



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5

Issue: V

Month of publication: May 2017

DOI:

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Free Form Query Language for Database Queries

M. Nagaseshudu¹, K. Bala Krishna²

¹Adhoc Lecturer, ²Assistant Professor, Dept. of CSE, JNTUACE, Pulivendula, AP, India

Abstract: Query form is one of the most widely used user interfaces for querying databases. 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. 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 before identifying the final candidates. The essence of FFQL is to capture user interests during user interactions and to adapt the query form iteratively. Each iteration consists of two types of user interactions: Query Form Enrichment and Query Execution. The basic query form is then enriched iteratively via the interactions between the user and our system until the user is satisfied with the query results. In this paper, we mainly study the ranking of query form components and the dynamic generation of query forms.
Keywords: FFQL.

I. INTRODUCTION

Query form is one of the most widely used user interfaces for querying databases. 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. 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 before identifying the final candidates. The essence of DQF is to capture user interests during user interactions and to adapt the query form iteratively. Each iteration consists of two types of user interactions: Query Form Enrichment and Query Execution. The basic query form is then enriched iteratively via the interactions between the user and our system until the user is satisfied with the query results. In this paper, we mainly study the ranking of query form components and the dynamic generation of query forms.

Traditional question forms square measure designed and pre-defined by developers or DBA in varied data management systems. With the fast development of internet data and scientific databases, trendy databases become terribly massive and sophisticated. Therefore, it is tough to

style a collection of static question forms to satisfy varied ad-hoc information queries on those complicated databases. Query forms are designed and pre-defined by developers in data management systems. Difficult to style a collection of static question forms to satisfy varied ad-hoc info queries on complicated databases. We propose a dynamic question type system that generates the question forms per the user's need at run time. The system provides an answer for the question interface in massive and arduous databases. This paper proposes DQF, a completely unique info question type interface that is in a position to dynamically generate question forms. The aspect of DQF is to capture a user's preference and rank question type parts, aiding him/her to form selections. The generation of a question type is associate degree unvaried method and is guided by the user. At every iteration, the system mechanically generates ranking lists of type parts and also the user then adds the specified type parts into the question type.

The ranking of type parts is predicated on the captured user preference. A user may also fill the question type and submit queries to look at the question result at every iteration. During this manner, type can be dynamically refined until the user satisfies with the query results. We suggest a dynamic question type generation approach that helps users dynamically generate question forms. The dynamic approach usually ends up in higher success rate and less complicated question forms compared with a static

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

approach. The ranking of type parts also makes it easier for users to customise question form.

II. LITERATURE SURVEY

A. *A Case for a Collaborative Query Management System*

Nodira Khousainova, Magdalena Balazinska, Wolfgang Gatterbauer, Yong Chul Kwon, and Dan Suciu. New environments are emerging where large numbers of users need to develop and run complex queries over a very large, shared data repository. Examples include large scientific databases and Web-related data. These users are not SQL savvy, yet they need to perform complex analysis on the data and are further constrained by the high cost of running and testing their queries, often on a shared server cluster.

B. *Similarity Measures for Categorical Data: A Comparative Evaluation*

Shyam Boriah Varun Chandola Vipin Kumar Department of Computer Science and Engineering. In this paper we tend to study the performance of a range of similarity measures within the context of a particular data processing task: outlier detection. Results on a range of information sets show that while no one live dominates others for all sorts of issues, some measures square measure ready to have systematically high performance.

C. *USHER: Improving Data Quality with Dynamic Forms*

Kuang Chen #1, Harr Chen _2, Neil Conway. In this paper, we've shown that probabilistic approaches are often accustomed style intelligent knowledge entry forms that promote high knowledge quality. USHER leverages knowledge-driven insights to modify multiple steps within the data entry pipeline. Before entry, we discover associate ordering of kind fields that promotes fast info capture, driven by a greedy info gain principle. throughout entry, we have a tendency to use a similar principle to dynamically adapt the shape supported entered values.

D. *Automated Ranking of Database Query Results*

Sanjay Agrawal Microsoft Research. In this paper, we have presented our experience in attempting to build a generic automated ranking infrastructure for SQL databases. This is consistent with our research philosophy of seeding the relational database management infrastructure with functionality necessary and useful for data exploration.

E. *Snip Suggest: Context Aware Auto completion for SQL*

Nodira Khousainova, YongChul Kwon, Magdalena Balazinska. In this paper, we have a tendency to gift Snip Suggest, a system that has on-the-go, context-aware help within the SQL composition method. Snip Suggest aims to assist the increasing population of non-expert database users, United Nations agency have to be compelled to perform advanced analysis on their large-scale datasets, however have issue writing SQL queries. As auser varieties a question, Snip Suggest recommends doable additions to various clauses within the question exploitation relevant snippets collected from a log of past queries.

F. *Probabilistic Information Retrieval Approach for Ranking of Database Query Results*

Surajit Chaudhuri Microsoft Research. We projected a totally automatic approach for the Many-Answers drawback which leverages information and work statistics and correlations. Our ranking functions area unit based upon the probabilistic IR models, judiciously tailored for structured information. We tend to presented results of preliminary experiments that demonstrate the potency in addition because the quality of our ranking system.

III. EXISTING SYSTEM

Traditional question forms square measure designed and pre-defined by developers or DBA in varied data management systems. With the fast development of internet data and scientific databases, trendy databases become terribly massive and sophisticated. Therefore, it's tough to style a collection of static question forms to satisfy varied ad-hoc information queries on those complicated databases.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A. Limitations

- 1) Query forms are designed and pre-defined by developers in data management systems.
- 2) Difficult to style a collection of static question forms to satisfy varied ad-hoc info queries on complicated databases.

IV. PROPOSED SYSTEM

We propose a dynamic question type system that generates the question forms per the user's need at run time. The system provides an answer for the question interface in massive and arduous databases. This paper proposes FFQL, a completely unique info question type interface that is in a position to dynamically generate question forms. The aspect of FFQL is to capture a user's preference and rank question type parts, aiding him/her to form selections. The generation of a question type is associated degree unvaried method and is guided by the user as shown in Fig.1. At every iteration, the system mechanically generates ranking lists of type parts and also the user then adds the specified type parts into the question type. The ranking of type parts is predicated on the captured user preference. A user may also fill the question type and submit queries to look at the question result at every iteration. During this manner, type can be dynamically refined until the user satisfies with the query results.

A. Advantages

- 1) We suggest a dynamic question type generation approach that helps users dynamically generate question forms.
- 2) The dynamic approach usually ends up in higher success rate and less complicated question forms compared with a static approach
- 3) The ranking of type parts also makes it easier for users to customize question form.

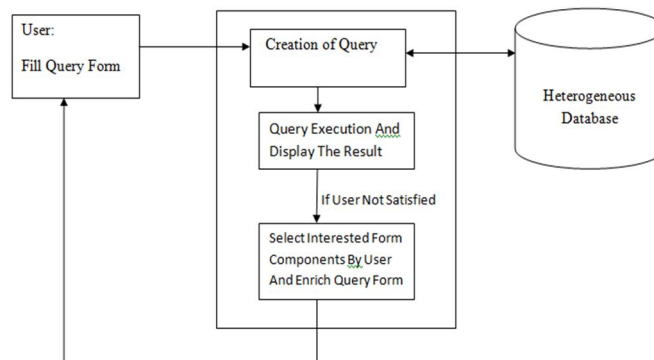


Fig.1. Architectural Diagram.

V. MODULES

A. Server Module

This module allowing this is able to change the employment of SQL for querying databases. Relational information bases (or XML) are mapped simply to the present primitive data model. The first key of a table is seen as a topic, a column label as a predicate, and therefore the information entry in this column as AN object. Foreign keys represent relationships between information components across tables. Mapping from computer database and XML. In our project, we tend to victimization MySQL because the information server and Apache house cat 5.0 because the internet server. These two are the most core of the server facet programming. Our application has been deployed within the Apache house cat so all reasonably HTTP Requests and response is handled simply. All the parameters passed from the request is retrieved victimization the `getParameter()` technique of Http Request Object.

B. Connection Module

SQL is not merely a single-purpose interface,. The Connection between the Client and the Web Server is maintained by the object Http Connection.. Using this connection module we could able to retrieve the database information by passing the Http Request. The requesting attributes is sent as the parameters of the url built for Http Connection.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

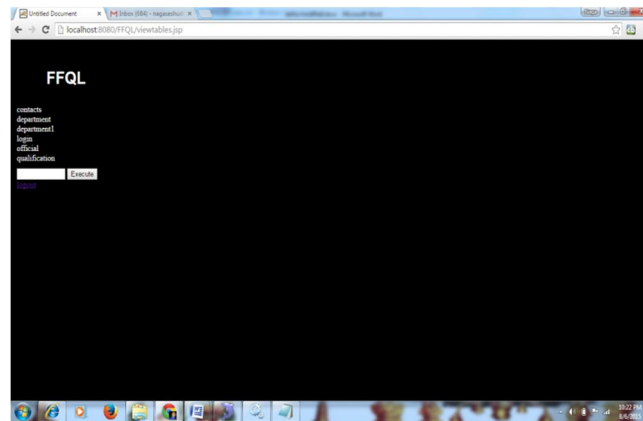
C. Design of the Application

People are not required to know the schema in advance. According to the request sent by the client, the server process the data and it is responded to the client, which is further received by the help of open Data Input Stream in Http Connection Object. So the client application is designed according to the Field information of the tables retrieved from the server. We would be using Choice Group Object for designing Radio Buttons, Check Boxes and Dropdown list boxes.

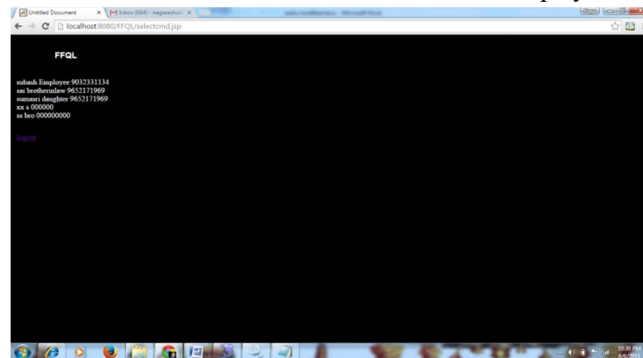
D. Query Generation Module

MashQL is meant to ease developers' work by generating mobile applications that need accessing a foreign info. It provides a awfully friendly computer program and builds each, the client and server sides. For the client-side it generates Java American state applications. The remote databases are managed exploitation JSP and My SQL. Once info is retrieved, knowledge is regenerate to an acceptable format so as to store it persistently on the mobile device exploitation the Record Management System (RMS). the look of the appliance is completed by the information sorts that we have a tendency to employed in the info. the selection that we have a tendency on the point of use, according the question are generated behind the screen. once the entire operation of choosing the alternatives we have a tendency to purported to execute the question by passing it because the parameter to the server through protocol Request.

1) Results



In this application in the drop down created the search box for which of the table would be inserted, deleted and updated the data in web application. Suppose shown the contacts table data show type on the search box then click execute button goes to viewfield.jsp that displays the table information. So, click on select all button that displays total information of contacts table.



After that insert the data on the particular table

VI. CONCLUSION AND FUTURE WORK

It is develop a database query formulation system for smart phones accept unplanned queries, by allowing freeform inputs. As cell phones are poor in terms of resources as compared to other mobile devices, the successfulness of implementing such a method is applicable to the other devices. Indefinite query method in the form of free-form language can provide a much simpler interface for users to formulate queries. The method can also helpful in reducing the number of query inputs, particularly in cases where joins of

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

relations are needed. This application has search text box top of the screen and provides search query results.

REFERENCES

- [1] ColdFusion. [http://www.adobe.com/products/cold fusion/](http://www.adobe.com/products/cold-fusion/).
- [2] DBPedia. <http://DBPedia.org>.
- [3] Easy Query. <http://devtools.korzh.com/eq/dotnet/>.
- [4] Freebase. <http://www.freebase.com>.
- [5] C.C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In Proceedings of VLDB, pages 81–92, Berlin, Germany, September 2003.
- [6] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In Proceedings of WSDM, pages 5–14, Barcelona, Spain, February 2009.
- [7] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In CIDR, 2003.
- [8] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In Proceedings of SIAM International Conference on Data Mining (SDM 2008), pages 243–254, Atlanta, Georgia, USA, April 2008.
- [9] 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.
- [10] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. Probabilistic information retrieval approach for ranking of database query results. ACM Trans. Database Syst. (TODS), 31(3):1134–1168, 2006.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)