PRIVACY PROTECTED DYNAMIC QUERY FORM USING ROB FRUGAL ENCRYPTION

***Revathy P S, ** Ramya J S**
**PG Scholar, Department of Computer Science,*
Mohandas College of Engineering and Technology, Anad
***Assistant Professor, Department of IT,*
Mohandas College of Engineering and Technology, Anad

ABSTRACT

Modern scientific databases and web databases maintains large number of data. These real world databases contain over hundreds of relations and attributes. Traditional predefined queries are not able to satisfy various adhoc queries from the users on those databases. With the rapid development of web information and scientific database modern database become very large and complex. For avoiding that a new approach is used called Dynamic Query Form: a query form system which is capable of dynamically generating query forms for the user. The use of DQF is to capture user interests during user interactions and to generate the query form dynamically. 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. The form could be dynamically refined till the user satisfies with the query results. ***.*** *Also an encryption technique is used called rob frugal encryption scheme. In this encryption method fake transactions are made. Also we try to add skyline queries. A skyline query is one of the most useful but least obvious data programming patterns. The experimental results show that this dynamic query form functioned in the best possible manner with the least waste of time and efforts.*

Index Terms— Query Forms, Scientific databases, Query results

1. 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 Easy Query, 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. 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 primary 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.

2. RELATED WORK

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.

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.

d) Form Customization

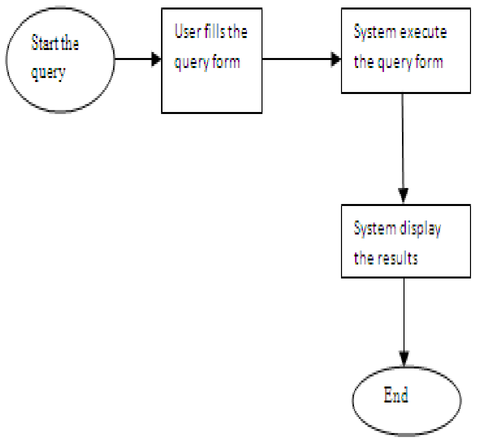
A form-based query interface is usually the preferred means to provide an unsophisticated user access to a database. The interface is very easy to use, requiring no training, but it also requires little or no knowledge of how the data is structured in the database. A typical form is static and can express 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 Easy Query [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.

3. PROBLEMS OF PREVIOUS PAPERS

- Query form doesn't deal with unstructured data stored in text files.
- The existing system are less secure they are susceptible to web bots.
- Forms can be used for finding the percentage of data accessed but cannot find the agent who accessed the data.
- More time consuming process.
- Not applicable for group based and multilevel data.
- In the existing system the user has to fill the query form and submit the query then the system will execute the query and shows the results

4. SYSTEM OVERVIEW

. With the rapid development of web information and web databases, modern database become more complex and large. Therefore it is very difficult to design a set of static query forms for the users.Modern scientific databases and web databases maintain large number of data. These real world databases contain over hundreds of relations and attributes. Traditional predefined queries are not able to satisfy various adhoc queries from the users on those databases. With the rapid development of web information and scientific database modern database become very large and complex. For avoiding that a new approach is used called Dynamic Query Form: a query form system which is capable of dynamically generating query forms for the user. The use of DQF is to capture user interests during user interactions and to generate the query form dynamically. 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. The form could be dynamically refined till the user satisfies with the query results**.** Also an encryption technique is used called rob frugal encryption scheme. In this encryption method fake transactions are made.


Start the query
User fills the query form
System execute the query form
System display the results
End

First the user does the transactions and encrypts those transactions in the Server side. When required the user will decrypt that transactions in the server side. Also we try to add skyline queries in my paper. A skyline query is one of the most useful but least obvious data programming patterns. When we have a set of data and a query with two constraints that are conflicting, the output of this is the "Skyline" Query. The experimental results show that this dynamic query form functioned in the best possible manner with the least waste of time and efforts.

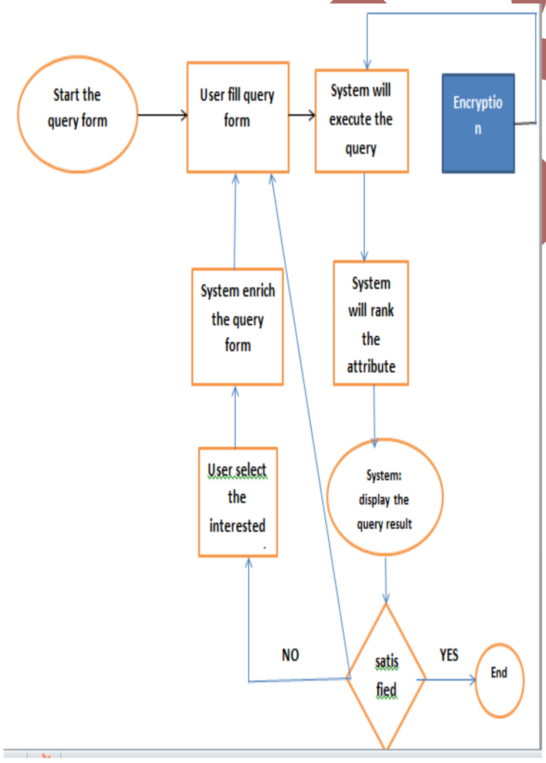

Start the query form
User fill query form
System will execute the query
Encryption
System enrich the query form
System will rank the attribute
User select the interested
System: display the query result
NO
satis fied
YES
End

Fig 1: DQF Architecture

DQF starts with a basic query form which contains very few primary attributes of the database. The query form is then enriched iteratively via the interactions between the user and our system until the user is satisfied with the query results. Dynamic query form system which generates the query forms according to the user's desire at run time. The dynamic query form provides a solution for the query interface in large and complex databases. Apply F-measure to estimate the goodness of a query form. F-measure is a typical metric to evaluate query results. This metric is also appropriate for query forms because query forms are designed to help users query the database. The query form goodness is determined by the query results generated from the query form. By using this, we rank and recommend the potential query form components so that users can refine the query form easily. The properties of dynamic query forms are

1. Graphical visualization of database and search results.
2. Deliver the search results immediately.
3. Faster entrance for the beginners.

Query Encryption

The encryption technique used here is called rob frugal encryption scheme. In this encryption method fake transactions are made. First the user does the transactions and encrypts those transactions in the server side. When required the user will decrypt that transactions in the server side. In order to extent the system performance security and make the system suitable for private database applications an encryption is added to it. The encryption is called Rob Frugal Encryption scheme.

Steps in Rob Frugal Encryption

1: One to one substitution: it helps to encrypt the plain text into cipher text using a private key.

2: K grouping method: grouping the cipher items into K adjacent item in decreasing order of support.

3: Constructing Fake Transactions: Given a noise table specifying the noise needed for each cipher item we generate the fake transactions as follows. First, we drop the rows with zero noise, corresponding to the most frequent items of each group or to other items with support equal to the maximum support of a group. Second, we sort the remaining rows in descending order of noise.

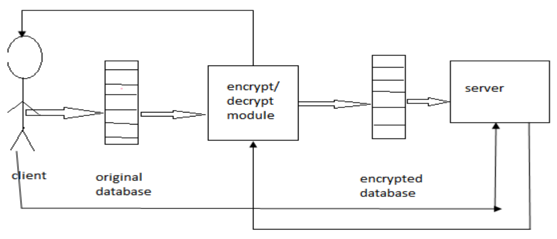


Fig 2: Encryption

5. SYSTEM TESTING

The dynamic query forms is implemented as a web based system using JDK with Java Server Page. The dynamic web interface for the query forms used open-source JavaScript library JQuery . We used MySQL as the database engine.

Data sets: 2databases: NBA, Car were used in our experiments.

While comparing the order by query and the skyline query, the error rate of order by is more than that of the skyline query. The error rate gradually increases when the number of query result increases.

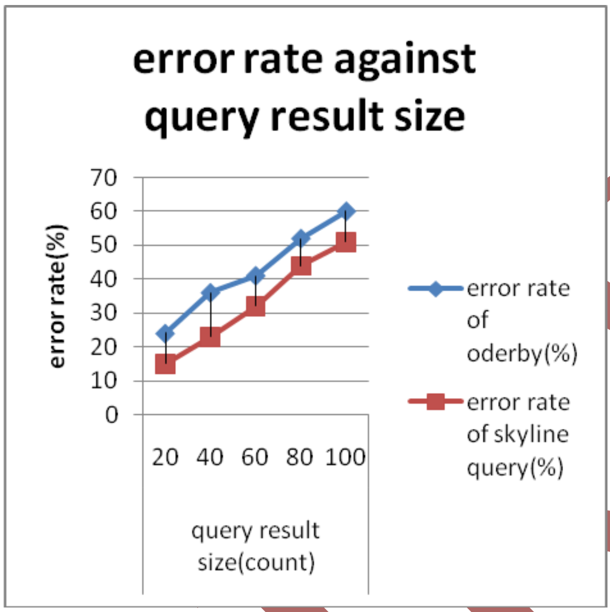


Fig 3: Error rate against query result size

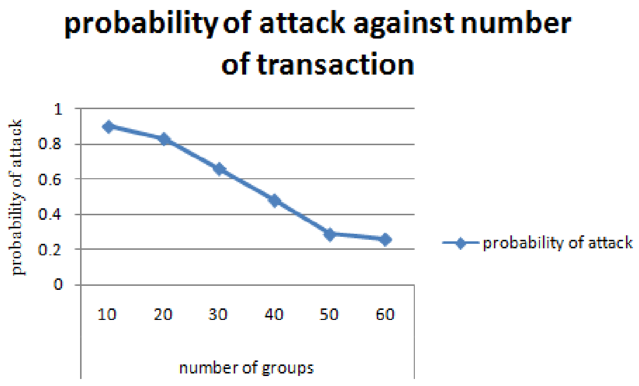


Fig 4: Attacking probability against transaction sets.

In fig 3 shows the output of the error rate against query result size. In this figure when comparing the order by query and skyline query, the error rate of skyline query is less than that of the order by query. But in fig 4, if the transaction set groups number increases the probability of attack decreases. Also the system performance security is improved with the help of the Rob Frugal Encryption.

6. CONCLUSION AND FUTURE WORK

Traditional method doesn't deals with user feedback. DQF capture user's preferences and rank query form components.DQF which is able to dynamically generate query forms. Also in DQF privacy is preserved using rob frugal encryption. With the rapid development of web information and scientific database modern database become very large and complex. For avoiding that a new approach is used called Dynamic Query Form: a query form system which is capable of dynamically generating query forms for the user. The use of DQF is to capture user interests during user interactions and to generate the query form dynamically. 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. The form could be dynamically refined till the user satisfies with the query results**.** Also an encryption technique is used called rob frugal encryption scheme. In this encryption method fake transactions are made. First the user does the transactions and encrypts those transactions in the server side. When required the user will decrypt that transactions in the server side. Also we try to add skyline queries in my paper. A skyline query is one of the most useful but least obvious data programming patterns. When we have a set of data and a query with two constraints that are conflicting, the output of this is the "Skyline" Query. In future work the Dynamic Query Form can be extended to non relational data. Also we can include text box for developers to manually type the SQL.

7. REFERENCES

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

[2] DBPedia. http://DBPedia.org.

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

[4] Freebase. http://www.freebase.com.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton. Combining keyword search and forms for ad hoc querying of databases. In *Proceedings of ACM SIGMOD Conference*, pages 349–360, Providence, Rhode Island, USA, June 2009.

[11] S. Cohen-Boulakia, O. Biton, S. Davidson, and C. Froidevaux. Bioguidesrs: querying multiple sources with a user-centric perspective. *Bioinformatics*, 23(10):1301–1303, 2007.

[12] G. Das and H. Mannila. Context-based similarity measures for categorical databases. In *Proceedings of PKDD 2000*, pages 201–210, Lyon, France, September 2000.

[13] W. B. Frakes and R. A. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, 1992.