



Web Based Dynamic Table Creation and Implementing Query Forms for End User

Ramakrishnan.R¹, Manju.R², Anitha.S³

Associate Professor, Dept. of MCA, Sri Manakula Vinayagar Engineering College, Pondicherry, India¹

MCA Student, Sri Manakula Vinayagar Engineering College, Pondicherry, India²

MCA Student, Sri Manakula Vinayagar Engineering College, Pondicherry, India³

ABSTRACT: This paper proposes the dynamic database creation and query form based on user requirements. In dynamic database, the tables are created dynamically in an iterative process. In the real world database there are heterogeneous data, and the storage and retrieving of those data are complex process. The traditional database is static and they don't satisfy the user needs every time, and this also require huge memory and retrieving time. The static database usually contains fixed number of fields and these predefined fields may not be according to the user requirement. In dynamic database the name of fields and the type of fields are specified by the user in iterative process, at each iteration the fields are raked and these ranked fields are automatically added to the query forms, based on user preferences. The user fills out the query forms and submits the query; the DQF executes the query and provides the results.

KEYWORDS: DQF (Dynamic Query Forms), CQF (Customized Query Forms), SQF (Static Query Forms).

I. INTRODUCTION

Query forms are used for the user interaction with the database. The traditional query forms are static forms. This predefined query forms are designed by the developer or database administrators. Mostly the static forms do not satisfy the user requirements. Most of the static forms cannot use heterogeneous data. The loading time of the static form will be very high and huge database are required. Adding and removing of fields in static query forms and database are complex task. In static forms only simple queries will be available; these simple queries may not satisfy the user need in complex or huge database.

1. User Authentication: In the user authentication module we mainly focus on providing the security to the user profile through the proper user name and password. Here registration is the first step where some details about the user are entered and then that information is stored in the database.

2. Dynamic table creation: In dynamic table creation module, the tables are created dynamically in an iterative process. At each iteration the name of the fields and type of the fields (integer, float, varchar etc.) are given by the user. During this process the fields are ranked automatically and added to the query forms based on the user preferences.

3. Query Execution: In query execution the user fills out the query forms and submits the query and then dynamically generated query form executes the query and provides the results.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

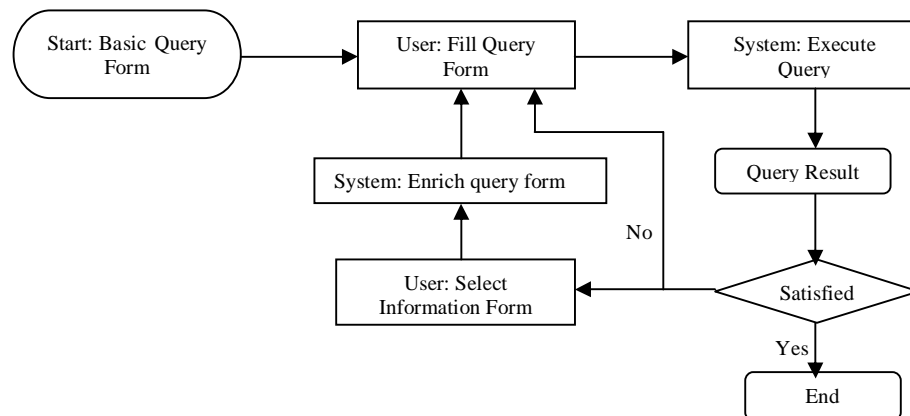


Fig.1. Flowchart of Dynamic Query Form

II. RELATED WORK

EXISTING SYSTEM

Recently proposed automatic approaches to generate the database query forms without user participation presented a data-driven method. It first finds a set of data attributes, which are most likely queried based on the database schema and data instances. Then the query forms are generated based on the selected attributes.

One problem of the aforementioned approaches is that, if the database schema is large and complex, user queries could be quite diverse. In that case, even if we generate lots of query forms in advance, there are still user queries that cannot be satisfied by any one of query forms. Another problem is that, when we generate a large number of query forms, how to let users find an appropriate and desired query form would be challenging. A solution that combines keyword search with query form generation is proposed. It automatically generates a lot of query forms in advance.

The user inputs several keywords to find relevant query forms from a large number of pre-generated query forms. It works well in the databases which have rich textual information in data tuples and schemas. However, it is not appropriate when the user does not have concrete keywords to describe the queries at the beginning, especially for the numeric attributes.

PROBLEM RECOGNITION

(i) *Most of the static forms do not support heterogeneous data:*

Modern and scientific database usually contain hundreds of heterogeneous data. The real world database contains various relations and attributes. The static forms don't support heterogeneous data and various relations. Predefined forms contain fixed number of data this may not satisfy the user needs all the times.

(ii) *Loading time gets delayed:*

Static tables contains various unused data these data occupies huge memory. So the loading time becomes higher. This paper proposes the creation of dynamic table, in which the user dynamically creates the required fields, hence number of unused data and memory will be reduced which in turn to reduces the loading time.

(iii) *Database Complexity:*

The insertion and deletion of fields in static table is a complex task. If we insert or delete fields in static table then the entire table schema should be restructured. Where as in dynamic table the fields are created dynamically, there will not be any complexity in adding and removing of fields.

PROPOSAL SYSTEM

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



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

perform many rounds of action (i.e., refining query conditions) before identifying the final candidates. The essence of DQF is to capture user interests during user interactions and to adapt the query form iteratively.

IMPLEMENTATION

The implementation process begins with preparing the plan for the implementation for the dynamic tables and query forms for database queries. This plan involves the actual activities that are to be carried out with the available resources and equipments. This phase is the final and critical stage which gains user's confidence of working effectively as it developed in the dynamic bases.

This paper includes the following functionalities which are segregated into various modules as follows.

- User Authentication
- Query Enrichment
- Query Execution

RANKING PURPOSE

Query forms are designed to return the user's desired result. There are two traditional measures to evaluate the quality of the query result: precision and recall. Query forms are able to provide different queries by different inputs, and different queries can output different query result and achieve different precision and recalls.

Query Construction

The goodness of the query form is determined by the query results generated from the query form. Based on this, we rank and recommend the potential query form components so that users can refine the query form easily. Based on the proposed metric, we develop efficient algorithms to estimate the goodness of the projection and selection form components. Here efficiency is important because DQF is an online system where users often expect quick response. A lot of research 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.

III. PROPOSED ALGORITHM

Algorithm

Query construction

Data: $Q = \{Q_1, Q_2, \dots\}$ is the set of previous queries executed on F_i .

Result: Q_{one} is the query of One-Query

begin

$\sigma_{one} \leftarrow 0$

for $Q \in Q$ do

$\sigma_{one} \leftarrow \sigma_{one} \vee \sigma_Q$

$A_{one} \leftarrow A_{F_i}$

$\cup Ar(F_i)$

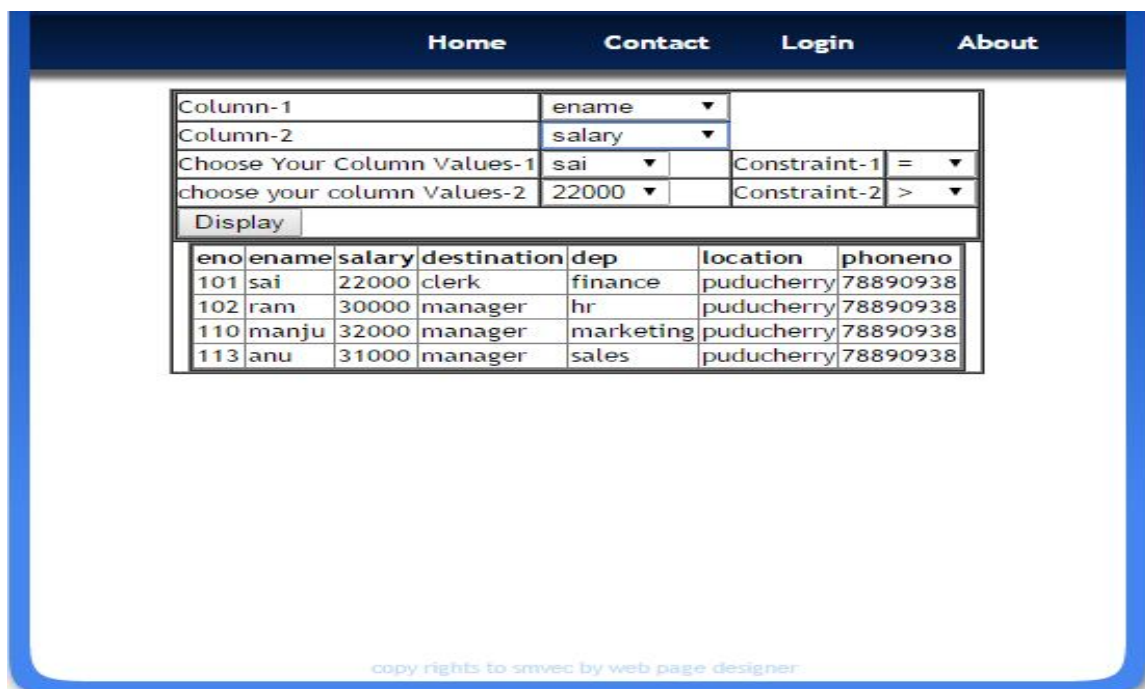
$Q_{one} \leftarrow \text{GenerateQuery}(A_{one}, \sigma_{one})$

After the expansion of both the events and the geographic terms, the query can be constructed using Boolean Operators. In order to achieve both high precision and recall, we setup four levels of queries, giving different weights for retrieved documents.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014



eno	ename	salary	destination	dep	location	phoneno
101	sai	22000	clerk	finance	puducherry	78890938
102	ram	30000	manager	hr	puducherry	78890938
110	manju	32000	manager	marketing	puducherry	78890938
113	anu	31000	manager	sales	puducherry	78890938

Fig2:screen shot for user created dynamic query form

IV. SIMULATION RESULTS

EVALUATION

The goal of our evaluation is to verify the following hypotheses:

- H1: Is DQF more useful than existing systems such as customized query form and static query form?
- H2: Is DQF more effective to rank projection and selection components then the baseline and the random method?
- H3: Is DQF effective to rank the related query form components in an online user interface?
- H4: Is DQF more useful for taking a good decision among the resultant values?

System Experimental Setup:

We implemented the dynamic query forms as a web based system using the ASP.NET code begins with C#. We used Ms-SQL Server as the database engine. All experiments were run using a machine with Intel Pentium IV 2.4GHz and running on windows 7.

Form Generation Approach:

We compared three approaches to generate query forms CQF: The Customized Query Form generation used by many existing database such as Microsoft access, Easy Query.

SQF: The Static Query Form approach proposed in [1]. A static query one that will be passed to the database and only simple queries should use the static query mechanisms. If you need more complex query means you should use DQF.

DQF: The Dynamic Query Form system proposed in this paper.

User Study Setup:

We conducted a user study to evaluate the usability of our approach. We recruited 20 participants of graduate students, UI designers, and software engineers. The system prototype is shown by Fig [2]. The user study contains 2 phases, a query collection phase and a testing phase. In the collection phase, each participant used our system to submit some queries and we collected these queries. These queries were used as query workload to train our system. In the second phase, we asked each participant to complete 12 tasks (none of these tasks appeared in the workload) listed in Table 2. Each participant used all three form generation approaches to form queries. The order of the three approaches

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

was randomized to remove bias. The databases collect a certain amount of historic queries so that we mainly consider the probability estimated from the historic queries.

TABLE 1
Usability Results

Task	Query Form	AC _{Min}	AC	AC _{Ratio}	FN _{Max}	FN	FN _{Ratio}	Success
T1	DQF	6	6.7	90.0%	40.0	3	7.5%	100.0%
	CQF	6	7.0	85.7%	60.0	3	5%	100.0%
	SQF	1	1.0	100.0%	35.0	3	8.6%	44.4%
T2	DQF	7	7.7	91.0%	65.0	4	6.2%	100.0%
	CQF	8	10.0	80.0%	86.7	4	4.6%	100.0%
	SQF	1	1.0	100.0%	38.3	4	10.4%	16.7%
T3	DQF	10	10.7	93.3%	133.3	6	3.8%	100.0%
	CQF	12	13.3	90.2%	121.7	6	4.9%	100.0%
	SQF	1	N/A	N/A	N/A	6	N/A	0.0%
T4	DQF	11	11.7	94.0%	71.7	6	8.4%	100.0%
	CQF	12	13.3	90.2%	103.3	6	5.8%	100.0%
	SQF	1	1.0	100.0%	70.0	6	8.6%	16.7%
T5	DQF	5	5.7	87.7%	28.3	3	10.6%	100.0%
	CQF	6	6.7	90.0%	56.7	3	5.3%	100.0%
	SQF	1	1.0	100.0%	10	3	30.0%	66.7%
T6	DQF	7	7.7	91.0%	61.7	4	6.5%	100.0%
	CQF	8	10	80.0%	61.7	4	6.5%	100.0%
	SQF	1	1	100.0%	23.3	4	17.2%	41.7%
T7	DQF	5	6.0	83.3%	48.3	3	6.2%	100.0%
	CQF	6	6.7	90.0%	50.0	3	6.0%	100.0%
	SQF	1	1.0	100.0%	18.3	3	16.4%	44.4%
T8	DQF	3	3.3	91.0%	21.7	2	9.2%	100.0%
	CQF	4	4.7	85.1%	26.7	2	7.5%	100.0%
	SQF	1	1.0	100.0%	38.3	2	5.2%	66.7%
T9	DQF	5	6.3	79.3%	31.7	3	9.5%	100.0%
	CQF	6	6.7	90.0%	36.7	3	8.2%	100.0%
	SQF	1	1.0	100.0%	106.7	3	2.8%	66.7%
T10	DQF	6	6.7	90.0%	43.3	4	9.2%	100.0%
	CQF	8	8.7	92.7%	63.6	4	6.3%	100.0%
	SQF	1	1.0	100.0%	75.0	4	5.3%	33.3%
T11	DQF	5	6.3	79.4%	36.7	3	8.2%	100.0%
	CQF	6	6.7	90.0%	50.0	3	6.0%	100.0%
	SQF	1	N/A	N/A	N/A	3	N/A	0.0%
T12	DQF	7	7.7	91.0%	46.7	4	8.6%	100.0%
	CQF	8	10.0	80.0%	85.0	4	4.7%	100.0%
	SQF	1	1.0	100.0%	31.7	4	12.6%	25.0%

User Study Result:

In this paper, we include some widely used metrics in Human-computer Interaction and Software Quality for measuring the usability of a system [8]. These metrics are listed in the Table 2.

TABLE 2
Usability Metrics

Metric	Definition
AC _{Min}	The minimal number of action for users.
AC	The actual number of action performed by users.
AC _{Ratio}	$AC_{Min} / AC \times 100.0\%$
FN _{Max}	The total number of provided UI function for user to choose
FN	The number of actual used UI function by the user.
FN _{Ratio}	$FN / FN_{Max} \times 100\%$
Success	The percentage of users successfully completed a specific task.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

In database query forms, one action means a mouse click or keyboard input for a textbox. One function means a provided option for the user to use, such as a query form or a form component. In a web page based system, FN_{Max} is the total number of UI components in web pages explored by the user. The smaller AC_{Min} , AC , FN_{Max} and FN are the better usability. Similarly, the higher AC_{Ratio} , FN_{Ratio} and Success are the better usability.

User Study Analysis:

Table 1 shows the average result of the usability experiments for those query tasks. As for SQF, we generated 10 static query forms based on the collected user queries for each database. The results show that user did not accomplish querying tasks by SQF. The reason is that, SQF is built from the query workload and may not be able to answer the ad hoc queries in the query tasks. DQF and CQF are capable of assisting users finish all querying tasks.

Static vs. Dynamic Query Forms:

If a query task is covered by historical queries in history, then SQF built on those historical queries can satisfy that query task. But the time and costs of using SQF and DQF to accomplish that task are different. Form complexity was discussed in [1] to evaluate the cost of using a query form.

Static execution requires configuring the database to run an SQL statement but delivers more consistent performance. Dynamic execution of SQL statements is more flexible because it does not require any special preparation on the database.

V. CONCLUSION AND FUTURE WORK

For future enhancement we would like to provide Decision Support System based on the user requirements. We would like to capture user preference as runtime feedback for effective performance. To enhance the query form further we would like to extend the query form even for non relational database and add the keyword search for ad-hoc queries.

In this paper we propose dynamic database creation and query forms based on the user requirements. By this dynamic database creation we may reduce the database size and complexity. The key idea of this paper is to generate query forms dynamically using the ranking components. The ranking are done based on the user preference. We capture user preference by user's frequent usage of queries and feedback. This paper also supports decision making system for end users by taking historical data of user's database.

REFERENCES

- [1] Jayapandian.M and Jagadish.H.V, "Automating the design and construction of query forms", IEEE Transactions on Knowledge and Data Engineering, Vol.21 (10), pp.1389-1402, 2009.
- [2] Agrawal.S, Chaudhuri.S, Das.G and Gionis.A," Automated ranking of database query results", In CIDR, 2003.
- [3]Bernstein.A and Kaufmann.E, "Making the semantic web accessible to the casual user: Empirical evidence on the usefulness of semiformal query languages". IEEE Transactions.
- [4]Embley.D.W," NFQL: The National Forms Query Language", ACM Transaction Database System, 1989.
- [5]Hristidis.V, Papakonstantiou.Y," DISCOVER: Keyword Search in Relational Databases", VLDB, 2002.
- [6]Jayapandian.M, Jagadish.H.V," Automated Creation of a Form based Database Query Interface." VLDB 2008.
- [7]Jayapandian.M, Jagadish.H.V," Expressive Query Specification through Form Customization". EDBT 2008.
- [8]Seffah.A, Donyaee.M, Kline.R.B, and Padda.H.K." Usability measurement and metrics: A Consolidated model. Software Quality Journal", Voll 14(2): 159-178, 2006.
- [9]Frakers.W.B and Baeza-Yates.R.B," Information Retrieval: Data Structure and Algorithms". Prentice-Hall, 1992.
- [10]Joachims.J and Radlinski.H,"Search engines that learn from implicit feedback", IEEE Computer (COMPUTER),Voll,40(8): 34-40, 2007.
- [11]Roy.S.B, Wang.H, Nambiar.U, Das.G and Mohania.M.K," Dynacet: Building dynamic faceted search systems over databases", March 2009.
- [12]Satlon.G and McGill.M. "Introducing to Modern Information Retrieval", McGraw-Hill, 1984.
- [13] ColdFusion. <http://www.adobe.com>
- [14] DBPedia. <http://DBPedia.org>
- [15] Easyquery. <http://devtools.korzh.com>
- [16] Freebase. <http://www.freebase.com>



ISSN(Online): 2320-9801
ISSN(Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

BIOGRAPHY



Prof. Mr. R. Ramakrishnan received his PG Degree in Master of Computer Applications, from Madurai Kamarajar University, Madurai in 2000, M.Phil in Computer Science from Periyar University and M.Tech in Computer Science and Engineering, from Pondicherry University, He has over 13 years of experience in teaching and 10 years in software industry. He is working as Assistant Professor in Department of Master of Computer Application at Sri Manakula Vinayagar Engineering College, Pondicherry. He has published and presented more than 25 papers in various International and national journals and conferences. His area of interest includes wireless networks and multimedia systems.



R. Manju obtained her BCA degree from Pondicherry University Community College Pondicherry, India. She is currently pursuing her MCA degree in at Sri. Manakula Vinayagar Engineering College, Madagadipet, India. Her areas of research interest accumulate in the areas of Software Testing.



S. Anitha obtained her B.Com degree from Barathidhashan Govt Girls College, Pondicherry, India. She is currently pursuing her MCA degree in at Sri. Manakula Vinayagar Engineering College, Madagadipet, India. Her areas of research interest accumulate in the areas of Software Engineering.