# Comparison of Static and Dynamic Query Form for Database Queries

Shinde M. M.
PG Student
MBESociety's of college of
engineering, Ambajogai
Maharashtra, India

Patil B. M.
Professor
MBESociety's of college of
engineering, Ambajogai
Maharashtra, India

## ABSTRACT

Implementing static query form based on user's requirement is very difficult task. Because these static queries are predefined and shows only constant results so user cannot get satisfactory result. Here, static versus dynamic query forms are compared where static query form is constant and dynamic query form is changed according to the user need. Searching and retrieving database from the heterogeneous data is very critical task. The traditional 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 user 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 iteration system generates ranked list for the fields and then user automatically add these fields to the query form. The ranking of fields is based on user's preference. To view the result user fills out the query forms at each iteration and submit the results.

## Keywords

DQF (Dynamic Query Form), CQF (Customized Query Form), SQF (Static Query Form).

## 1. INTRODUCTION

The Traditional database is a static and designed by professionals to maintain information [1].Now a day's DBPedia[2],Freebase [3]web databases become very complex which contains many relations and attribute and they may confuse to the user. Those complex databases are created to handle database with large size. It is very tedious task to design bunch of static query form to satisfy various database queries on those complicated databases. There are number of existing database development tools, such as Easy Query [4], Cold Fusion [5], SAP and Microsoft Access provide several ways to the users to create customized queries on that databases. If a user does not have knowledge of SQL database then thousands of records and attributes may confuse to user. Dynamic Query Form system which is query interface able to generate dynamically query forms for the user. The user will interact to the system iteratively until his/her satisfaction. Each iteration of user interaction has two types.

- Query Form Enrichment: DQF (Dynamic Query Form) generates query form according to the user. Every iteration DQF generates ranking list of form so user can select the form according to his/her need.
- Query Execution: The user fills and submits the query form then this dynamic query form executes the query and display the results.DQF shows the result and user gives the feedback about that query result. The major goal of the DQF is to work with large databases and user can get many results by executing one query even if the user is not familiar with the query syntax or SQL structure. It make user easy to access or modify data.

In this paper dynamic query form generates the query form according to the user's desire. Static query form generates the query form for only simple queries where as dynamic query form generates query form for more complex databases. At every execution time the user gets different updated results, where static query does not updated because these queries are constant.

The remaining paper is organized as follows Section 2 describes the related work. Section 3 gives the system overview. Section 4 describes the algorithm. Section 5 gives the purpose of ranking metric. Section 6 describes the experimental results, and finally Section 7 ends with the concludes the paper.

## 2. RELATED WORK

Nandi and Jagdish [6], Based on query workload a visual interface developed to guide the user to type the queries. This auto completion approach proposed for database queries.

Jayapandian[7],proposed customize queries on database.Todays many development tools like SAP,Easy Query, Cold Fusion provide let user develop customized queries. The problem is that, customized queries based on user's manual editing, if there are thousands of relations and attribute of data then it would may confuse the user, if they are not familiar with database. Another one problem is that, if database is complex and huge, it is hard for user to find specific database attribute.

Jagdish [8], proposed workload driven method. To find the representative queries clustering algorithm applied on historical queries. Clustering is used to show high level view of the query result, because it outputs only compressed result table. User clicks on his interested cluster to view the detailed data instances.

Jayapandian and Jagadish [9], proposed form-based query interface. When user fills relevant information and obtains desired result by submitting the form. They present framework for generating forms in an automatic way.

Chu, et al. [10], proposed, to find relevant query user can input combination of many keywords. In advance, system generates many query form. This system allows keyword based search, optimization, and query processing of relational database of dynamically generated query forms.

Eirinaki and Polyzotis [11], Recommender system produces a list of recommendations in two ways through collaborative or content based-altering for database exploration to recommend database query components. In collaborative filtering approach they treat SQL (Static Query Language) queries as items. Similar queries recommended to related users one issue is that they do not consider goodness of query result. In existing system, recommendation is a complete query and in proposed system, recommendation is a query component for each iteration.

Roy, et al. [12], proposed Dynamic Faceted Search is a search engine. According to navigation path it searches relevant facets for the user. According to the database schema system generates dynamic faceted search engine interfaces. Here, on the query form components they focus only projection and selection. Join is not handled by dynamic query form because these are not part of query form and is invisible for users.

In Usher: Traditional databases form design is expensive. Usher is a new approach to form design and data quality. This is end-to-end system that automatically generates data entry forms [13].

## 3. SYSTEM OVERVIEW

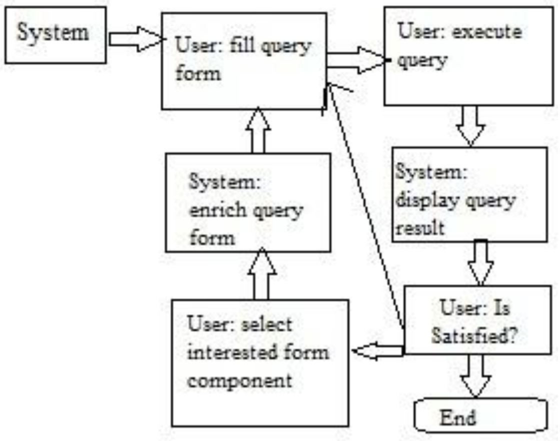
System
User: fill query form
User: execute query
System: enrich query form
System: display query result
User: select interested form component
User: Is Satisfied?
End

**Fig 1.Flowchart of Dynamic Query Form [1]**

Fig1 shows flow chart of dynamic query form. Multiple, heterogeneous and large data is maintained by rapid development of internet information databases and web databases. These databases contain large number of relations and attributes.Desigining static query forms are very difficult. They are unable to satisfy various queries from users on those scientific databases. DQF is new interface introduced to consider user's need at run time. This generates query forms dynamically at run time. The basic function of DQF captures user's preference and rank query form components. According to user interest dynamic query form has two interactions. First one is Query form enrichment and second one is query execution. Creating a query form is an iterative process. User selects the desired form component from the current form. Then user fills out the current query form and submits it to view the result. The user then gives feedback about the query results.

Static query forms are predefined by professionals and developers. Only those have the knowledge of the information system.Todays databases become very large, so static query forms does not satisfy user's desire. This is the limitation of static query form so dynamic query forms are generated. This helps to user to dynamically generate query forms.

At every execution user will get different and huge amount of results. User does not have much time to check whether he gets desired result or not. To avoid this problem, the result will be shown in only compressed table.

Many databases queries output a huge amount of data instances because database is very large. It is very difficult to decide which query form is desired or not. Clustering algorithm applied to find the representative queries. Clustering outputs only compressed result table and then user clicks on his interested cluster to view the detailed data instances. Fig 2 shows User Action.

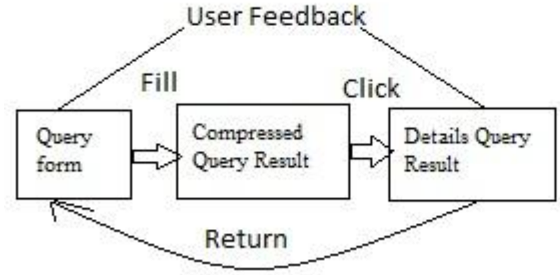

**Fig 2.User Action [1]**

The clustering problem is a main and difficult problem for the data stream. There are some one-pass clustering algorithms have been developed for the problem in data stream. One-pass clustering algorithms are used for generating the compressed view efficiently. Here, because of the efficiency issue incremental data clustering framework is used. Different data clustering methods produces different compressed views for the user. There are different clustering methods effectively used for different data types. Clustering is used to provide better view to the user of the query result. Compressed view is used to collect the user feedback for query.

## 4. ALGORITHM

```
Data: Q = {Q1, Q2 … g} 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
            $\sigma one \leftarrow \sigma one \vee \sigma_Q$
        Aone    $\leftarrow$AFi $\cup$ $A_{r(Fi)}$
         Qone$\leftarrow$GenerateQuery ($A_{one}$, $\sigma_{one}$) [1]
```

The one query can be constructed using Boolean operator. To achieve high precision and Recall here four levels of queries

Queryform1, Queryform2, Queryform3, Queryform4 has been set up with different inputs for retrieving documents.

## 5. RANKING METRIC PURPOSE

Query forms are mainly used for the user interaction with the database. Query forms are designed to return user's desired result at run time. Precision and Recall are the two measures to evaluate the quality of the query result. When different inputs given query form outputs different queries, and they output different query results and achieve different precisions and recalls, to use expected precision is the expected proportion of the query results which are interested by the current user and expected recall is the expected performance of the query form. Ranking projection form components, and ranking selection form components are the two ranking score estimation phases. Ranking projection form components consist two levels. First is ranked entities and second one is ranked list of attributes in the same entity. In ranking selection form components if selected attribute is not relevant then selection attribute is useless. Creating the selection components, the system should find out the relevant attributes first. Now, let's see two strengths expected precision and expected Recall for query forms. To calculate expected Precision and expected Recall of a query form F=($A_F$,$R_F$,$\sigma_F \bowtie R_F$) where F is the query form,$A_F$ is the projection attribute of F,$R_F$ is relations involved in F and $\sigma_F$ is the selection expression of F.For these $\text{Precision}_E$(F) and $\text{Recall}_E$(F) are

$$\text{Precision}_E(F)=\frac{\sum dAF \ \ pu(dAF)\,p(dAF)\,p(\sigma F|\,d)N}{\sum dAF\,p(dAF)\,p(\sigma F|\,d)N}\text{……….(1)}$$

$$\text{Recall}_E(F)=\,)=\frac{\sum dAF \ \ pu(dAF)\,p(dAF)\,p(\sigma F|\,d)N}{\sigma N}\text{……….(2)}$$

The Equation (1) and (2) numerator denotes the expected number of data instances and denominator denotes number of instances in the query result.$p_u$ ($d_{AF}$) is user interest on instance of in the query result.D is the data instance in relation $R_F$.while summing up over all data instances gives the expected number of data instances. N is the expected number of rows in D where N will cancel in both equation, so there is no need to consider N when estimating precision and Recall.

## 6. EXPERIMENTAL RESULT

Here compared Static query form and dynamic query form query results. Here consider the differences between static query form and dynamic query form which are the best for application. Static Query Form is SQL (Static Query Language) statements in an application that do not change at runtime. They are hard-coded into the application. At Every query execution you could get the same result. Following are the two figures Fig3 and Fig4 which one shown graphical representation of the experimental result where Y-axis shows the Running Time (Milliseconds) and X-axis shows number of data instances in query result.QueryForm1, QueryForm2, QueryForm3, and queryForm4 are the Four Static queries from these static results generated. By comparing two databases NBA database and Car database two static results has generated.

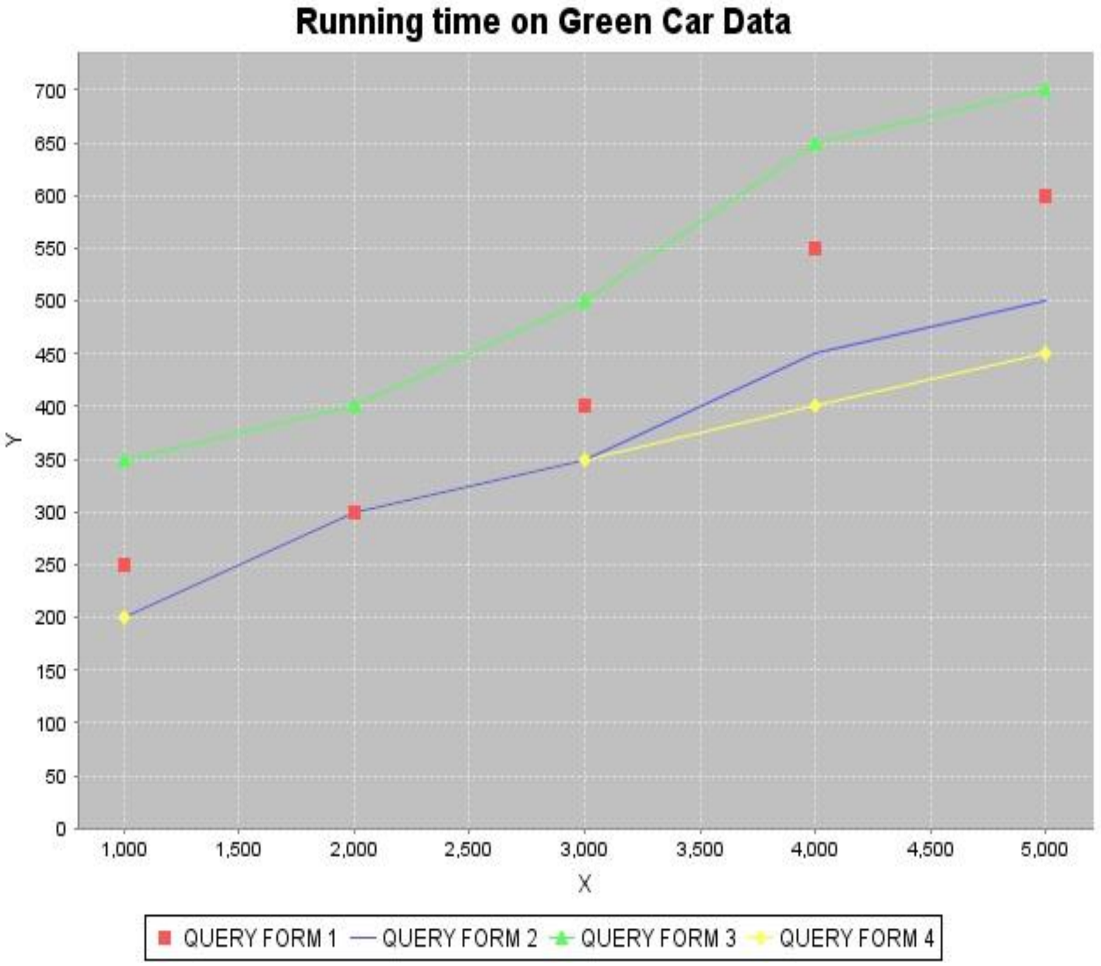

**Fig 3 Static Graph for Green Car database**

Above Fig3 shows the static graph for Green Car database. These databases contain 5000 records. In this work 4 static queries with large result size generated based on the collected user queries for each database. The only difference of static and dynamic query form is that in static query form execute different queries with different result size and the result remains same at every execution it does not change. For fetching 5000 records it requires 0.7 second running time. The execution time grows with respect to the query result size.

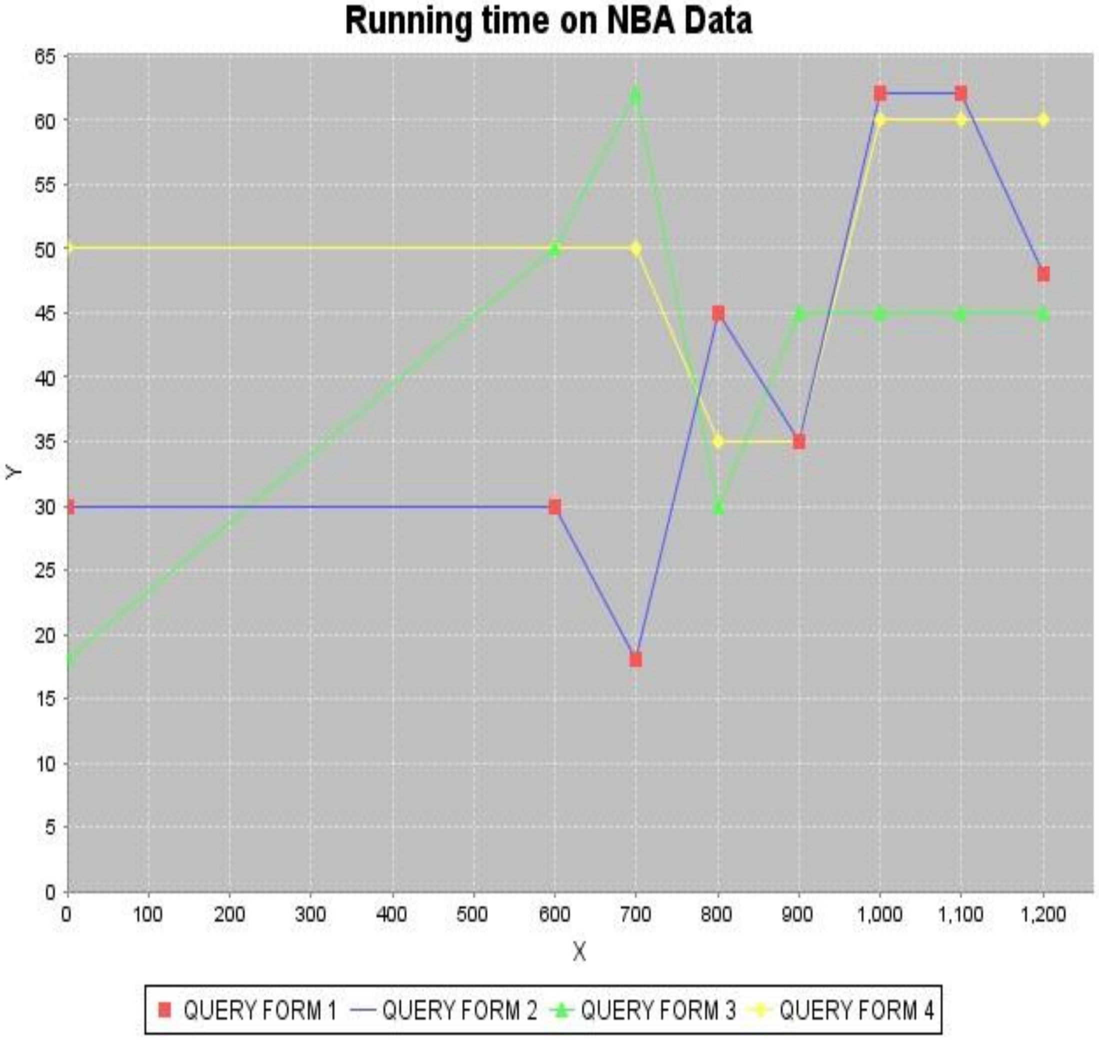

**Fig4. Static Graph for NBA Database**

Above Fig4 shows the static graph for NBA database. These databases contain 3000 records. Execution time require less than 0.067 second for NBA database when the results contain less records. When Comparing two databases with different records the execution time increase for high records and execution time decrease for less records with respect to the query result size but result of different 4 queries are constant at every execution.

Dynamic Query Form is SQL (Static Query Language) statements that are executed at runtime. Dynamic Query form generation is itself is dynamic. Dynamic Query cannot be Hard-coded so applications allow user to enter their own queries.So, user can get different results at every query execution. Following are the results generated from the query. Where X-axis shows the Running Time (Milliseconds) and Y-axis shows number of data instances the cost of ranking projection and selection components of DQF are depend on the current form components and the query result size. The results show that the execution time grows with respect to the query result size. There are two databases. Both databases compared by taking different query size.

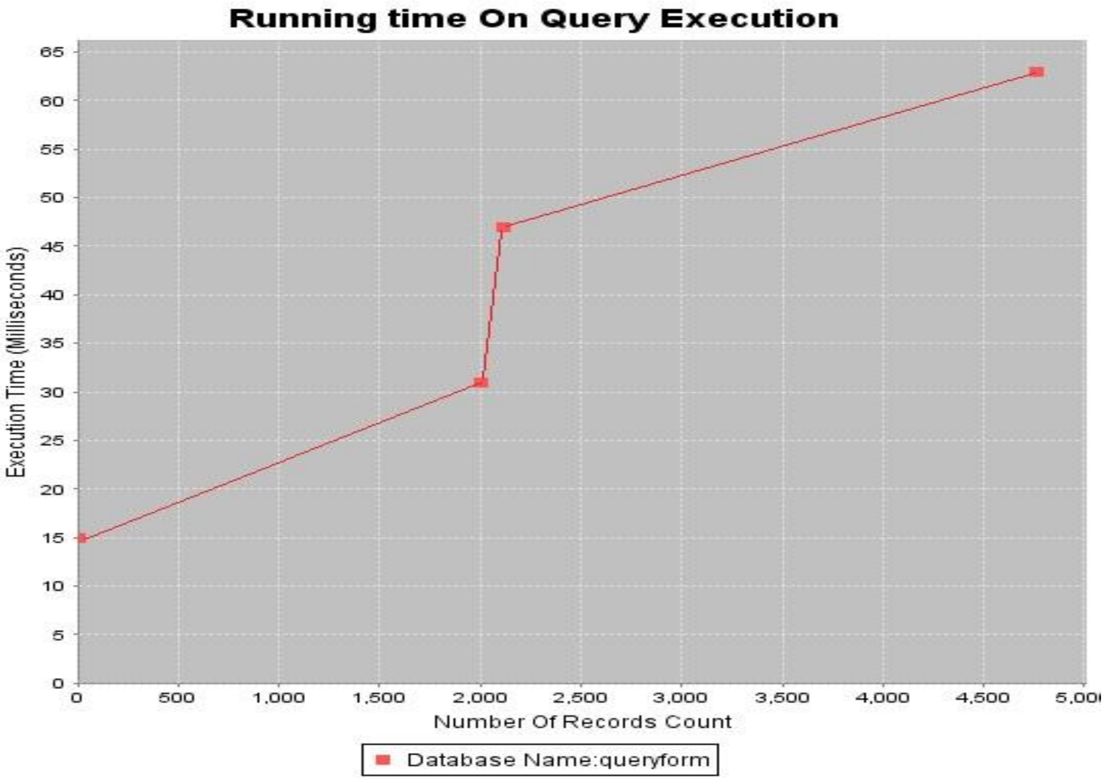

**Fig5. Dynamic Graph for NBA Database**

Above Fig5 shows the Dynamic graph for NBA database. These databases contain 5000 records. Here no hardcoded queries are written. One query can execute by different way. User can execute one query by giving different conditions. At every execution of query it shows different results to the user. Dynamic queries with large result size generated based on the collected user queries for each database. Fig5 shows query form database which contains 5000 records fetched from dynamically and it takes 0.063second for NBA database.

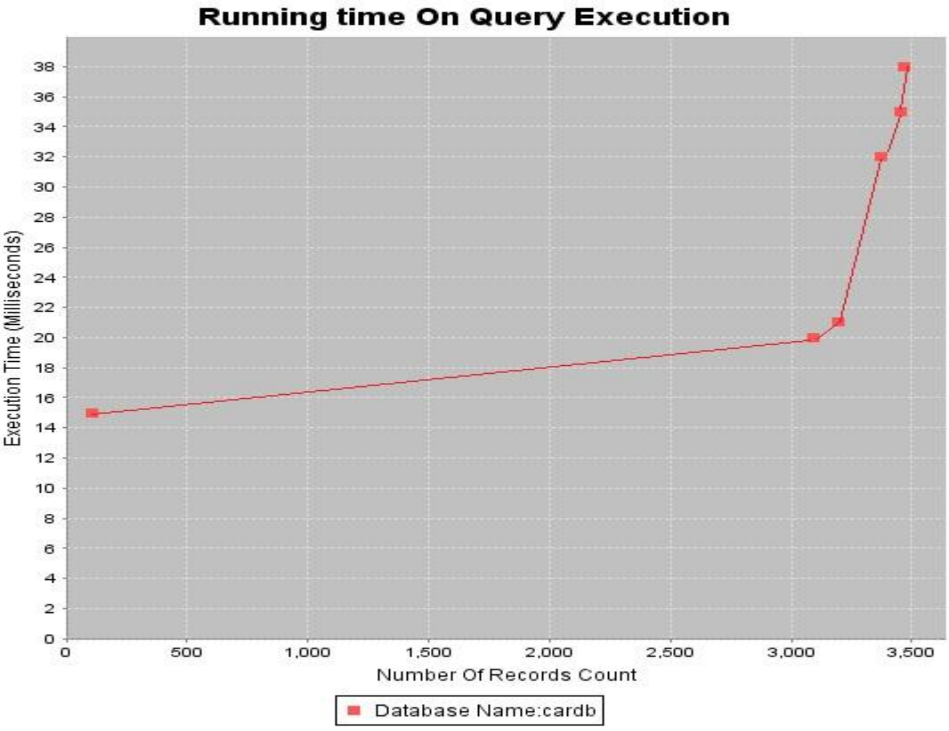

**Fig6. Dynamic Graph for Green Car Database**

Above Fig6 shows the Dynamic graph for Green Car database. These databases contain 3500 records. Green Car database contain less records so it requires less execution time to fetch these records. Here no hardcoded queries are written.

One query can execute by different way. At every execution of query it shows different results to the user.Fig6 shows Green Car database contains 3500 records and it took 0.038 second.

## 7. CONCLUSION

In this work, static and dynamic query form comparison has shown. Dynamic query Form approach 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 component query forms generate dynamically. The ranking are done based on the user preference.

As for the future work, this system to develop multiple methods to capture the user's interest for the queries besides the click feedback. For example text-box can be added for users to input some keywords queries. Dynamic query form can extend for non-relational database.

## 8. 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 2013.

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

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

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

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

[6] A.Nandi and H.V. Jagdish. Assisted querying using instant response interfaces. In proceedings of ACMSIGMOD, pages 1156-1158, 2007.

[7] 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.

[8] 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.

[9] M. Jayapandian and H. V. Jagadish. Automating the design and construction of query forms. IEEE TKDE, 21(10): 1389- 1402, 2009.

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

[11] 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.

[12] S. B. Roy, H. Wang, U. Nambiar, G. Das and M. K. Mohsnia. Dynacet: Building faceted search systems over databases. In proceedings of ICDE Conference, pages 1463-1466, Shanghai, China, March 2009.

[13] 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.