Public Transparency Database System (PTDS) - Based on Dynamic Query Form (DQF)

Yash Kumar
Information Technology Department.
Atharva College of Engineering
Mumbai, India

Vinit Salian
Information Technology Department.
Atharva College of Engineering
Mumbai, India

Akhil Talashi
Information Technology Department.
Atharva College of Engineering
Mumbai, India

Snigdha Wasnik
Information Technology Department.
Atharva College of Engineering
Mumbai, India

***Abstract*—Implementing static query form based on user's requirement is a difficult task. Hence, dynamic query form is used because of its flexibility when it comes to customizing Query forms to narrow down results in huge databases. PTDS is a concept that uses DQF because of its ability to handle large databases. Here by using the concept of relational databases, we were able to represent various attributes (characteristics) of an individual in such a way that it's easy to find a specific information about a specific person. The major goal is to give people the freedom to search others based on almost any characteristic, DQF now comes into the picture, since user can customize the Query form into a new query form, which now contains the fields that they can fill out, and discard the ones that they can't. Also, precision and recall is implemented to give the best and the following best results.**

Keywords—Flexibility, Narrow, DQF, Query Form, Precision And Recall, Precision, Recall, Dynamic Query Form, Databases, Relational Database, PTDS.

I. INTRODUCTION

In a database context, a form is a window or screen that contains numerous fields, or spaces to enter data. Each field has to hold a field label so that any user who views the form gets an idea of its contents. A form is more user friendly than generating queries to create tables and insert data into fields.Implementing static query form based on user‟s requirement is very difficult task. Because these queries are predefined and only shows constant results so user cannot get satisfactory result. Here, dynamic query form is changed according to the user need. Searching and retrieving database from the heterogeneous data is a critical task. The conventional databases do not satisfy the user desired instances. These databases contain constant fields and may not change according to the user‟s requirement. To get the satisfactory result, dynamic database proposed which create dynamic query forms. In dynamic database generating fields are specified by the user which is iterative process, at every apparent iteration the system generates ranked list for the fields and then user automatically add these fields to the query form.

II. LITERATURE REVIEW

Nowadays the databases and content of the web that we handle are enormous, so the expected results we get from those cannot be accurate.
In order to narrow down the desired result we implement DQF (Dynamic Query Form).
Purpose system is used to generate query forms dynamically to the user. From the large and complex database user can get different results by performing one query with different conditions at runtime. Using the ranking algorithm, components are suggested, which will be used to make customized query form as per user's convenience/requirement. The ranking are done based on the user preference.

After making a dynamic query form for a user to input user based searching criteria will make a search result be accurate and precise. This application can be used by large scale databases, where searching becomes too complicated and results are not much accurate and precise as needed by the user. As number of internet user increases companies and organization stores a large amount of data online to analyze according to their needs. This DQF will help them fetching a needed results which can help them to make some useful information. From the large and complex database user can get different results by performing one query with different conditions at runtime. Here, dynamic query form is changed according to the user need. Searching and retrieving database from the heterogeneous data is a critical task. Searching and retrieving database from the heterogeneous data is a critical task. The conventional databases do not satisfy the user desired instances. These databases contain constant fields and may not change according to the user‟s requirement.

III. COMPONENTS

1) Front End: ASP.Net:

ASP.NET is a Web application framework developed and marketed by Microsoft to allow programmers to build dynamic Web sites, Web applications and Web services the .NET Framework, and is the next to Microsoft's Active Server Pages technology. ASP.NET is developed on the
Common Language Runtime allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET extension framework lets ASP.NET components to process SOAP messages. ASP.NET Web pages, known officially as Web Forms, are the building block for application development.] Web forms are mostly contained in files with an ".aspx" extension; these files almost usually contain static (X)HTML markup, as well as markup showing server-side Web Controls and User Controls where the developers place all the required static and dynamic contents for the Web page.

2) Back End: SQL Server 2005:

Microsoft SQL Server is a RDBMS server, developed by Microsoft: it is a product whose primary motive is to store and obtain data as requested by other applications, be it those on the same system or those running on another computer across a network. There are many different versions of Microsoft SQL Server aimed at different public and for different loads (ranging from small applications that store and retrieve data on the same system, to millions of users and computers that use huge amounts of data from the Internet at the same time). True to its name, Microsoft

IV. EQUATIONS

1) Ranking Metric:

Here for ranking the tables we use the concept of precision and recall.
Following is the table depicting all the components of the upcoming equation.

Table 1

Symbols and Notations

F	query form
$\mathcal{R}_F$	set of relations involved in F
$\mathcal{A}$	set of all attributes in $\bowtie(\mathcal{R}_F)$
$\mathcal{A}_F$	set of projection attributes of query form F
$\mathcal{A}_r(F)$	set of relevant attributes of query form F
σ_F	set of selection expressions of query form F
$\mathcal{OP}$	set of relational operators in selection
d	data instance in $\bowtie(\mathcal{R}_F)$
D	the collection of data instances in $\bowtie(\mathcal{R}_F)$
N	number of data instances in D
$d_{\mathcal{A}_1}$	data instance d projected on attribute set $\mathcal{A}_1$
$D_{\mathcal{A}_1}$	set of unique values D projected on attribute set $\mathcal{A}_1$
Q	database query
D_Q	results of Q
D_{uf}	user feedback as clicked instances in D_Q
α	fraction of instances desired by users

Given a set of attributes A and a universe of selection expressions σ, the precision and expected recall of a query form F=(AF, RF, σF, (RF)) are PrecisionE(F) and RecallE(F) respectively, i.e.,

Equation I:

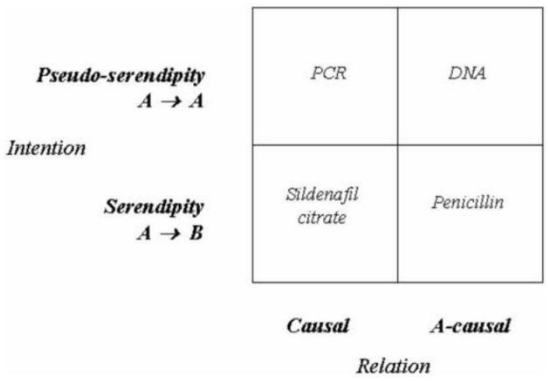
$$Precision_E(F) = \frac{\sum_{d \in D_{\mathcal{A}_F}} P_u(d_{\mathcal{A}_F}) P(d_{\mathcal{A}_F}) P(\sigma_F|d) N}{\sum_{d \in D_{\mathcal{A}_F}} P(d_{\mathcal{A}_F}) P(\sigma_F|d) N}, \quad (1)$$

$$Recall_E(F) = \frac{\sum_{d \in D_{\mathcal{A}_F}} P_u(d_{\mathcal{A}_F}) P(d_{\mathcal{A}_F}) P(\sigma_F|d) N}{\alpha N}, \quad (2)$$

2) The Structure of Serendipity.

This work by Mark de Rond was carried out in the year 2005 . This paper explains the structure of serendipity , Serendipity is mistakenly used as synonymous with chance events, luck. It is thus not surprising that serendipity remains comparatively under-researched. After all, how is one supposed to unlock the „black box" of this chance? Rather than being synonymous with any chance, serendipity results from identifying matching pairs" of those events that are put to practical use. With this etymologically accurate definition is in mind, serendipity thus describes a reliable capability, not a particular event. It follows that human calculation, and not probability, is properly the focus of attention. Drawing on its 16th century etymological origins, we unravel the four serendipitous innovations in the science field to illustrate the nature of serendipity. In developing the argument, we shall propose a novel typology. Hereby we conclude by exploring implications for research and practice.[2]

Fig I: Structure of Serendipity:



V. METHOD

1) Difference between Static and Dynamic Query Form: If a query is covered with multiple historical type of queries in history, then the SQF built on those historical queries can satisfy that particular query task without any hassle. But the expense of using SQF and DQF technologies to accomplish that task are different. Form-Complexity was initially proposed to evaluate the cost of using a query form. It is the sum of the total number of selection , projection, and relations, as shown below:

Equation II:
Form − Complexity(F) = |AF|+|σF|+|RF|.

DQF generates atleast one customized query form for each query. This result shows that, in order to satisfy various query tasks, the generated query form has to be alot more complex.[1]

2) Flow chart for DQF concept that supports PTDS: (Execution)

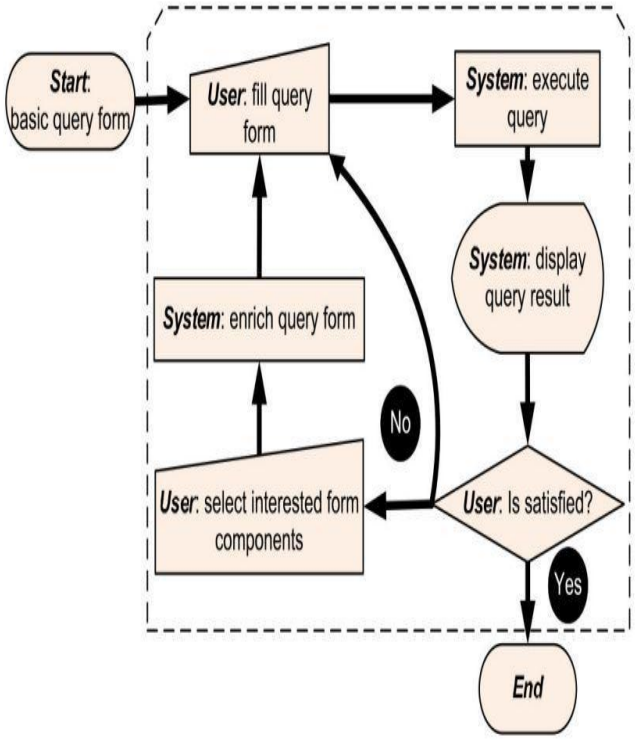


Fig II: Flow chart for DQF:

1) Effectiveness:

We now compare the ranking function of DQF with two other ranking methods: the baseline and the random method. The baseline ranks projection and selection attributes in ascending order of schema to the current query form. For query condition, it usually chooses the most frequent used condition in the training set for that attribute. The random method randomly suggests one query form element. Ranking projection elements: Ranking score is a supervised way to measure the accuracy of the recommendation. It is obtained by comparing the ranking with the optimal ranking. In the optimal ranking, the selected component by the user is ranked first. So ranking score calculates how far away the actual selected component is ranked from the first. The formula of ranks score is as follows:[6]

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 users 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

Table 3
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	93.5%	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	92.0%	63.3	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%

Equation III:
RankScore(Q, Aj) = 1 log(rˆ(Aj)) + 1 ,

where Q is a trial query, Aj is the j-th projection attribute of Q, rˆ(Aj) is the rank of Aj. Fig. 4 shows the average rank scores for all queries in the workload. We compare three methods: DQF, Baseline, and Random. The x-axis depicts the portion of the training queries, and the rest queries are used as trial queries. The y-axis indicates the average ranking scores among all the testing queries. DQF always outperforms the baseline and random method. The gap also increases as the portion of training queries increases because DQF can better utilize the training queries. [6]

2) Proposed System:

The system we intend to build is based on database which can be based on government citizen records or community records or company employee records.

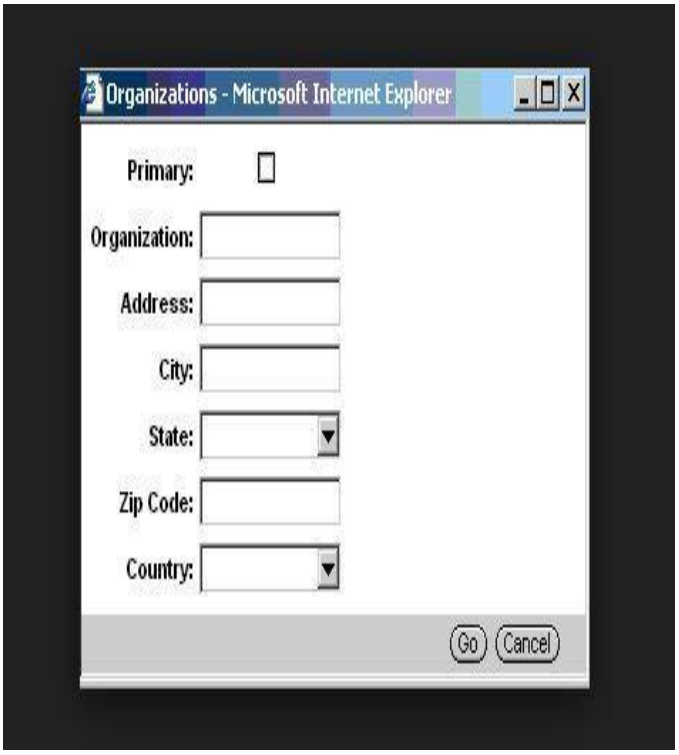


Fig 3: Query Form:

Since everyone has some or the other government approved proof of individuality, for eg: Passport, aadhar card,pan card etc. These can be linked together in order to find an individual regardless of how many people with the same or similar name exist. That is where DQF comes into the picture, for instance, if there are ***n*** number of people with the same name, DQF is used to narrow down the search result via several iterations depending on the complexity of the database.

The user can modify the query form and select those attributes for form components which the user has correct knowledge of. Since 2 people cannot share every detailed features, the result comes down to 1.

Here after the database has been acquired, it will be processed to eliminate redundancies and the attributes will be joined or linked to the respective data input.

The administrator is responsible for handling the linkage of the databases.

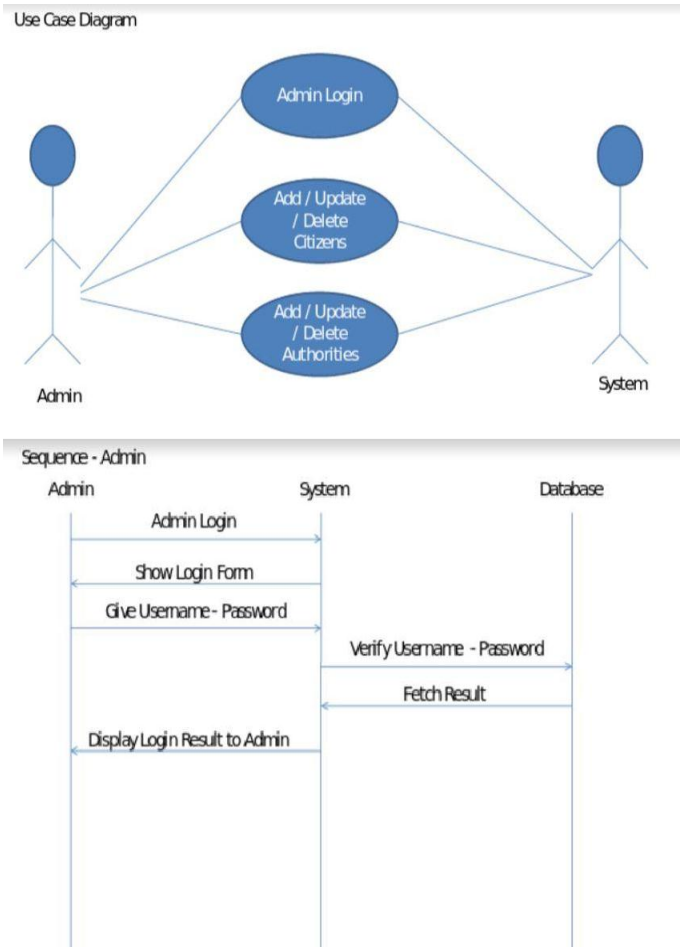


Fig 4: Admin Use case and Sequence Diagram:

Levels of security are assigned depending on the authority of an individual.

For instance, an official person will be allowed to access all the details of all the people on that engine.

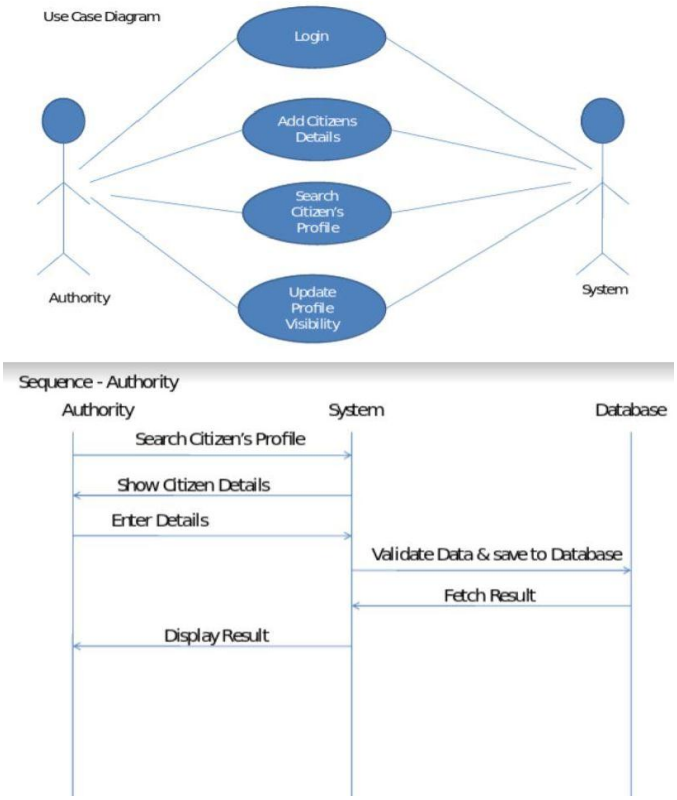


Fig V: Authority Use case and Sequence Diagram:

Whereas, the common citizen will be allowed to access only that information which can be deemed safe for everyone, because any sensitive information can cause threat.

Now, depending on each user's preference, their information will be made public. This can be well adapted in any scenario, from past employment to hospital records to criminal records. This proposed system will make everything transparent. Thereby allowing public to know everything about their politicians, doctors, social servants, potential employers. This attempt could suppress corruption.

REFERENCES

[1] Comparison of Static and Dynamic Query Form or Database Queries. Authors: Shinde M.M, Patil B.M. July 2016.
[2] The Structure Of Serendipity. Author: Mark de Rond. 2005.
[3] The Relationship Between Precision-Recall and ROC Curves.Authors: Jesse Davis,Mark Goadrich. 2016.
[4] A Review on Dynamic Query Forms for Database Queries. Author:Priyanka P.Nikam. November 2014.
[5] Creation of Dynamic Query Forms and Ranking of its Components based on User"s. Authors:Greeshma Radhakrishnan, Dr. S. Sasidhar Babu. August 2015.
[6] Dynamic Query Forms for Database Queries Authors:Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen. SEPTEMBER 2014.
[7] K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh, "Usher: Improving data quality with dynamic forms," in Proc. ICDE, Long Beach, CA, USA, , pp. 321–332.MARCH 2010
[8] T. Joachims and F. Radlinski, "Search engines that learn from implicit feedback," IEEE Comput., vol. 40, no. 8, pp.34–40,AUGUST 2007.