Dynamic Query Form Generation On Multiple Database

Ms. Kalyani Gaikwad[1] and **Prof. A. M. Jagtap**[2]

[1]*Department of Information Technology SKNCOE, Pune*
[1]*Savitribai Phule Pune University, India.*

[2]*Department of Information Technology SKNCOE, Pune.*
[2]*Savitribai Phule Pune University, India.*

***Abstract*—Now a day's web data is getting increased tremendously and as per the database and its users are also getting increased. Traditional data query forms are not able to solved Ad-hoc queries, to maintain and solve such huge data DQF (Dynamic data forms) are introduced. By using DQF system user are capable of dynamically generating query forms for users. The essence of DQF is to capture user's interests during user interactions and adapt the query form interactively. It generates the Ranking on the base of the user feedback collection. So, by using this system non-technical user can also able to query in the database without having any technical knowledge of SQL.**

***Index Terms*—Query,Database,DQF,Form**

I. Introduction

Query forms are the mostly used forms to interact with the huge amount of the database and its relational attributes. As the traditional database applications are having the Query form design by the developer and database admin.[1] So as the size of database getting increase it is difficult to design such static Query forms for complex databases.[3] Some of the modern tools are there which customized Query such as SAP, MSaccess. By using such tools user can manually edit and develop the Query as he required. But if the user is not having knowledge of the Database and its syntax, Data attributes and there relations will confuse him.[2] Then it is difficult to use the Such database system. So, in this system the Dynamic Query forms are introduces which provides additional benefits as: 1.Dynamic Runtime Query generation 2. Iteratively Query Form refinements 3.Collection of the feedback from the users 4.Ranking on the bases of the feedback 5.Genration of Query as per user interest.

A. Existing System

Customary query forms are outlined and pre-characterized by engineers or DBA in different information administration systems. With the fast improvement of web information what's more, logical databases, current databases turn out to be exceptionally vast and complex. Along these lines, it is hard to plan a set of static query forms to fulfill different specially appointed database queries on those mind boggling databases. Above system has taking after constraints : Query forms are outlined and pre-characterized by designers in information administration systems. Hard to plan an arrangement of static query forms to fulfill different database queries on complex databases. The above disservices are propelled to plan such DQF system which can explain the confinements.

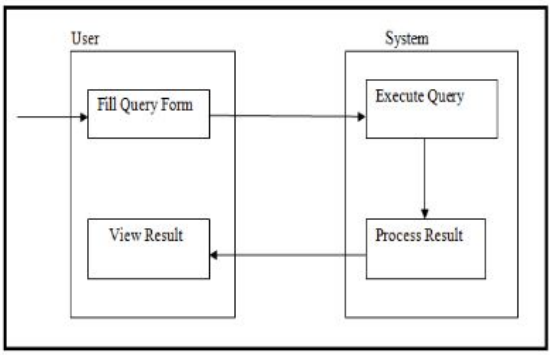

Fig. 1. Existing System

II. Review of Literature

S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, "Automated ranking of database query results," state that the results of the prelimary expriments by the different approches to enable ranking in databases.[1] M. Jayapandian and H. V. Jagadish, in their proposed "Automated creation of a forms based database query interface,"stated that the design of the form based query is the key step in the deployment of a database. They explain the various existing database management and development tools, such as EasyQuery, Cold Fusion, SAP and Microsoft Access,it provides several mechanisms to let users create customized queries on databases.[2] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, "Combining keyword search and forms for ad hoc querying of databases," provide the solution which combines keyword search with query form generation.[3] M. Jayapandian and H. V. Jagadish, "Automating the design and construction of query forms,"

proposed a workload driven method.It explain the tunable clustering algorithm for the simple query.[4] L. Tang, T. Li, Y. Jiang, Z. Chen, "Dynamic Query Forms for Database Queries," it provides a solution that nontechnical users can make usage of relational database which is the challenging task.Dynamic Query Form (DQF), a curious database query form interface which is able to tackle the large and complex relational databases. A system captures the users preference during the user communication and guides user to make decisions. Query form generation is a iterative process. The ranking of form components is based on the captured user preferences.[5] H.Li, Jaesoo Yoo,"An schema for continuous skyline Query Processing Over Dynamic Data set" Propose a multiple layer grids scheme for efficiently peocessing for continuous skyline Query processing.[6] Z. Dou, Z. Jiang, S.Hu, J.Wen, and R.Song,"Automatically Mining Facets for Queries from Their Search Results"states that the systematic solution for queries which is reffer as QDMiner which automatically mine the query facts by extracting and then groupung frequent list from free text , html tags within the top search results.[7] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia," Dynamic faceted search is a type of search engines where relevant facets are presented for the users according to their navigation paths. Dynamic faceted search engines are similar to our dynamic query forms if we only consider Selection components in a query.Selections, a database query form has other important components, such as Projection components.The projection components control the output of the query form and cannot be ignored. Moreover, designs of Selection and Projection have inherent influences to each other.[8]

III. Proposed System

Basic Query Form The user fills the present query form and presents a query. The basic query form incorporates not very many primary attributes of the database. The basic query form is then enhanced iteratively by trading between the user and system until the user is satisfied with the query result. Each query form relates to a SQL query design. A query form F is characterized as a tuple (AF, RF, F, (RF)), which speaks to database query design as takes after: F = (SELECT A1, A2,. . . . . .., AK

FROM (RF) WHERE F),

Where, AF = A1, A2,. . . . . .,Ak are k attributes for projection, k > 0. RF = R1, R2,. . . . . .,Rn is the arrangement of n relations (or elements) worried in this query, n > 0. Each ascribe in AF has a place with one connection in RF. F is a conjunction of expressions for selections on relations in RF. (RF) is a join capacity to make a conjunction of expressions for joining relations of RF. In the user interface of a query form F, AF is the arrangement of sections of the result table. F is the arrangement of info segments for users to fill. Query forms permit users to fill parameters to produce diverse questions. RF and (RF) are not noticeable in the user interface, which are typically created by the system as per the database outline. For a query form F, (RF) is consequently built by the foreign keys among relations in RF. In the interim, RF is controlled by AF and F. RF is the union arrangement of relations which contains no less than one trait of AF or F. Thus, the parts of query form F are really controlled by AF and F. As we said, just AF and F are unmistakable to the user in the user interface This module is planned for taking user input on the premise of result appeared for query execution. The user input will be as whether user is satisfied or not. This will help the user to change the query and get best sought results.

A. Application Example

This system can be used in many organizations.Share broking firm will be good example for this.As all share market investor invest their money through share broker and they want report on daily basis.So this system will help to broker to generate the report based on the investment of the user.In this case even if user don't have the SQL knowledge then also he can generate the report using DQF. System:

Fig 2 , starts with a basic query form which contains not very many primary attributes of the database. The users rounds out a present query form and submit it. Query is executed by the system and it likewise show a query result at a user side, and if user is not satisfied with the result then basic query form is advanced iteratively between the user and the system until the user is satisfied with the query results.

B. Query Generation:

Query generation module will give a simple approach to handle GUI for end users to create the SQL queries. It will help end users to recover information from the database without the knowledge of the SQL. The GUI will contain distinctive segments, for example, List of tables, arrangement for applying diverse conditions, arrangement for selecting at least one section names and so on. The query construction algorithm is utilized for producing a query which is helpful to show the information at user side. Query is produced by utilizing query construction algorithm. subsectionDisplay Query Result:

System displays a query result at user side.

C. Query Execution:

At first the user fills out the present query form and presents a query. After that DQF executes the query and show the results at a user side.

D. Query Enrichment:

Query enrichment module will have already executed query and it will permit the user to alter it for getting best results according to their needs. The query advancement is the procedure in which query gets altered and created on the premise of user feedback. This will be the tedious procedure till user gets the results according to their needs and gets satisfied. DQF additionally prescribes a positioned rundown of query form segments to the user.

E. Database:

A database is an aggregation of information that is sorted out with the goal that it can be straight forwardly gotten to, overseen, and updated. A database module comprises of two sub-modules i.e. Data Manipulation and Data Definition. A data manipulation language (DML) is a gathering of codings including commands that allow users to control data in a database. This manipulation incorporates inserting data into database tables, recuperating existing data, erasing data from existing tables and modifying existing data. The DML operation are SELECT, INSERT, UPDATE, DELETE. Data definition Language (DDL) is a standard for commands that depict the various structures in a database. DDL statements create, modify, and remove database queries, for example, tables, lists, and users. Customary DDL statements are CREATE, ALTER, DROP.

F. System Modules

Query Form Enhancement : In query form enhancement module DQF prescribes a positioned list of query form segments to the user. So that the user chooses the fancied form parts into the present query form.
Query Execution : Firstly the user rounds out current query form and present a query. At that point DQF executes the query more, shows results their databases, not for end-users. In proposed a system which permits end-users to modify the existing query form at run time. Be that as it may, an end-user may not be acquainted with the database. On the off chance that the database schema is extensive, it is troublesome for them to discover suitable database elements and credits and to make fancied query forms.
Database Query Recommendation : Late reviews present community oriented ways to deal with recomm end database query parts for database investigation. They regard SQL queries as things in the synergistic separating approach, and suggest comparative queries to related users.

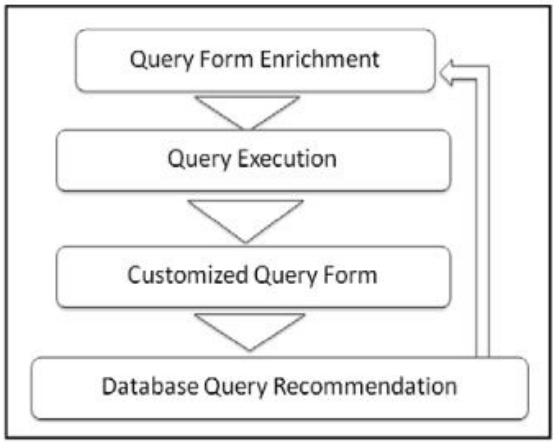

Fig. 2. System Modules

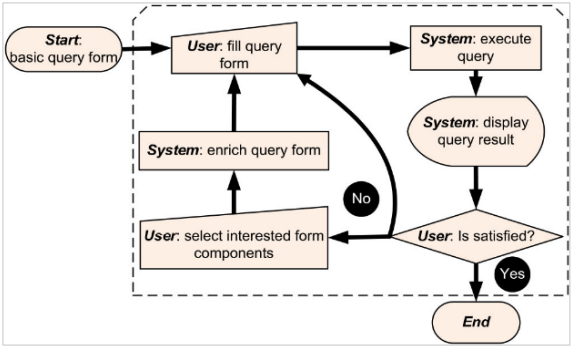

Fig. 3. System Flow

G. Static Vs. Dynamic Query Forms

On the off chance that a query task is secured by one historical queries ever, at that point SQF based on those historical queries can fulfill that query task. In any case, the costs of utilizing SQF and DQF to achieve that task are distinctive. Form-Complexity was proposed into assess the cost of utilizing a query form. On the introduce of fulfilling every one of client's queries, the complexities of query forms ought to be as little as could be expected under the circumstances. DQF creates one tweaked query form for each query. The normal form unpredictability is 8.1 for NBA, 4.5 for Green Car also, 6.7 for Geobase. Yet, for SQF, the multifaceted nature is 30 for NBA and 16 for Green Car (2 static query forms). This result demonstrates that, so as to fulfill different query tasks, the statically created query form must be more intricate.

IV. Contribution

The existing system generates query based on mathematical operater and only select statement , the proposed system consiste of insert,update,delete functionality.Also based of the user feedback the query form ranking is get generated which will help to user for generating same query based on existing form.The propose a dynamic query form system which creates the query forms as per the user's covet at run time. The system gives an answer for the query interface in substantial and complex databases. F-measure is applied to assess the decency of a query form . F-measure is a run of the mill metric to assess query results . This metric is additionally proper for query forms since query forms are planned to help users query the database. The decency of a query form is dictated by the query results created from the query form. In light of this, system rank and suggest the potential query form parts so that users can refine the query form effortlessly. In light of the proposed metric, efficient algorithms are created to evaluate the decency of the projection also, selection form parts. Here efficiency is critical on the grounds that DQF is an online system where users often expect brisk reaction.

V. Query Task

In the second phase, user finished 12 tasks (none of these tasks showed up in the workload) recorded previously. Every

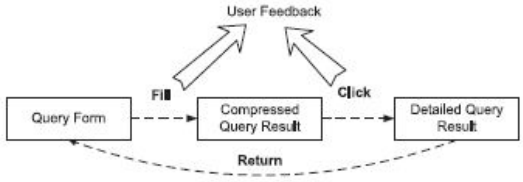

Fig. 4. System Flow

member utilized each of the three form generation approaches to form inquiries. The request of the three methodologies were randomized to remove predisposition. System set parameter = 0.001 in our examinations in light of the fact that our databases gather a specific measure of noteworthy inquiries so that we fundamentally consider the likelihood assessed from the noteworthy queries. This system develop the queries for above task dynamically.

Task	SQL	Meaning
T1	SELECT ilkid, firstname, lastname FROM players	Find all NBA players' ID and full names.
T2	SELECT p.ilkid, p.firstname, p.lastname FROM players p, player_playoffs_career c WHERE p.ilkid = c.ilkid AND c.minutes > 5000	Find players who have played more than 5000 minutes in the playoff.
T4	SELECT t.team, t.location, c.firstname, c.lastname, c.year FROM teams t, coaches c WHERE t.team=c.team AND t.location = 'Los Angeles'	Find the name of teams located in Los Angeles with their coaches.
T5	SELECT Models, Hwy_MPG FROM cars WHERE City_MPG > 20	Find the high way MPG of cars whose city road MPG is greater than 20.
T6	SELECT Models, Displ, Fuel FROM cars WHERE Sales_Area = 'CA'	Find the model, displacement and fuel of cars which is sold in California.
T7	SELECT Models, Displ FROM cars WHERE Veh_Class = 'SUV'	Find the displacement of all SUV cars.
T8	SELECT Models FROM cars WHERE Drive = '4WD'	Find all 4 wheel-driven cars.
T9	SELECT t0.population FROM city t0 WHERE t0.state = 'california'	Find all cities in California with the population of each city.
T10	SELECT t1.state, t0.area FROM state t0, border t1 WHERE t1.name = 'wisconsin' and t0.area > '80000' and t0.name = t1.name	Find the neighbor states of Wisconsin whose area is greater than 80,000 square miles.
T11	SELECT t0.name, t0.length FROM river t0 WHERE t0.state = 'illinois'	Find all rivers across Illinois with each river's length.
T12	SELECT t0.name, t0.elevation, t0.state FROM mountain t0 WHERE t0.elevation > '5000'	Find all mountains whose elevation is greater than 5000 meters and each mountain's state.

Fig. 5. Query Task

VI. Effectiveness

System contrast the ranking function of DQF and two other ranking techniques: the gauge strategy and the irregular technique. The gauge strategy ranks projection and selection attributes in rising request of their mapping separation to the present query form. For the query condition, it picks the most frequent utilized condition in the preparation set for that trait. The irregular strategy haphazardly recommends one query form segment. The ground truth of the query form part ranking is acquired from the query workloads and expressed. Ranking projection parts: Ranking score is a regulated strategy to measure the exactness of the suggestion. It is gotten by contrasting the figured ranking and the ideal ranking. In the ideal ranking, the real selected part by the user is ranked first. So ranking score assesses how far away the real selected part is ranked from the first.

VII. Efficiency

The run-time cost of ranking projection and selection segments for DQF relies on upon the present form segments, the query result size. Accordingly we selected 4 complex queries with huge result size for every data set. Tables 10, 11, and 12 list these questions, where those join conditions are verifiable inner joins and written in WHERE condition. We shifted the query result size by query paging in MySQL motor. The running times of ranking projection are all less than 1 millisecond, since DQF just figures the pattern separate and contingent probabilities of attributes. It demonstrates the time for DQF to rank selection parts for inquiries on the data sets. The results demonstrate that the execution time develops roughly straightly as for the query result size. The execution time is between 1 to 3 seconds for NBA when the results contain 10000 records, under 0.11 second for Green Car when the results contain 2000 records, and under 0.5 second at the point when results contain 10000 records. So DQF can be utilized as a part of an intuitive domain.

VIII. Algorithm

Query Construction for Select Query.

1. Select attributes from tablename
2. String s = "select"
3. String[] column
4. For(i to column. length; i++)
5. column [i] = column
6. String column = join(column, " ")
7. String tablename
8.check column and conditional operator
9.if true
10. String query = s + column + tablename+column +conditional operator
11.if false
12.String query = s + column + tablename

IX. Mathematical Model

Problem statement comes under the polynomial class ; the problem is solved in P-time. So above two deterministic algorithms called P-class algorithms.

A. Set Theory

Set: S={I, R, P, O}

Where, I= Set of Inputs for our system

R= Set of Rules that are applied while processes are performed.

P= Set of Processes

O= Set of Outputs

I={I1, I2, I3, I4 }

Where,

I1:User Details

I2: Select Databases

I3 : Select Query

R={R1, R2, R3}

Where,

R1= Database Selection

R2= Query Generation

P={P1, P2, P3 }

Where, P1= All Databases

P2= Ranking of Results

P3=Query Formation
O={O1, O2, O3}
Where, O1: Data accessing properly
O2: Top Search Find Out
O3: Ranking of Data

X. Result And Discussion

Id	Form Name	Better	Average	Good	Bad	Fscore	Precision	Recall
220	Form221	1	null	null	null	0.019656615426436	0.0262588235294118	0.86589
221	Form222	1	null	null	null	0.00391705237947555	0.00522471910112359	0.86589
222	Form223	1	null	null	null	0.0300742863167724	0.0402162162162162	0.86589
223	Form224	1	null	null	null	0.64749811436481	0.93	0.86589
224	Form225	1	null	null	null	0.0575017276924791	0.0770984455958549	0.86589
225	Form226	1	null	null	null	0.042589637502869	0.057055214723926 4	0.86589
226	Form227	1	null	null	null	0.0440965322878188	0.0590476190476 19	0.86589
227	Form228	1	null	null	null	0.192295857959035	0.261971830985916	0.86589
228	Form229	1	null	null	null	0.0107593687561539	0.0143629343629344	0.86589
229	Form230	1	null	null	null	0.017264465322719	0.0230578512396694	0.86589
230	Form231	1	null	null	null	0.64749811436481	0.93	0.86589

Fig. 6. Precision and Recall

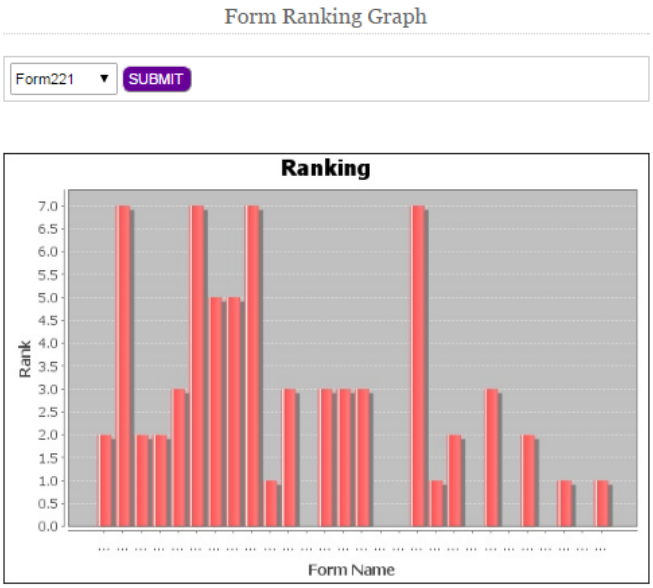

Fig. 7. Ranking Graph

precesion=Total number of relevant record retrieved / total number of records
recall=total number of irrelevant record retrieved / total number of record
fscore=(2*precesion*recall)
In result analysis the F-score is figured with the assistance of accuracy and review. F-score is utilized to quantify the integrity of query forms. Likewise the feedbacks backup is store with the goal that come to realize what feedback is given by the user to a specific form. Additionally by tapping on the form name appeared in the demonstrates that one can alter form again and furthermore see the result at whatever point required.

XI. Conclusion

This system express that a DQF i.e., Dynamic Query Form generation approach which help users progressively produce query forms. The system is planned to create different sorts of questions on the premise of user feedback with cycles of the query advancement prepare. A user can likewise fill the query form and submit inquiries to analysis the query result at every emphasis. The dynamic query form generation system focuses on giving user friendly GUI, efficient and fast query generation, and query improvement.

Acknowledgment

All the work dedicated to esteemed guide Prof.A.M.Jagtap,whose interest and guidance help to complete the work successfully.The experience will always steer to do the work perfectly and professionally.Also extend the gratitude to H.O.D.(Information Technology) who has provided facilities to explore the subject with more enthusiasm. This work was supported by the teachers and staff of the Department of Information Technology.

References

[1] M. Jayapandian and H. V. Jagadish, “Automated Creation of a Forms -Based Database Query Interface", in *IEEE Transatcion On VLDB*, vol. 1, no. 1, pp. 695709, 2008.
[2] M. Jayapandian and H. V. Jagadish, “Automating the Design and Construction of Query Forms", in *IEEE Trans. On Know. Data Engg.*, vol. 21, no. 10, 10, pp. 13891402, Oct. 2009.
[3] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, “Combining Keyword Search and Forms for Ad Hoc Querying of Databases”, in *Proc. ACM SIGMOD, Providence, RI, USA"*, vol. 26, no. 9, pp. 349360, Jun. 2009.
[4] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, “Automated Ranking of Database Query Results”, in *in Proc.CIDR*, pp. 888899, 2003
[5] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, “Query Recommendations for Interactive Database Exploration”, in *Proc. SSDBM, New Orleans, LA, USA*, pp. 318, Jun. 2009.
[6] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, “Facetedpedia: Dynamic Generation of Query-Dependent Faceted Interfaces for Wikipedia”, in *Proc. WWW, Raleigh, NC, USA*, pp. 651660, Apr. 2010.
[7] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, “Dynamic Query Forms for Database Queries”, in *IEEE Trans. On Know. Data Engg.*,vol:PP no:9 Sept. 2014.
[8] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu “A Framework for Clustering Evolving Data Streams", in *Proceedings of VLDB, Berlin, Germany*,pages 81–92, Sept. 2003.
[9] M. Jayapandian and H. V. Jagadish, “Automating the design and construction of query forms”, in *IEEE TKDE*, vol. 62, no. 2, 21(10):1389–1402, Feb. 2009.
[10] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, “Combining Keyword Search and Forms forAd Hoc Querying of Databases", in *Proceedings of ACM SIGMOD Conference, Providence, Rhode Island, USA*, pages 349 360, June 2009.
[11] Zhicheng Dou, Member, IEEE, Zhengbao Jiang, Sha Hu, Ji-Rong Wen, and Ruihua Song, “Automatically Mining Facets for Queries from Their Search Results", in *IEEE Trans. On Know. Data Engg.*, vol. 28, no. 2, Feb. 2016.