

Dynamic Query Forms for Database Queries

Dekshmukh.G¹, Kolase.A², Koshti.S³, Varma. P⁴ & G.M.Dahane⁵
^{1,2,3,4,5}(Dept of IT, P.D.V.V.P.C.O.E, SPPU,MS, India)

Abstract: Database maintains a large and heterogeneous data, with large number of relations and attributes. This query form is one of the most widely used user interface for querying database. Traditional predefined query forms are not able to satisfy various ad-hoc queries for users on those databases. It is very difficult to design a set of static query forms to assure various ad-hoc database questions lying on that complex database. There is a need of such a system which generates query forms dynamically according to the users' need at run time. The main purpose of this paper is to generate random query formulations for database queries, a novel database query form interface, which is able to dynamically produce query forms, for both relational and non-relational data.

Keywords: Database, Random query, Dynamic Query, relational data

I. Introduction

1.1 Aim

The generation of query forms is an iterative process and is guided by the users. At each iteration, the system automatically generates a ranking list of form components and the user then adds the desired form component into the query form. The ranking list of form components is based on the captured user preference. A user can also fill the query and then submit the query to view the result at each iteration. In this way, a query form could be dynamically refined till the user is satisfied with the query result. Our experimental evaluation and user study demonstrated the effectiveness and efficiency of the system.

1.2 Objectives

1. DQF is the main proposal of the system, is an oval database query form interface, which dynamically generates query forms.
2. The system generates the efficient algorithm to estimate the goodness of projection and dynamically generates query forms.
3. The key idea is to use a probability model to rank form components based on user preference. They capture user preference using both historical and run-time feedback such as click-through method.
4. Ranking form components also make the use of customized query forms.

1.3 Project Scope

DQF, in various database query form interfaces which are able to dynamically create query forms. The significance of DQF is to capture a user's choice and classify query form components to support users to make a conclusion. The creation of query forms is a repetitive process and is conducted by the user. In each iteration, the system automatically creates a classification list of form components and the user adds the desired form component into the query form. The classification form component is based on the captured user choice. A user also fills the query and delivers queries to view the query output at each step. Thus, query forms could be dynamically refined till the answer with the query output. A probabilities model is developed for estimating the excellence of a query-forming DQF.

1.4 Relevant Theory

The system proposes dynamic query forms. In this

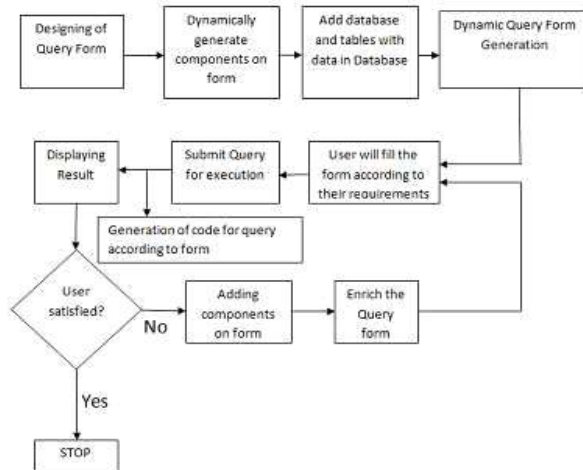


Fig 1: Flowchart of DQF

In this system we can generate query forms like control panels, aggregation functions, table etc. These are all dynamically generated components. After that generation of component the system adds database and components with respect to components. The user fills the query according to their own requirements and user query submitted for the execution. If the user requirements are satisfied the query proceeds for the further execution and the results are display. But if the user is not satisfied their query result then again fills the query form. Again the complete process is follow. The new component are added for the given query form. The ranking enrichment system will define how much time the query is executes and according to it ranking is given to the query. This query is done by click through method.

II. System Architecture

In this system, the database contains the very few primary attribute which is used to from basic query. Formation of basic query is first step. The user fill the query and generate the query and generate the result. The process of formation of basic query is continue until the user is satisfied with the query result and process is end otherwise the user is again fill the query these process is continuous. The system provides result of generation of query.

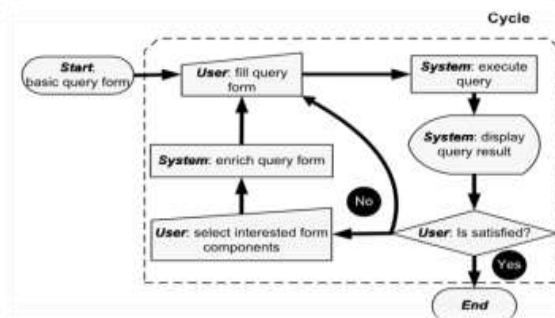


Fig 2: Work-flow of DQF

Implementation Detail:

Modules:

For an implementation of proposed system is following four module are use:

1. Query Forms Enrichment
2. Query Executions
3. Customized Query Forms
4. Database Query Recommendation

2.1 Query Forms Enrichment:

a) DQF is recommends a ranked list of query form components to the user.

b) The user selects the desired query forms component to the current query forms.

2.2 Query Execution:

- a) The user fills the current query form and submits it.
- b) DQF executes the dynamic query and shows the result.
- c) The user provides feedback about the query result.

2.3 Customized Query Form:

For creation of customizes query they provide the visual interface for the developer .In this query the is not easy for non-technical user because user is not familiar with their database It is only use full for professional developer who are familiar with their database, that is who are not familiar with database for large and complex database user select the attribute present in the database and create desired query forms.

2.4 Database Query Recommendation:

Recent studies introduction collaborative to recommended database query component for database exploration. They treat SQL queries a system in the collaborative filtering approach and recommend similar queries to related user.

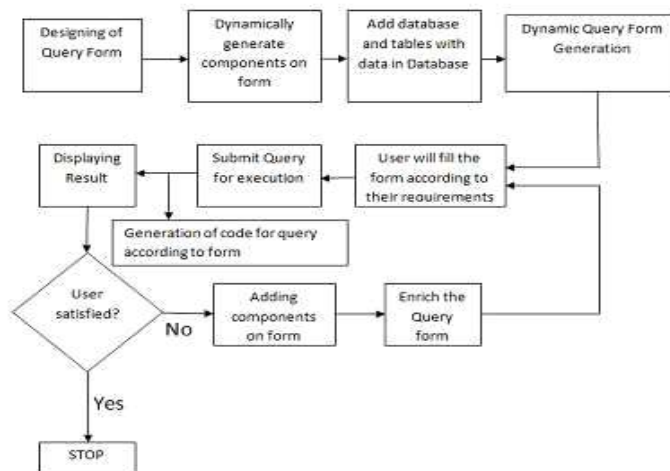


Fig 3: Flowchart of DQF

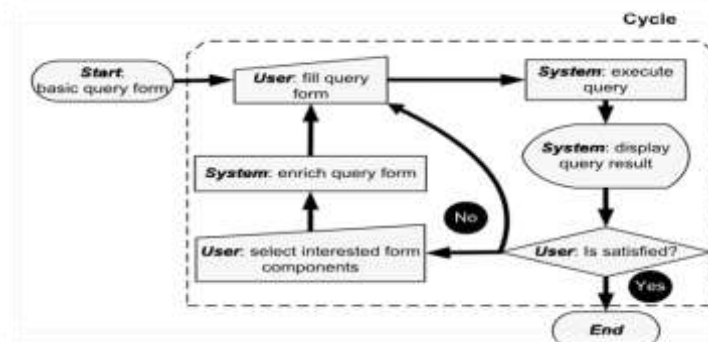


Fig 4: Work-flow of DQF

III. Conclusion

Query interfaces play a most important role in determining the usefulness of a database. user-friendly querying method is provided by form-based interface that are widely used in large and complex database .In the proposed system user interact with the system and form the dynamic query as per requirement. The proposed system provides higher success rate and simple query for user. We capture user feedback at run-time by click-through method. As future work, In future proposed system works with non relational data. In the future we plan to develop multiple methods to capture the users interested for the queries besides the click feedback. For instance, we can add a text-box for users to write some input keywords queries.

References

- [1] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen Paper: “Dynamic Query Forms for Database Queries” <http://www.adobe.com/products/coldfusion/>.
- [2] Gopi Krishna Lakkasani Balakrishna Nayudori Paper “ Random Query Formulation for Database Queries “ [.http://DBPedia.org](http://DBPedia.org).
- [3] Priyanka P. Nikam Paper “A Review on Dynamic Query Forms for Database Queries” <http://www.ijarccse.com>