ISSN 2277-3061

AN AMELIORATED METHODOLOGY FOR RANKING THE TUPLE

Ajeet A. Chikkamannur, Shivanand M. Handigund
Department of Information Science and Engg, New Horizon College of Engineering, Bangalore 562157, India
ac.ajeet@gmail.com
Department of Computer Science & Engineering, Bangalore Institute of Technology, Bangalore 560004, India
smhandigund@gmail.com

ABSTRACT

Many users searching databases through the web in various domains like vehicles, real estate, etc. One of the predicament, we observed in this task is ranking the results retrieved from a database for a user query. The contemporary methods are addressing this problem by sorting the database values. A common line in these methods is that ranking is done in a user and query-independent manner i.e. there is no correspondence between different users' queries.

We proposed a query and user-dependent approach for ranking query results in relational databases, which is articulating the opinion expressed in query by different users. The designed and developed methodology shows that the ranking of the data in database is constituted depending on the various user opinions i.e. access of tuple through users query expression. The design is based on counting the access of database tuple by users' query(ies)

Indexing terms/Keywords

Rank; query; count; database; tuple; relation

Academic Discipline And Sub-Disciplines

Engineering, Computer Science and Engineering, Information Technology;

SUBJECT CLASSIFICATION

Relational Database Management Systems

TYPE (METHOD/APPROACH)

Innovative Design and Deployment

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 14 . No. 4
www.ijctonline.com , editorijctonline@gmail.com

ISSN 2277-3061

1.0 INTRODUCTION

At pragmatic scenario, most of the applications in computer system are designed and developed with a support of database. The supported database is continually queried and accessed for the information by various users in public domain. The data existing in the database is fundamental and helpful in a decision making with flexibility that the user can pose number of queries for data retrieval. Varieties of users are posing the queries on the same database depending on their desire, will and wish. The challenge observed here is that how to assure the decisive necessitate of various users?

Current tools and techniques, after searching the database yield the large number of results to a picky search. Among those results the user has to select the exact wants and get the related answer for the study, which may be presumed as time consuming activity and there is a scope for incorrect separation. The present web databases simplified this task by querying on the single attribute value search but the user needs the values in terms of multiple values, those are related closely to each other.

This paper proposes an approach for the integration of the user queries depending on the access of query attribute value(s) on a relation. Then the priority of the tuple is determined depending on the number of accesses to the data of database. The any i.e. top, bottom, between k priorities tuples are retrieved from the relation.

2.0 BACKGORUND

This section discusses the current methodologies of the searching a database of an application. Generally the database is searched by formulating and calculating the query condition depending on the schema attribute(s) value of the relation. These queries are formulated by the application of Structured Query Language. These queries are blended in the web database search engine for simplifying the task by displaying query results sorted on the value of a single attribute or unrelated attribute values. But, most of the time, users would prefer to have ordering on multiple attribute values. This is because; some time the single value of an attribute will not support the expected decision of a user.

The current SQL semiotic permits the specification of attribute weights [1] [2] for the purpose of retrieval (specified in WHERE clause). This approach is burdensome in multiple user systems since different users' specification of weightage on attribute is divergent. The query-dependent ranking techniques [3, 4] provide a single ranking for a user query depending on the condition specified. The user profiles [5] as well as requiring users to order data tuples [6], proposed for user-dependent ranking, do not distinguish between queries and provide a single ranking order for any query given by the same user. The users and their required/desired items similarity extraction techniques are either in collaborative [7, 8, 9] or content filtering [10, 11]. These existing contributions are efficient and useful in the data manipulation computation point of view.

Here, we have attempted to design and develop a technique for ranking a tuple of relational database from the data definition and manipulation structured query language point of view.

3.0 FRAMEWORK

Searching for requirement of user in various domains through the web is common practice. In the process of searching, the problem we observed is the reach-ability of result(s) to users yielding from the query. Most of the methods provide the results by sorting the result according to the particular user perspective i.e. user independent manner. Further, the user needs to formulate the various queries on a schema attributes depending on the requirement and need. The results returned from the execution of these queries are large in size. In such case, the investigation of useful result is time consuming and miscalculation.

The current methodologies are simplifying this task by sorting the result on single attribute. However, the web users will prefer the result inferred from multiple attributes, which are nearer to their expectation. Further, multiple users are posing the variety of the queries on the relation to reveal their interest of data existing in the database. Here, *we deduce that the integration of multiple users' insight strongly support the ranking of the value.* Hence, this work integrates the multiple users' queries expression automatically for the ranking of a tuple.

The grouping of attributes for a schema on the good database design principles is available in [12, 13, 14, 15]. These databases are utilized in the web application for the back end support. Further, the scratch work of this paper is proposed by the authors in [16]. In web browsing the user specifies the data for searching from a database. On this attempt, the user may get the relevant result depending on the availability of data in a database. In existing system, on retrieval of the result, the user should give the opinion about liking of a particular data. Instead here, we have derived the opinion from the value(s) existing in a query and database. The "equal" value(s) among the two objects i.e., value of WHERE clause of query and data existing in database is considered as the interest of the user. On satisfying this criterion the appropriate data is updated to the special attribute that is appended to the relation.

The mathematical model for the opinion integration of many users and extraction of top ranking is depicted below;

$$\textbf{Rank = count (} \sum_{i=1}^{n} access(t))$$

Where the n is the number of users, t is the value(s), which is accessed by the users to reveal their interest or opinion. The count is function which will count the number of access to the existing value.

The explanation of the step by step procedure is as follows:

Step 1. The relation R of which the tuples are to be ranked is appended with the attribute. This attribute is termed as "count"

Step 2. The NULL value is inserted in the count attribute corresponding to each tuple or row.

Step 3. Extract the tuple value(s) specified in the WHERE clause of SQL query applied on the relation R.

Step 4. If the value(s) found in the tuple of Relation R then the count attribute value is incremented by 1 corresponding to that tuple

Step 5. The rows are sorted on count attribute values depending on the ascending order.

The algorithm for the designed procedure is given below;

Input: Queries to a Relation

Output: Sorted tuples with top-k rank

R: relation

A: attribute

Q_i: i^{th} query

T: tuple

V_i: value(s) of tuple specified in Q_i

1. Append an attribute A_{count} to a relation R

   $R \leftarrow R \cup A_{count}$
2. $A_{count} \leftarrow 0$
3. Read the corresponding tuple specified in the WHERE clause from the relation R i.e. T_W
4. if ($V_i == T_w$) then

   A_{count} value$\leftarrow$ A_{count} value + 1

   end if
5. $R \leftarrow$ Sort (A_{count})
6. Display R

4.0 CASE STUDY

To illustrate the ranking methodology, we have sifted the most commonly known "Bike-Domain" among the various domains. This domain is characterized by the various and numerous domain attributes but we have considered the attributes as name, price, model, cc, and engine for the illustration purpose. The relational database designed from these attributes is depicted in the figure1.

NAME	PRICE	MODEL	CC	ENGINE
Honda	20000	2014	800	petrol
Maruti	25000	2013	800	diesel

Figure 1. Relation of Bike domain

The new attribute "count" is appended to this relation. Then the attribute "count" content is maintained NULL initially. The resulting table is shown in the figure 2.

NAME	PRICE	MODEL	CC	ENGINE	COUNT
Honda	20000	2014	800	petrol	NULL
Maruti	25000	2013	800	diesel	NULL

Figure 2.Attribute appended to relation

With five customers posing their queries on this table. The first customer is interested in the brand "maruti" then the row two is accessed, so increments the count value. The second customer is interested in 2014 model and the query is accessing the row one value of 2014. On this access the count value of row one is incremented.

The third customer interested on petrol engine. So on access of this value the row one count value is incremented. Similarly, the fourth customer's interest is on 800cc, on accessing the value row one's as well as row two's count value is incremented. The fifth customer is interested on the price of 20000. Hence the row one count value is incremented. The resulting values of count are shown in figure 3.

NAME	PRICE	MODEL	CC	ENGINE	COUNT
Honda	20000	2014	800	petrol	04
Maruti	25000	2013	800	diesel	02

Figure 3 Updating of count value

Among the rows, the row one of relation have the count value as 4 and row two have the count value as 2. Since the row one have the count value of 4, the row one is top ranked after sorting the count values in descending order.

5.0 CONCLUSION

In our proposed methodology, the additional attribute is appended to a designed relation for the storage of count value pertaining to the access of a tuple value. The count value is ameliorated depending on the "exact match" value between the tuple and the value specified in the query. The exact match among the values and updation of count attribute value is carried out by the "trigger concept", which is supported by the all vendors of relation database products. The work is focused on the select operation of the relational algebraic expression, which is utilized

The data mining techniques provide the data on the interest of particular user. The interest of user is strengthened by the support of many other users' opinion for strong decision. This is automatically carried out in our methodology. Since the ranking of tuple is made on the count value, the users have the flexibility of opting the desired range i.e. top, bottom or middle.

In future, this work is to be extended for the provisioning of tuple to the authenticated and authorized users.

ACKNOWLEDGMENT

We are thankful to the organizer of the National Conference on Recent Advances in Science, Engineering and Technology (NCERSAT-14), held at Sri Venkateshwara College of Engineering Bangalore for presenting this paper

REFERENCES

[1] Li C., Chang K. C, Ilyas I. F., Song S. "Ranksql: Query algebra and optimization for relational top-k queries". SIGMOD Conference, pp 131–142, 2005

[2] Marian A, Bruno N, Gravano L. "Evaluating top-k queries over web-accessible databases". ACM Transactions of Database Systems, 29(2) pp 319–362, 2004

[3] Su W, Wang, Huang, Lochovsky L, "Query result ranking over e-commerce web databases", Conference on Information and Knowledge Management (CIKM), pp 575–584, 2006

[4] Yu, Kim, Won Hwang. "Rv-svm: An efficient method for learning ranking svm", PAKDD, pp 426–438, 2009

[5] Koutrika, Ioannidis, "Constrained optimalities in query personalization. In SIGMOD Conference, pp 73–84, 2005.

[6] Hwang, "Supporting Ranking for Data Retrieval" PhD thesis, University of Illinois, Urbana Champaign, 2005.

[7] Hofmann, "Collaborative filtering via Gaussian probabilistic latent semantic analysis", SIGIR, pp 259–266, 2003.

[8] Basilico, Hofmann. "A joint framework for collaborative and content filtering", SIGIR, pp 550–551, 2004

[9] Billsus, Pazzani. Learning collaborative information filters, International Conference on Machine Learning (ICML), pp 46–54, 1998.

[10] Balabanovic, Shoham. "Content-based collaborative recommendation", ACM Communications, 40(3):66–72, 1997.

[11] Gauch et. al., "User profiles for personalized information access" Adaptive Web, pp 54–89, 2007

[12] Handigund S. M., "Reverse Engineering of COBOL Legacy Systems", Ph. D., thesis Indian Institute of Technology Bombay, 2001

[13] Chikkamannur A. A, "Design of Fourth Generation Language with the blend of Structured Query Language and Japanese Basic English", Ph. D. thesis Visvesvaraya Technological University Belgaum, India, 2013

[14] Chikkamannur A. A., Handigund S. M., "An ameliorated methodology to design normalized relations" ACS /IEEE sponsored, 7th international Conference on Computer Systems and Applications (AICCSA09), Rabat, Morocco, pp 861-864, 2009.

[15] Chikkamannur A. A., Handigund S. M. "Design of Normalized Relation: An Ameliorated Tool", International Journal of Computing and Information Sciences (IJCIS), ISSN 1708-0460, Vol 9, No. 1, pp-28-30, 2011

[16] Suresh M., Ajeet Chikkamannur, Yogesh Kumar, “Ranking for Web Database: An Approach”, National Conference on Recent Advances in Science, Engineering and Technology (NCERSAT-14), Sri Venkateshwara College of Engineering Bangalore,

Biography

Prof. Ajeet A. Chikkamannur received his Ph. D. degree in Computer & Information Science and Engineering in 2013 from the Visvesvaraya Technological University, India. on thesis titled “Design of Fourth Generation Languages with the blend of SQL and Japanese Basic English”. Currently working as Professor & Head, Department of Information Science in New Horizon College of Engineering Bangalore. His research interests are Object Oriented System Development, Database Management Systems, System Simulation and Modeling, Cloud Computing, Bigdata. He teaches under graduate and graduate. Published number of papers in several national, international conferences and journals.

Prof. Shivanand M. Handigund received his Ph.D. degree from Department of Computer Science & Engineering, Indian Institute of Technology, Bombay in 2001. Currently, working as a full time Professor and Dean, Research and Development, Bangalore Institute of Technology, Bangalore. His research interests are Software Engineering, Reverse Engineering, Database Management Systems, Object Technology and Computer Graphics. He teaches several courses to Academia and Industry Engineers. He has organized number of conferences and delivered keynote addresses & invited talks at several conferences. He is a Ph. D. referee and IEEE technical papers reviewer.