

# Runtime Database Query Recommender For Dynamic Query Forms

Shamika S. Mule<sup>1</sup>, Rajendra D. Gawali<sup>2</sup>

P.G. Student, Department of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai,  
Maharashtra, India<sup>1</sup>

Assistant Professor, Department of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai,  
Maharashtra, India<sup>2</sup>

**ABSTRACT:** At present there has been massive amount of growth in the number of users who needs to access online databases. Many of them do not have detailed knowledge of query languages. But to query structured database user either must have knowledge of database schema or there must be a system which is capable of generating query forms dynamically according to user's desire. So this paper proposes DQF dynamic query form, a novel query form interface which can dynamically generate query forms for users. Implication of DQF is to capture user's preference and rank query form components. User can refine his query form until he gets satisfied with the query results. Also this paper proposes NLP, natural language processing module for users who are comfortable with natural language query.

**KEYWORDS:** Query Form, Query Form interface, Database schema, Natural Language Processing.

## I. INTRODUCTION

Search engines are most widely used to access data from unstructured databases but to retrieve data from relational databases structured query language is required. To execute suitable queries user needs to know database schema. Common public is successful using keyword search and follow the hyperlinks to navigate from one document to another document. But keyword search techniques alone cannot be used on data stored in relational databases since information needed to answer a keyword query is often split across the tables due to normalization.

To design a set of a static query forms which can satisfy various ad-hoc database queries is a difficult task because of thousands of structured web entities. If a user is unaware with database schemas then these hundreds or thousands of attributes might puzzle him/her [1]. Our mission in this paper is to explore techniques which assist users who are reluctant to use SQL in creating ad-hoc queries over relational databases.

## II. DATABASE INTERFACES

There are many database interfaces for querying database. Two most widely used interfaces are Query By Example and Query Form.

1. **Query By Example (QBE) :** QBE is also used for querying relational data. It has graphical user interface. Using GUI users can create example tables. Thus user needs minimum information to get started. This language contains relatively very few concepts. QBE is usually suited for simple queries and it can be expressed in terms of less number of tables. QBE was developed by IBM. Some systems, such as Microsoft Access, offer partial support for form-based queries. QBE-like interface is offered in addition to SQL. QBE deals with simpler queries and SQL deals with complex queries.
2. **Query Form :** Query forms offer freedom to users such that they may or not may be familiar with the database query language. Still they can make use of query forms. Each form is an interface for an incomplete SQL query in which some parameter values are not known until user provides them at the time of filling the forms [4]. For example consider a relation Employee having attributes, (Ecode, Ename, Dept, Design, Branch)  
**SELECT \* FROM Employee WHERE Ecode op value AND Ename op value AND Dept op value AND Design op value AND Branch op value**

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

where op and value are parameters representing an operator and a constant respectively. Then Query executed will be shown as follows,

```
SELECT * FROM Employee WHERE Dept = 'Computer'
```

In other words, a template with user-specified parameters corresponds to a SQL query. Predicates for which users have not specified parameters, such as those for Ename, Ecode, Design, and Branch in this example are excluded from query evaluation.

### III. FORM GENERATION

M.Jayapandian and H.V.Jagdish in their “Automated creation of a forms based database query interface and in Automating the design and construction of query forms” stated that there are two approaches to generate the database query form without user participation. One is Data driven method and other is workload driven method. Data driven method first finds a set of data attributes which are most likely queried based on the database schema and data instances and then query forms are generated based on selected attributes. Workload driven method applies clustering algorithm on historical queries to find the representative queries and then query forms are generated based on representative queries [2][3]. Problem with these approaches are if database schema is large and complex user queries could be quite miscellaneous. In such a circumstances even if we generate lots of query forms in advance there are still user queries which cannot be satisfied by any one of query forms. Another challenge is how to let users to find an appropriate and desired query form. A solution that combines keyword search with query form generation is proposed in Combining keyword search and forms for ad hoc querying of databases by [4].

Form Generation Approaches are:

**a) Static Query Form (SQF) :**

This is also known as Traditional Query form. SQF can be used at many places where data is hidden behind forms. User just need to fill mandatory fields of the query forms for example reserving tours or purchasing of books can be done using SQF where structure of query forms is fixed. User cannot make any changes to forms while filling those forms.

**b) Customized Query Form (CQF) :**

Only professional developers who are familiar with their databases can make use of customized query forms. They are not for end users.

**c) Dynamic Query Form (DQF) :**

Form should be simple to understand and at the same time must provide broadest possible querying capability to user. Dynamic query form system allows end users to customize the existing query form at run time even though they may not be familiar with database schema.

### IV. FRAMEWORK OF DYNAMIC QUERY FORM SYSTEM

DQF is a user friendly query form interface, which is able to dynamically generate query forms. DQF captures user's preference and rank query form components, assisting him/her to make decisions. At each iteration the system automatically generates ranking lists of form components and the user then adds the desired form components into the query form until gets satisfied results [1].

Additionally if user is more comfortable using natural language English for querying relational database then he / she can make use of NLP query. If user is not satisfied with the query results he /she can rewrite the NLP query until satisfied with the results.

As shown in Figure 1 users can either start with filling basic query form which contains primary attributes of the database or with writing NLP query. System will execute queries and display results. Basic query form is enriched iteratively via interactions between user and the system until user is satisfied with the query results. NLP users also can rewrite the query or keywords to get satisfied results.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

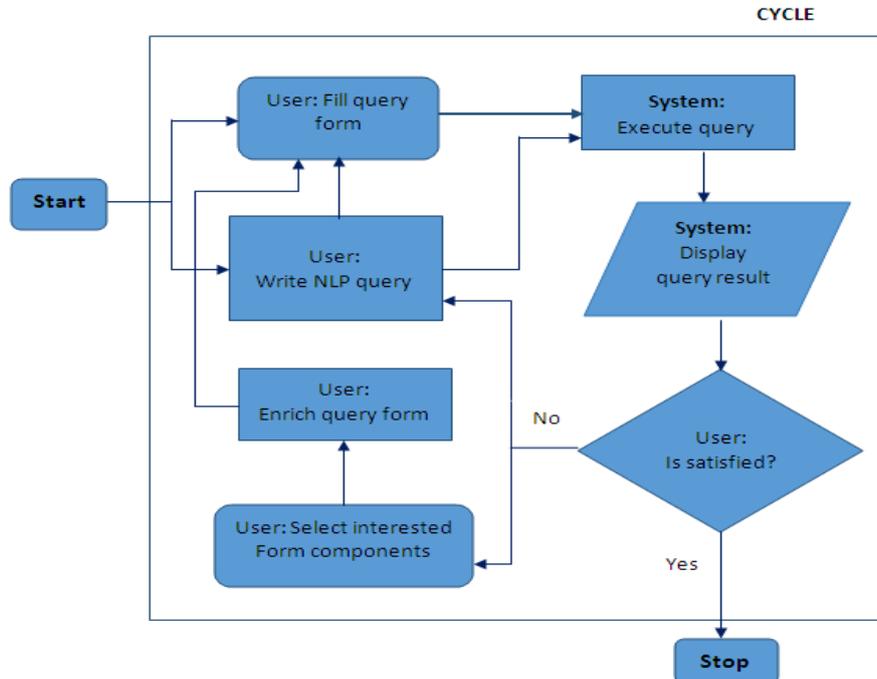


Figure 1 System Architecture of Dynamic Query Form system

## A. Modules of DQF System :

System has following modules along with functional requirements.

### 1) Query Form Enhancement :

- DQF system offers a ranked list of query form components for the user.
- The user has to select the preferred form components to enhance the current query form.

### 2) Query Execution :

- The user has to fill out the current query form and submits the query form.
- System executes the query and displays the results.

### 3) Database Query Recommendation :

- Most frequently executed queries are of high interest.
- The query form components which can capture these high interested queries are ranked higher.

### 4) NLP (Natural Language Processing) :

- User and Admin are two login accounts available.
- User can write or rewrite queries or keywords until gets satisfied results.
- Queries which do not get satisfied results are handed to Admin for further processing.

## B. Query Form Interface :

### i) Query Forms:

Each query form corresponds to an SQL query template. In the user interface of query form  $F$ ,  $A_F$  is the set of columns in the result table.  $\sigma_F$  is the set of input components for users to fill. Query forms allow users to fill parameters to generate different queries [1].

### ii) Query Results:

Database Queries output large amount of data instances and user does not have time to go through all these data instances then to decide whether the query form is desired or not compressed table is used which show high level view of result table. Each instance represents cluster of actual data instances[1].

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

### iii) Ranking Metric :

Precision and Recall are the two traditional measures used to estimate the quality of the query results. Since different inputs can be given to the same query form, each time user may get different results. Hence expected precision which is expected proportion of query results in which current user is interested and expected recall which is expected proportion of user interested data instances returned by current query form are used to estimate expected performance of the query form. User interest is evaluated based on the user's click through on query results and historical queries. The form components which are responsible to capture user interested data instances are ranked higher than other components.

Let F be a query form with selection condition  $\sigma_F$  and projection attribute set  $A_F$ . Let D be the collection of instances in  $\langle \triangleright (R_F) \rangle$ . N is the number of data instances in D. Let d be the instance in D with set of attributes  $A = \{A_1, A_2, \dots, A_n\}$  where  $n = |A|$ . We use  $D_{AF}$  to denote the projection of instance d on attribute set  $A_F$  and we call it a projected instance,  $P(d)$  is the occurrence probability of d in D.  $P(\sigma_F|d)$  is the probability of d satisfies  $\sigma_F$ .  $P(\sigma_F|d) \in \{0,1\}$ ,  $P(\sigma_F|d) = 1$  if d is returned by F and  $P(\sigma_F|d) = 0$  otherwise.  $\alpha$  is the fraction of instances desired by user. Then expected precision and expected recall is as shown below respectively.

$$\text{Precision}_E(F) = \frac{\sum_{d \in D_{AF}} P_u(d_{AF}) P(d_{AF}) P(\sigma_F|d) N}{\sum_{d \in D_{AF}} P(d_{AF}) P(\sigma_F|d) N}$$

$$\text{Recall}_E(F) = \frac{\sum_{d \in D_{AF}} P_u(d_{AF}) P(d_{AF}) P(\sigma_F|d) N}{\alpha N}$$

**Query construction algorithm** constructs dynamic query executed by user at run time. It uses a function which generates database query based on set of projection attributes with selection expression. To find the best selection component for next query form first data instances needs to be retrieved by querying the database. Also this algorithm does not send each query condition to the database engine to select data instances since it will create unnecessarily heavy burden for database engine due to large number of query conditions. Hence it retrieves the set of data instances and checks every data instance with query condition by its own. For this reason the algorithm needs to know the values of all selection attributes. Hence it adds all selection attributes into the projection of query. In this way union of all query conditions are executed.

Another algorithm called **FindBestLessEq** condition is used to find best query condition. This algorithm first sorts values of specific attribute. Then it goes through every data instance and calculates corresponding Fscore.

$$\text{Fscore}_E(F_{i+1}) = \frac{(1 + \beta^2) \text{Precision}_E(F_{i+1}) \cdot \text{Recall}_E(F_{i+1})}{\beta^2 \cdot \text{Precision}_E(F_{i+1}) + \text{Recall}_E(F_{i+1})}$$

Where  $\text{Fscore}_E(F_{i+1})$  is the estimated goodness of the next query form  $F_{i+1}$  and  $\beta$  is the constant parameter to control the preference on expected precision and recall. Our aim is to maximize the goodness of next query form. So the form components are ranked in descending order of  $\text{Fscore}_E[1]$ .

### iv) Natural Language Processing :

In the context of the presented application language can be considered as a communication channel which assists users who have no knowledge of structured language still wants to access relational database. Users can express their database queries in English language using NLP module. User can login through user role and may have access to the database. Admin is another login present for administrator. Those queries which do not get satisfied results are stored to a log. Log is managed and controlled by admin by uploading new training set which contains solutions for unsolved queries.

As Shown in Figure 2 preprocessing phase analyzes the input database query given by user. It involves segmentation of sentence, tokenization, stop word removal and stemming. Additionally spell checker can be used in case of any spelling mistakes in queries. It automatically provides replacements for misspelled words [6].

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

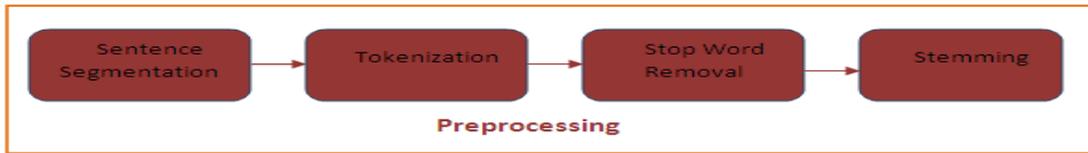


Figure 2 Preprocessing phase

**Segmentation** is the process of decomposing the given database query into its constituent sentences along with its word count. In English, sentence is segmented by identifying the boundary of sentence which ends with full stop (.).Applicable only when query is long more than one sentence.

**Tokenization** is the process of splitting the sentences into words by identifying the spaces ,comma and special symbols between word.

**Stop words** are common words that carry less important meaning than keywords. Such a common words with no semantics and which do not consists of relevant information to the task are eliminated.

**Stemming** reduces diversity of a word to a canonical, morphological representation. Porter stemmer is one of the popular stemmer. It removes prefixes and suffixes to get affix free word [6].

### C. System Implementation :

Dynamic Query Forms are implemented as a web based system using Dot Net 4.0 , C#, jQuery. MSSQL Server is used as Database Engine. Sample database like Baseball is used which was created by Sean Lahman which provides baseball statistics freely available to general public [7].

Sr. No	Name Of Relation	#Attribute	#instances
1	Master	33	18,125
2	Batting	24	96,600
3	Pitching	30	41,857
4	Fielding	18	1,64,899
5	AwardsPlayers	06	5,919
6	Salaries	05	23,141

Table 1: Database Description

Table 1 contains various relations along with their number of attributes and instances. It contains various relations like Master, Batting, Fielding, Pitching, Salaries and AwardsPlayers. These relations contains information of batting, fielding, pitching statistics as well as player’s salaries and awards details.

playerID	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear
permele01	11	25	USA	OH	Cleveland	NULL
pernohu01	3	14	USA	OR	Applegate	1944
perraro01	4	1	USA	NJ	Paterson	NULL
perribi01	6	23	USA	LA	New Orleans	1974
perrijo01	2	4	USA	MI	EsCANABA	1969
perrini01	1	14	USA	WI	Clinton	1948
perrige01	8	13	USA	WI	Sharon	1960
perripo01	8	30	USA	LA	Arcadia	1947
perrybo02	9	14	USA	NC	New Bern	NULL
perrybo01	3	21	USA	NC	Snow Camp	1990
perrych01	9	13	USA	FL	Live Oak	NULL
perrycl01	12	18	USA	WI	Clayton	1954
perryga01	9	15	USA	NC	Williamston	NULL
perryge01	10	30	USA	GA	Savannah	NULL
perryha01	7	28	USA	MI	Howell	1956
perryhe01	9	15	USA	FL	Live Oak	NULL
perryjj01	10	30	USA	NC	Williamston	NULL

Figure. 3 Master Relation from BaseBall Dataset

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

Master relation from BaseBall Dataset is shown in Figure3 Master relation comprises of player`s data like their playerid, birthdate, birthcity, nick name, debut game, final game etc. All relations are connected to Master relation.

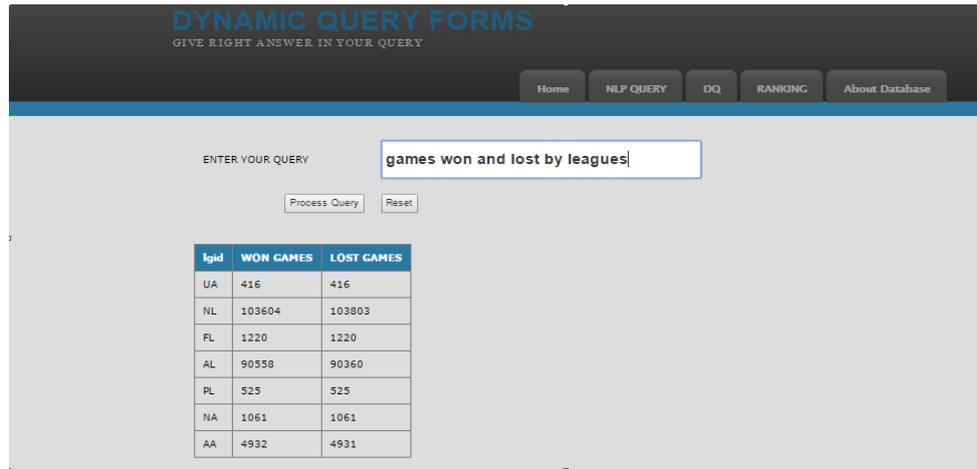


Figure 4 screenshot of Query result using NLP module

As shown in Figure 4 user can get query result for their written database query using NLP module. User can also specify only keywords to access database. Where lgid stands for league identification number. If user is not satisfied with the result, he can rewrite the query until gets satisfied with the result.

A log is maintained in case of any unanswered queries as shown in Figure 5. Only Administrator can have access to log. By examining the log admin will form new queries so that next user can get results for these unsolved queries.

id	query
2	year=2008
3	player=2008
4	year=2008
5	max batting
6	year =2008
7	max played
8	home chenbr01
9	home chenbr01
10	birth date
11	what is the birthday=2008

Figure 5 Unsolved queries Log

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

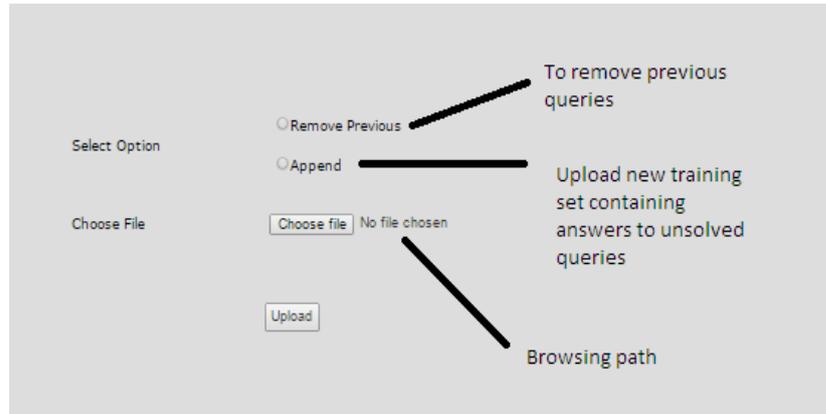


Figure 6 Training Set Upload

Whenever administrator login his account, a log is handed over to him. By forming queries, he will upload answers in a text document for unsolved queries and save this text document. By browsing as shown in Figure 6. new training set can be uploaded so that user can get expected results for these unsolved queries.

Figure 7 shows system prototype. User needs to fill the query form. User also can refine the query form until gets satisfied query results. In addition to that user can make use of recommendations which are present in previous queries. Most frequently accessed queries are present on the top of previous queries. In this way DQF system assists users to communicate with relational database efficiently.

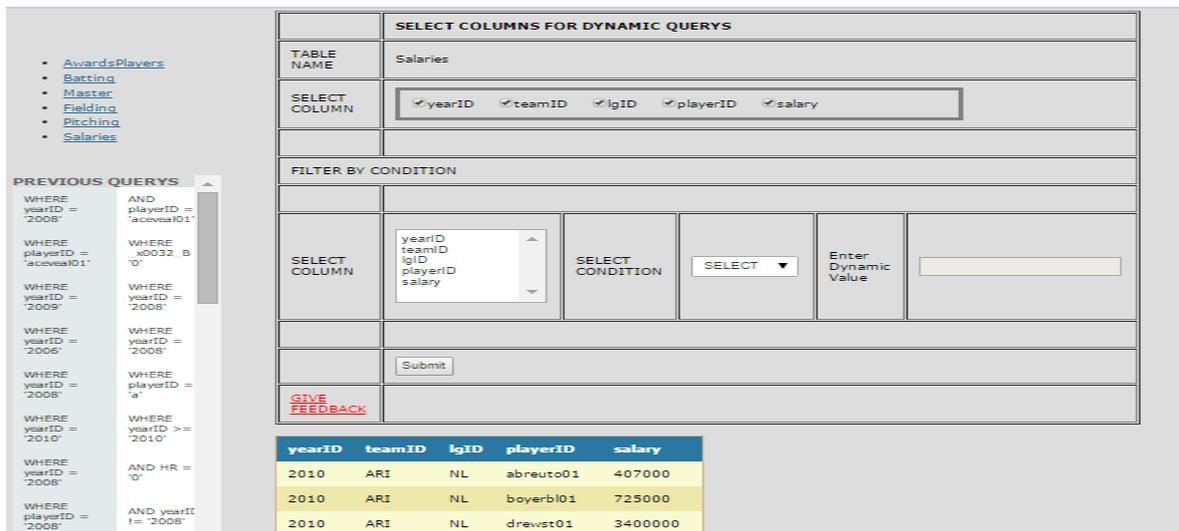


Figure. 7 Screenshot of Web Based Dynamic Query Form

## V. CONCLUSION

In this paper we propose a dynamic query form system which provides user friendly approach to dynamically generate query forms according to user's desire at run time. Probabilistic model is used to rank form components which recommends form components which are of user's interest liked by other peoples in the past. A typical metric Fscore is used to estimate query result. Based on this system can rank and recommend potential query form components. Here

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

efficiency is important since users often expect quick response. Also user can write NLP query if more comfortable in using natural language. Practical results show that the dynamic approach is more powerful and leads to higher success rate than static approach. As a part of our planned future work we intend to extend this approach to non relational data.

## REFERENCES

- [1] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, "Dynamic Query Forms for Database Queries", IEEE Transactions On Knowledge And Data Engineering Vol:Pp No:99 Year April 2013.
- [2] M. Jayapandian and H. V. Jagadish, "Automating the design and construction of query forms", IEEE TKDE, 21(10):1389– 1402, 2009.
- [3] M. Jayapandian and H. V. Jagadish , " Automated creation of form based query interface ",In proceedings of VLDB Endowment,pages 695-709,August 2008.
- [4] Eric Chu Xiaoyong chai , " 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.
- [5] Gonzalez Azadeh Nilfarjam,Emadzadeh and Graciela, "A Hybrid System for Emotion Extraction from Suicide Notes",Department of Biomedical Informatics,Arizona state University ,Arizona, USA.
- [6] Chetana Thaokar, Dr. Latesh Malik,"Test Model for Summarizing Hindi Text using Extraction Method " ,In Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).
- [7] <http://www.baseball1.com> or <http://www.seanlahman.com>

## BIOGRAPHY

	<p>Shamika S. Mule is a PG student in Dept. of Computer Engineering at Lokmanya Tilak College of Engineering, Navi Mumbai, India. She is having academic experience of 10 years at UG level courses of Mumbai University. Her area of interest are database management system and data mining.</p>
	<p>Rajendra D. Gawali is working as assistant professor in Dept. of Computer Engineering at Lokmanya Tilak College of Engineering, Navi Mumbai, India. He is having academic experience of 21 years at UG and PG level courses of University of Mumbai. He has guided many projects at UG and PG level. His areas of interest are Data base management, Data Warehousing and Data mining, Information Retrieval.</p>