Automated ranking of the results of a query is a popular aspect of the query model in Information Retrieval (IR) that we have grown to depend on. The database systems support, a selection query on a SQL database returns all tuples that satisfy the conditions in the query. The following two scenarios are not handled by a SQL system:

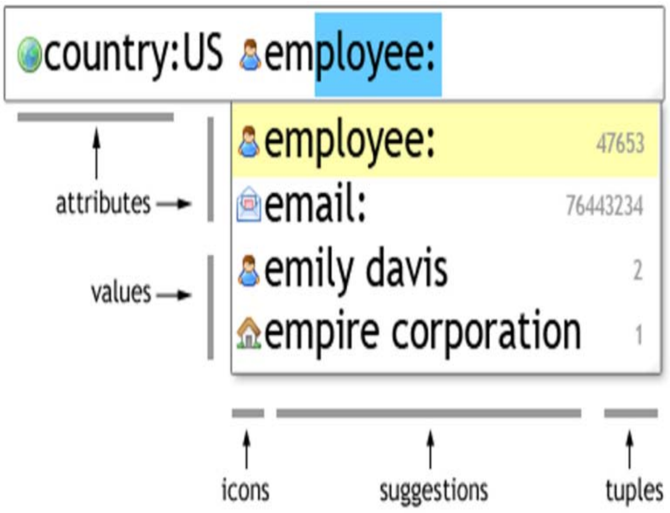
1. *Empty answers*: When the query is selective, the answer may be empty. In that case, it is desirable to have the option of requesting a ranked list of approximately matching tuples without having to specify the ranking function that captures “proximity” to the query. An FBI agent or an analyst involved in data exploration will find such functionality appealing.

2. *Many answers*: When the query is not selective, many tuples may be in the answer. In that case, it will be desirable to have the option of ordering the matches automatically that ranks more “globally important” answer tuples higher and returning only the best matches. A customer browsing a product catalogue will find such functionality attractive. The problem of automated ranking of database query result is the ranking functions might fail to perform this is because many tuples may tie for the same similarity score. It can be also arise for empty answer problem also.

The query forms are generated based on the selected attributes. Then apply clustering algorithm on historical queries to find the representative queries. The forms are then generated based on those representative queries. The problem of the approaches is that, if the database schema is complex and large, user queries could be quite diverse. Even if we generate lots of query forms, there are still user queries that cannot be satisfied by any one of query forms. The problem is that, when we generate a large number of query forms, to let users find an appropriate and desired query form would be challenging. The solution that combines keyword search with query form generation is proposed in. It automatically generates a lot of query forms in advance. The user gives some keywords to find relevant query forms from a large number of regenerated query forms. It works well in the databases which have rich textual information in data schemas. However, it is not appropriate when the user does not have concrete keywords to describe the queries especially for the numeric attributes.

c) Querying using Instant-Response Interfaces

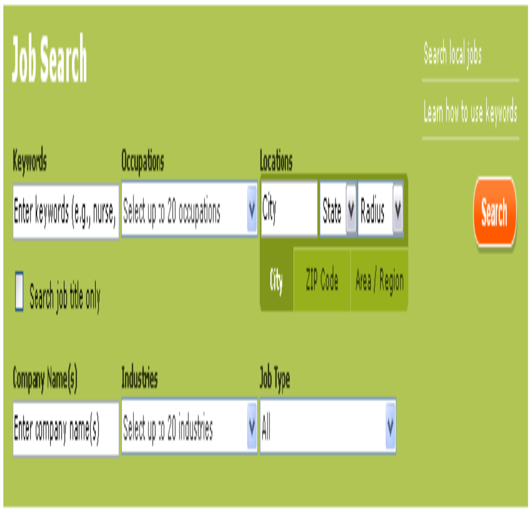
The problem of searching for information in large databases has always been a daunting task. In current database systems, the user has to overcome a multitude of challenges. The major challenge is that of schema complexity: large organizations may have employee records in varied schema, typical to each department. The user may not be aware of the exact values of the selection and provide only a partial or misspelled attribute value. The user to issue queries that are meaningful in terms of result size a query listing all employees in the organization would not be helpful to the user, and could be expensive for the system to compute. Lastly, we do not expect the user to be proficient in any complex database query language to access the database. The problem of assisted querying using instant response interface is the user's information need is explicit .Attempt to apply efficient index structures for basic keyword querying.



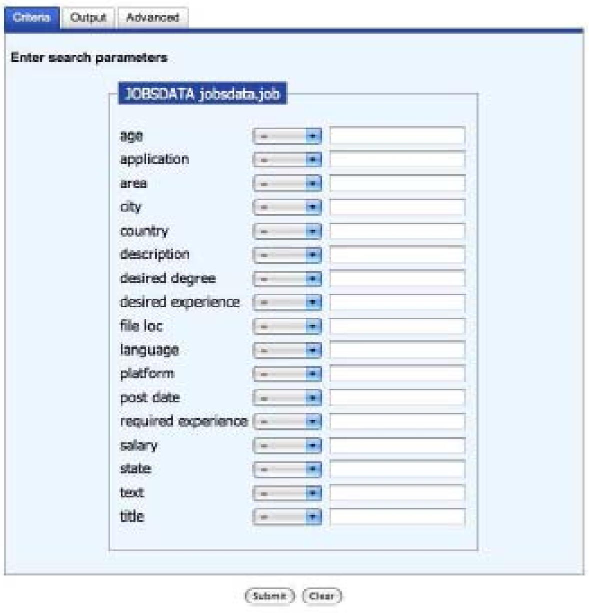
Example: instant interfaces

d) Forms-based Database Query Interface

Forms-based query interfaces are widely used to access databases today. The forms-based interface is often a key step in the deployment of a database. The form is an interface is capable of expressing only a very limited range of queries. The set of forms as a whole must be able to express all possible queries that any user may have. For creating an interface that approaches this ideal is surprisingly hard. To maximize the ability of a forms-based interface to support queries a user may ask when bounding both the number of forms and the complexity of any form. A database schema and content we present an automated technique to generate a good set of forms that meet the above criteria. A careful analysis of real or expected query workloads are useful in designing the interface the query sets are often unavailable or hard to obtain prior to the database even being deployed. The generating a good set of forms just using the database itself is a challenging yet important problem. The problem of automated creation of forms based database querying interfaces is to use history log and use association mining for Related Entities.



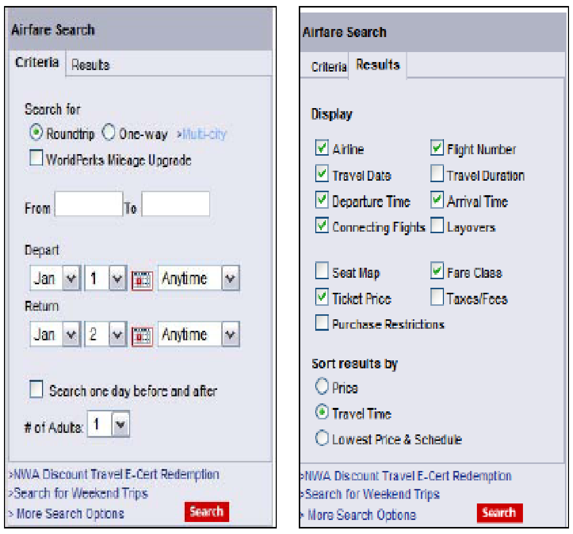
Form based database query form



Form automatically generated

e) Form Customization

A form-based query interface is usually the preferred means to provide an unsophisticated user access to a database. The interface is very easy to use, requiring no training, but it also requires little or no knowledge of how the data is structured in the database. A typical form is static and can express only a very limited set of queries. Query specification is limited by the expertise and vision of the interface developer at the time the form was created.The modifications are themselves specified through filling forms to create an expression in an underlying form manipulation expression language we define. To modify forms is not much greater than form filling. A form editor is used to implements form manipulation language. A query generator that modifies the form's original query based on a user's changes. The tool provides an effective means for specifying complex queries. The problem of form customization is limiting the usefulness of forms: their restrictive nature. Existing database clients and tools make great efforts to help developers design and generate the query forms, such as EasyQuery [8], Cold Fusion [6], SAP, Microsoft Access and so on. The existing databases provide visual interfaces for developers to create or customize query forms. This tool is used by the professional developers who are familiar with their databases, not for end-users a system which allows end-users to customize the existing query form at run time. An end-user may not be familiar with the database. If the database is very large and complex, it is difficult for them to find appropriate database entities and attributes and to create desired query forms.



Example for Customized form

III. COMPARISON

Title	Features	Problems
Query-by example	Provides a simple interface for a user to enter queries.	1) Relational 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, uncomfortable with a formal query language

IV. CONCLUSION

Modern scientific databases and web databases maintain large data. These real-world databases contain over hundreds or even thousands of relations and attributes. Query forms are not able to satisfy various ad-hoc queries from users on those databases. DQF, a novel database query form interface, used to dynamically generate query forms. The 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. Ranking of form components also makes it easier for users to customize query form.

References

1. S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In CIDR, 2003.
2. 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.
3. 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.
4. M. Jayapandian and H. V. Jagadish. Automating the design and construction of query forms. IEEE TKDE, 21(10):1389–1402, 2009.
5. 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.
6. Cold Fusion. http://www.adobe.com/products/coldfusion/.
7. DBPedia. http://DBPedia.org.
8. EasyQuery. http://devtools.korzh.com/eq/dotnet/.
9. Freebase. http://www.freebase.com.