Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 7, July 2014, pg.420 – 424

RESEARCH ARTICLE

Dynamic Query Generator

Miss. B.Rawali, Miss. S.Geetha Sree, Mr. I.S.Raghuram, Mr. N.Srikar Setty
Department of Information Technology, GPCET, Kurnool

Abstract: In Modern Database and Web Based Database it will maintain large and heterogeneous data. To fetch the data from the database we are going to use this Query Form. Query Form is the one of the most widely used user interface for querying databases. These Query Forms are defined and designed by the Developers or Data Base Administrators in different Management Systems. With the Rapid increment of the internet and scientific databases and modern databases these all become very large and complex. Recognizing the drawbacks of website based approaches, we address the question of how to improve the fetching the results from query results. Traditional predefined query forms are not able to satisfy various ad-hoc queries from users on those databases. Here, we propose DQF, a novel database query form interface, which is able to dynamically generate query forms. The essence of DQF is to capture a user's preference and rank query form components, assisting him/her to make decisions.

Introduction to Query Form:

The query forms are classified based upon their database query form or information retrieval languages. Query form is a property of object oriented language. Query form can be defined as a form for the specification of procedures for the retrieval information from a database. Here the set of commands are used to retrieve the information in database.

Select Statement:

The most commonly used SQL command is SELECT statement. The SQL SELECT statement is used to query or retrieve data from a table in the database. A query may retrieve information from specified columns or from all of the columns in the table. To create a simple SQL SELECT Statement, you must specify the column(s) name and the table name. The whole query is called SQL SELECT Statement

Syntax:

SELECT <*column list*> FROM <*table name*> [WHERE Clause] [GROUP BY clause] [HAVING clause] [ORDER BY clause];

SQL Joins: These are used to combine the rows from two or more tables. Here we have different types of joins

- Inner Join
- Left Join

- Right Join
- Full Join

Inner Join: The Inner Join keyword selects all rows from both the tables as long as there is a match between the columns in the both table.

Syntax:
SELECT column _name FROM table1 INNER JOIN table2 ON table1.column_name=table2. column_name;
Here, the both Join and Inner Join are same.

Left Join: The Left Join keyword returns the all rows from left table with the matching rows in the right table. If there is no match then the result will be NULL.

Syntax:
SELECT column_name FROM table1 LEFT JOIN table2 ON table1.column_name = table2. column_ name;

Right Join: The Right Join keyword returns all rows from the right table with the matching rows in left table. If there is no match then the result will be NULL.

Syntax:
SELECT column_name FROM table1 RIGHT OUTER JOIN table2 ON table1.column_name=table2.column_name;

Full Outer Join: The Full Outer Join keyword returns the all rows from both left table and right table. It combines the result of both the LEFT and RIGHT Joins.

Syntax:
SELECT column_name FROM table1 table2 ON FULL OUTER JOIN table1.column_name=table2.column_name;

Many Databases such as natural database and scientific database, web structured database, typically have thousands or hundreds of entities. Therefore, it is very difficult to search the data by using these query forms. If we have relevant information also by using query forms it can't satisfy the various ad-hoc database quires.

Problem Definition

A Primary cause for user is not familiar with the fetching data from database schema; those data will be confusing them. Aforementioned approaches are 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.

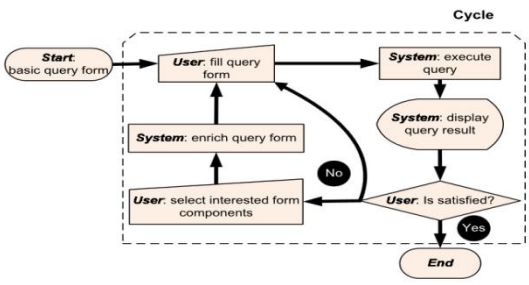
Another problem is that, when we generate a large number of query forms, how to let users find an appropriate and desired query form would be challenging. A solution that combines keyword search with query form generation is proposed. It automatically generates a lot of query forms in advance. How to let Non-Expert users make use of relational database is challenging topic.

Dynamic Query Generator:

Here in the modern scientific databases and web databases maintain large and heterogeneous data. In the traditional databases are not able to satisfy the various ad-hoc quires from users on those databases.

We propose a Dynamic Query Generator system: DQG, 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 perform many rounds of actions (i.e., refining query conditions) before identifying the final candidates.

System Architecture Diagram



The essence of DQG is to capture user interests during user interactions and to adapt the query form iteratively. Each iteration consists of two types of user interactions: Query Form Enrichment and Query Execution. In the above architecture it starts with a basic query form which contains very few primary attributes of the database. The basic query form is then enriched iteratively via the interactions between the user and our system until the user is satisfied with the query results.

Here in the Dynamic Query Generator it is a novel database query form interface, which able to dynamically generate the query forms. At each Iteration it will capture the user preferences and rank the query form interface, which assisting the user to make decisions here, user can also fill the query form and submit the query so that we can view the query results at each iteration. At Iteration it will generates the ranking list of form components and then user may add desire components into the query form.

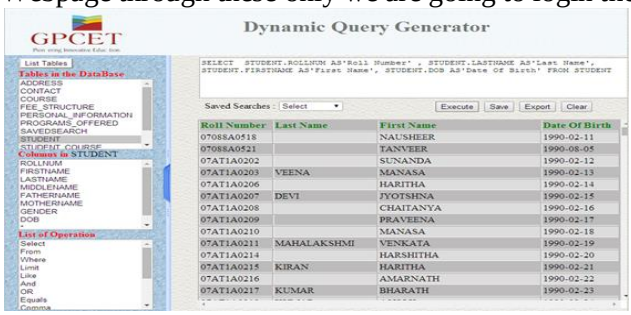
User Study:

Here, we are going to recruit some big amount of data like some 100000 participates of engineering students. Here, in that consists of all the student details like their branch, email-id's, their residence address etc,. Here, it consists of 2 phases one is query collection phase and another one is testing phase.

Here, in the collection phase, each participant is going to submit the quires. In the testing phase each participant going to select the components. Here in the DQG here the database will going to collect some historic quires based on the historic quires we are going to rank the query forms.

Here action means a mouse click or keyboard input for a text box it will carry minimal number of actions for query task. Here, in the webpage based system we needed some UI components so the user can easily generate the all possible query forms in single web page, the user need to choose one query form to finish his query task.

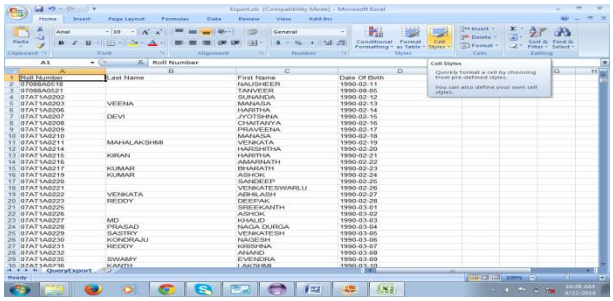
The DQG Computes the ranked list the query form components at each iteration. In those ranked list users will found desired entities and attributes at the top the ranked list. Here, these is Our Dynamic Query Generator Webpage through these only we are going to login the system



Here, after login in to the system we are going to perform the query task here, we just click on those UI components and then we can perform query task

Roll Number	Last Name	First Name	Date Of Birth
07088A0518		NAUSHEER	1990-02-11
07088A0521		TANVEER	1990-08-05
07AT1A0202		SUNANDA	1990-02-12
07AT1A0203	VEENA	MANASA	1990-02-13
07AT1A0206		HARITHA	1990-02-14
07AT1A0207	DEVI	JYOTSHNA	1990-02-15
07AT1A0208		CHAITANYA	1990-02-16
07AT1A0209		PRAVEENA	1990-02-17
07AT1A0210		MANASA	1990-02-18
07AT1A0211	MAHALAKSHMI	VENKATA	1990-02-19
07AT1A0214		HARSHITHA	1990-02-20
07AT1A0215	KIRAN	HARITHA	1990-02-21
07AT1A0216		AMARNATH	1990-02-22
07AT1A0217	KUMAR	BHARATH	1990-02-23

After executing the query, we are getting some ranked list of entities and attributes from these lists of components user is going to select his desired entities and attributes.



Conclusion and Future Work:

In Dynamic Query Generator for Database Queries we propose a dynamic query generator generation approach which helps users dynamically generate query forms. We capture user preference using both historical queries and run-time feedback such as click through. The ranking of form components also makes it easier for users to customize query forms. Here, we only related for relational data, where as in future work it can be extends to non-relational data. And here we are going to develop multiple methods to capture the user preferences.

References:

[1] Cold Fusion. http://www.adobe.com/products/coldfusion/.

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

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

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

[5] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of VLDB*, pages 81–92, Berlin, Germany, September 2003.

[6] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of WSDM*, pages 5–14, Barcelona, Spain, February 2009.

[7] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In *CIDR*, 2003.

[8] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das. Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia. In *Proceedings of WWW*, pages 651–660, Raleigh, North Carolina, USA, April 2010

[9] 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.

[10] 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.

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

[1 2] T. Joachims and F. Radlinski. Search engines that learn from implicit feedback. *IEEE Computer (COMPUTER)*, 40(8):34–40, 2007.

[13] N. Khoussainova, M. Balazinska, W. Gatterbauer, Y. Kwon, and D. Suciu. A case for a collaborative query management system. In *Proceedings of CIDR*, Asilomar, CA, USA, January 2009.

Authors:

Miss. B.Rawali received his B.Tech degree in Information Technology from JNTU Anatapur (A.P), India. He studied in GPCET, Kurnool.

Miss. S.Geetha Sree received his B.Tech degree in Information Technology from JNTU Anatapur (A.P), India. He studied in GPCET, Kurnool.

Mr. I.S.Raghuram received his M.Tech degree in Computer Science and Engineering from JNTU Hyderabad (A.P), India. He was a Lecture, Assistant Professor with Department of CSE, GPCET, Kurnool. His research interests include Data Mining, and Ad-hoc Networks

Mr. N.Srikar Setty received his B.Tech degree in Information Technology from JNTU Anatapur (A.P), India. He studied in GPCET, Kurnool.

