Implementing Abstract Information Model for Presenting Database Query Results

M.Sumathi
PG Scholar,
Department of Computer Science and Engineering
SNS College of Technology,
Coimbatore-641035,
Tamilnadu, India.

Prof.T.Kalaikumaran
Professor
Department of Computer Science and Engineering
SNS College of Technology,
Coimbatore-641035,
Tamilnadu, India.

Dr.S.Karthik
Dean & Head
Department of Computer Science and Engineering
SNS College of Technology,
Coimbatore-641035,
Tamilnadu, India.

Abstract

The exponential growth in Internet technologies has resulted in the usage of data stored in databases. But still, the database search techniques need more improvement to make it user friendly and efficient in fulfilling user's search needs. Even though the full text search engines are developed using the database search techniques as a base, the full text search engines has acquired good response from the users because of its simplicity and the efficiency of search engines to list the related results in the first page. But in the case of database search, there are database results, which cannot be ranked by usual sorting technique used in database engines. This results in information overload. Abstract Information Model helps to overcome this problem by displaying the abstract information of the data and guides to find the result set by refining the search in minimum steps.

Keywords - *data mining; database information retrieval; ranking query results; machine learning; decision support system*

1. Introduction

The web having huge repository of documents, attracts million of users globally to search for the information needed by them using simple keywords. With the advancements in the internet technologies, usage of web has increased exponentially. And now more data that are stored in databases are also available in the web. Normally users search these databases to find some useful information based on the domain knowledge and so they know what they want. But they don't know the content of the databases. And it is normal human tendency to compare between information that is available. So to understand the actual contents, the user starts exploring the details with broad queries. This will result in more tuples from which the user has to compare and find the small set of tuples they need. For some queries the results can be ordered using sorting technique. But there are many results which cannot be ordered using such techniques and the user has to view the whole result set to understand the contents. This is called information overload.

If the results are limited to few pages, the user can browse through the results page by page and can easily comprehend the result. But when the result span more pages, the user cannot get an overall view of the information available in the result set. In document information retrieval system, the resultant documents list is huge and most of them are irrelevant also. So many techniques are developed to rank the documents so that most relevant document will be displayed in the first page. As opposed to document search, all the tuples fetched by the query from a database based on the user constraint are not just relevant to the search but they are all the correct answers to the query. And when no ranking or sorting techniques can be used to order the result set, it requires some method to display the results usefully and efficiently. Access to all the fetched data should be provided to the users and also the total dataset should be displayed in one screen so that the user can get the overall view of the fetched data.

By seeing the total result set, all the users cannot get the actual information available in them. Also it is not easy for even an expert to extract the actual information hidden in the dataset without doing some processing. So the big challenge is to provide minimum processed contents that give an overall view of the total result set and by exploring the contents the user should be directed towards their purpose of search. For that the Abstract Information Model (AIM) was described in [11], which will display only the abstract information of the total result set instead of displaying the actual result set. And in this AIM, some processing techniques are suggested to present the hidden information to the users and also to provide a means to guide the users to see the unprocessed data.

The AIM requires the Field Meta Information (FMI) and Preference Meta Information (PMI) to display the contents. This paper describes the implementation procedure for the suggested Abstract Information Model using a Used Car Database.

2. Related works

[11] Illustrates an Abstract Information Model for presenting the database query results to reduce the information overload. In this initially the abstract information will be displayed to help the user to understand the content of the database and it guides the user to get the expected result in minimum number of steps by using query refinement technique.

In [2], the authors have presented an approach to build a generic automated ranking infrastructure for SQL databases with the concept that it will be desirable to have the option of ordering the matches automatically that ranks more “globally important” answer tuples higher and returning only the best. For that they have proposed a technique to extend TF-IDF based techniques from information retrieval to numerical and mixed data. In the vehicles database example, the globally important answer will be to have minimum price and minimum mileage. But this combination may occur because of some problem with the vehicle also. So the results cannot be ranked only with globally important concept.

In [3], query results are represented with perceptual glyphs positioned along a space-filling spiral, with colour and texture properties used to encode the element’s attribute values. Animations are used to highlight similarities and differences between pairs of query results. They have used queries from a movie recommender system for implementing that. This system is really good to present a large number of results in a single spiral. [5] proposes a completely automated approach for the information overload Problem which leverages data and workload statistics and correlations using ranking functions that are based upon the probabilistic IR models, judiciously adapted for structured data.

[6] addresses the problem of selecting the top ‘m’ attributes from the view point of helping a user understand what factors most influenced a ranking system in its ranking decisions. It presented several variants of the problem, showed that several of these variants are NP-hard, and presented efficient greedy heuristics and performed a user study demonstrating the benefits of a hybrid approach that returns the top attributes from each of these variants.

In [7], the authors have proposed techniques to automatically categorize the results of SQL queries on a relational database in order to reduce information overload. They have proposed a method that dynamically generate a labelled, hierarchical category structure - the user can determine whether a category is relevant or not by examining simply its label and explore only the relevant categories, thereby reducing information overload. And they have developed algorithms to generate the tree that minimizes information overload. This enables restriction of results based on the user selection.

[10] presents a representative model for displaying only a small set of tuples which are actually the representatives of the large number of tuples attached to it. For finding the representatives they have used average k-medoid clustering technique. But this technique is applied to a pre-formed cluster using cover-tree data structure proposed by [4]. The cover-tree structure helps to form clusters from the large data set based on Euclidean distance. From the cover-tree clusters, k-medoid clustering technique is applied to find the medoid using distance cost. In [12], the authors propose a novel categorization approach which takes advantages of the user contextual preferences to construct a navigational tree in order to reduce the information overload. In [8], the author suggests an efficient way of presenting database query results

through audio user interfaces. [9] Analyses the possible ways of presenting relational database query results.

In [1] also, the authors emphasis on the importance of user preferences and have proposed a method for finding different user preferences based on the browse history.

3. Implementation

As described in [11], the implementation of the abstract information model for the sample Used Cars Database is described in the Fig 1. In this block diagram, first the block, *1.Make,* displays all the available Make of cars. When the user selects any one from the available list, next block, *2.Car Type,* is displayed. Based on the user's selection for the second block, the third block, *3.Model,* is displayed. Based on the user's selection for the first three blocks, the information to be displayed in the column, row and within the cells contents are chosen using Preference meta information available in the Table 1. For displaying the list of choice for the blocks 4 to 6, the field meta information available in the Table 2 is used.

By using the user preference and FMI & PMI, the abstract information model is displayed in figures 2 to 6. These figures display the AIM for the selected preferences of Make, Type and Model.

As explained in [11], the field meta information is used to determine the field type as numeric or string. If it is a string field like *Colour,* grouping is done statically. But if the field is numeric like *Price* or *Mileage* then based on the number of range given by the user, the available value is split into steps dynamically at run time.

In Figure 2, the Price value is split into 8 groups based on the selected Price range and displayed in rows. And the Mileage value is split into 5 groups and displayed in columns. The cell contents are filled with minimum year of cars available for that Price and Mileage.

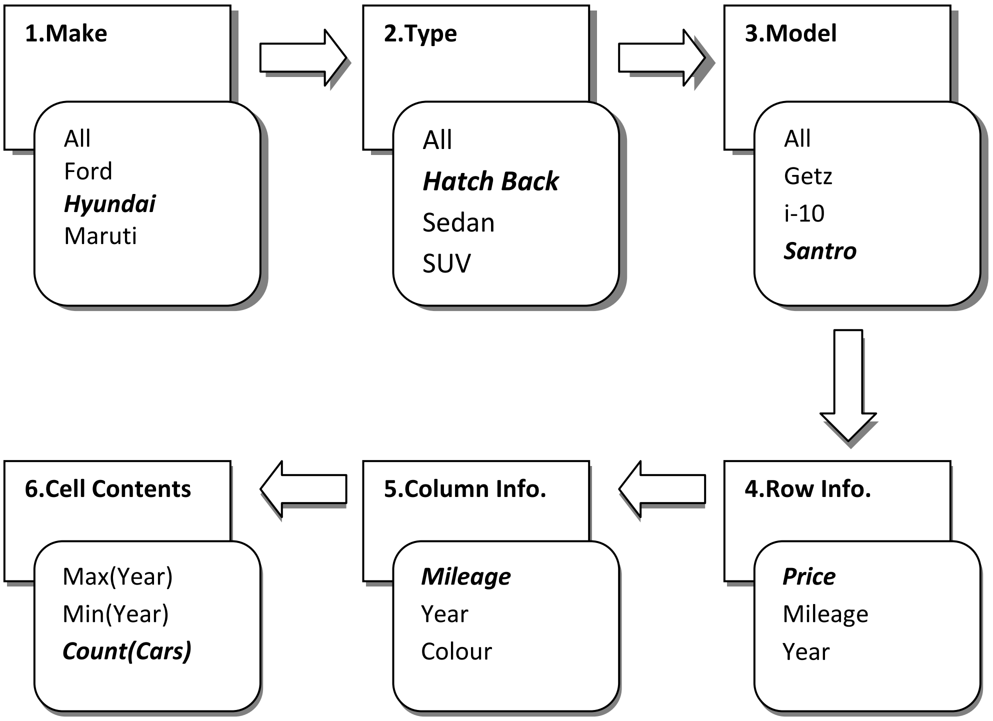


Figure 1. Block Diagram for Implementing AIM

Table 1. Preference Meta Information

Make	Type	Model	Row Info	Column Info	Cell Content
All	All	None	Make	Car Type	Selection
All	One	None	Make	Selection	Selection
One	All	None	Model	Car Type	Selection
One	One	All	Model	Selection	Selection
One	One	One	Selection	Selection	Selection

Table 2. Field Meta Information

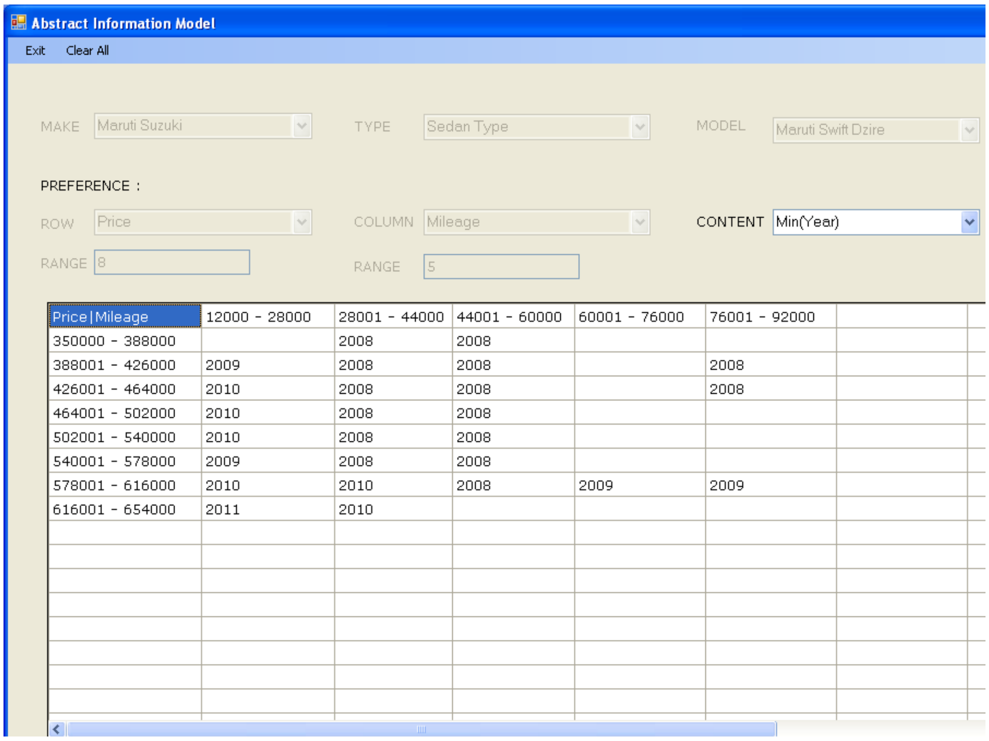
Field Name	Pref. No	Status	Count	Max	Min	Field Type
Price	1	Y	No	Yes	Yes	Numeric
Mileage	2	Y	No	Yes	Yes	Numeric
Year	3	Y	No	Yes	Yes	Numeric
Colour	4	Y	No	No	No	String
Car Name	5	Y	Yes	No	No	String

Figure 6 displays the detailed information of cars like Car Name, Price, Mileage, Year and Colour for the selected Price range of 236001-324000 and Mileage range of 3000-17000. The number of cars available in that range is 9. The detailed information of all the 9 selected cars is displayed in Figure 6 using the query refinement process.

4. Advantages

The advantages of this model are

- It provides overall view of all the data in the result set.
- It is easier to compare the details in single page.
- This model helps to take a decision by comparing different attributes.
- And it guides to get the required information in few steps.
- This grid shows the distribution of data, like cars with high price have low mileage and cars with low price have high mileage.
- Also, it provides information about the exceptional items by seeing the pattern, like the one in the lowest price and mileage range.



MAKE Maruti Suzuki
TYPE Sedan Type
MODEL Maruti Swift Dzire

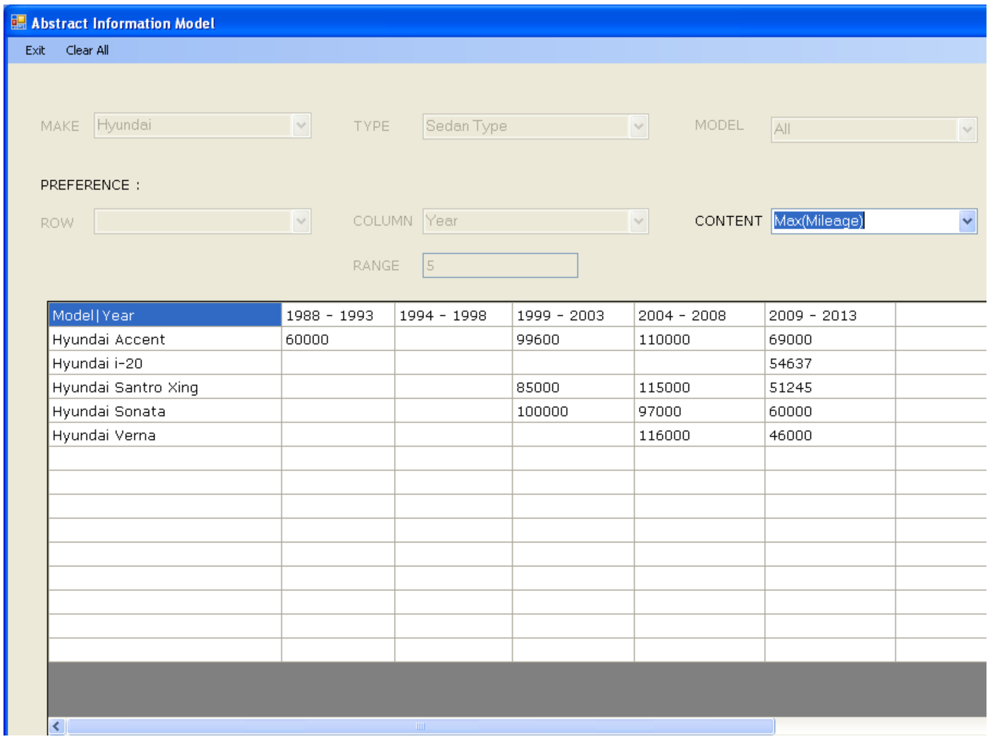
PREFERENCE :

ROW Price
COLUMN Mileage
CONTENT Min(Year)

RANGE 8
RANGE 5

Price|Mileage	12000 - 28000	28001 - 44000	44001 - 60000	60001 - 76000	76001 - 92000
350000 - 388000		2008	2008		
388001 - 426000	2009	2008	2008		2008
426001 - 464000	2010	2008	2008		2008
464001 - 502000	2010	2008	2008		
502001 - 540000	2010	2008	2008		
540001 - 578000	2009	2008	2008		
578001 - 616000	2010	2010	2008	2009	2009
616001 - 654000	2011	2010			

Figure 2. AIM displaying No of Cars with Price Vs Mileage Details for Maruti Swift Dzire



MAKE Hyundai
TYPE Sedan Type
MODEL All

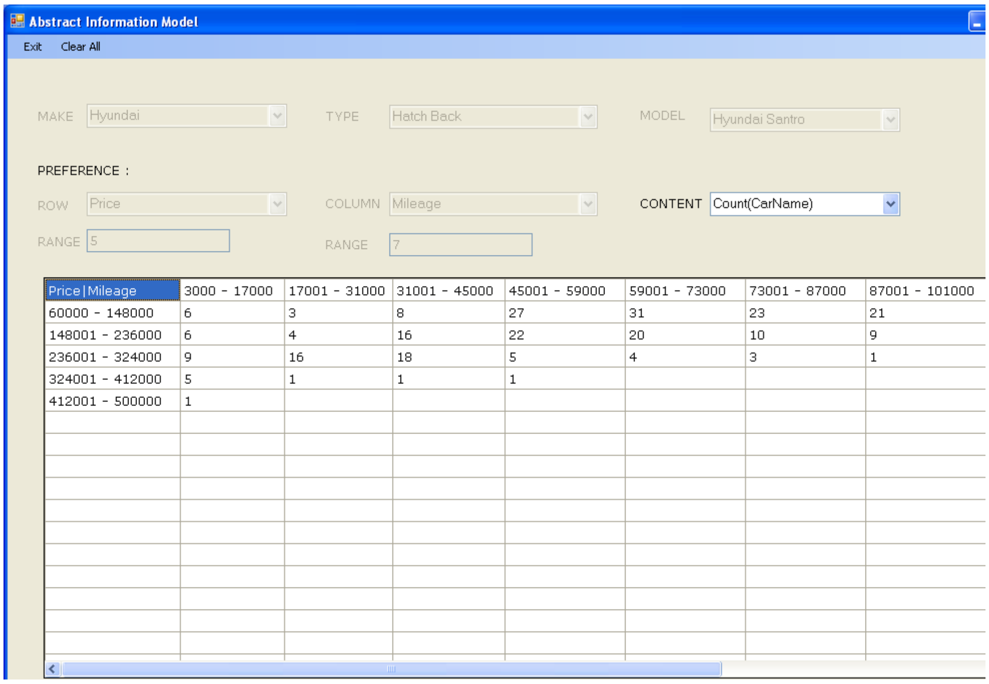
PREFERENCE :

ROW
COLUMN Year
CONTENT Max(Mileage)

RANGE 5

Model|Year	1988 - 1993	1994 - 1998	1999 - 2003	2004 - 2008	2009 - 2013
Hyundai Accent	60000		99600	110000	69000
Hyundai i-20					54637
Hyundai Santro Xing			85000	115000	51245
Hyundai Sonata			100000	97000	60000
Hyundai Verna				116000	46000

Figure 3. AIM displaying Max. Mileage Yearwise for Hyundai Sedan Type Cars



MAKE Hyundai TYPE Hatch Back MODEL Hyundai Santro

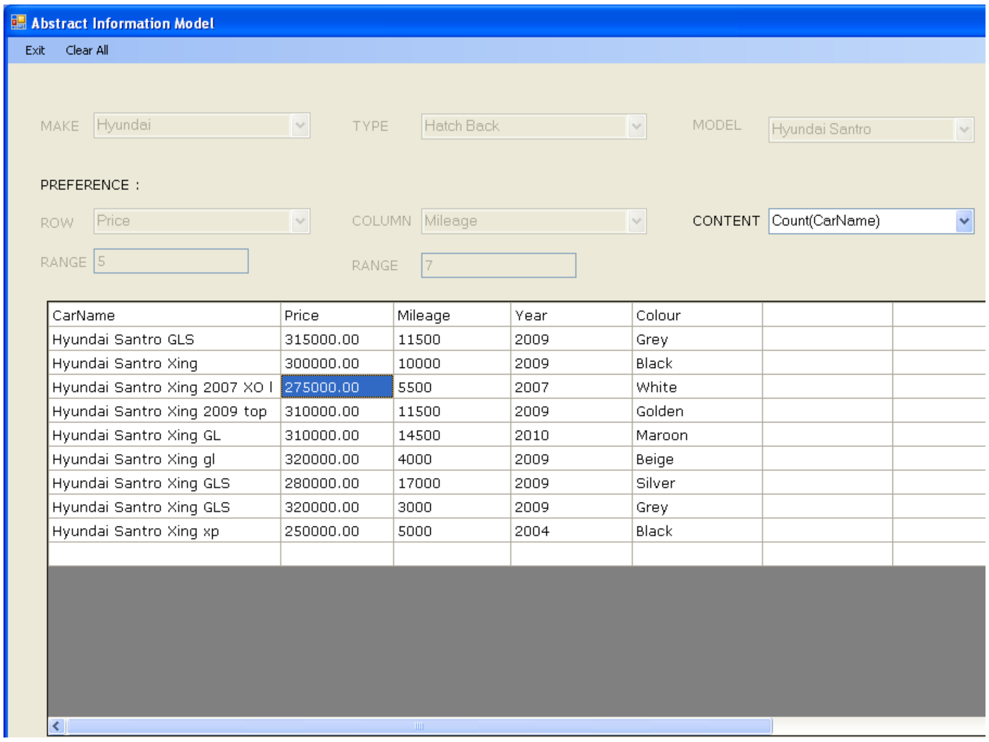
PREFERENCE :

ROW Price COLUMN Mileage CONTENT Count(CarName)

RANGE 5 RANGE 7

Price|Mileage	3000 - 17000	17001 - 31000	31001 - 45000	45001 - 59000	59001 - 73000	73001 - 87000	87001 - 101000
60000 - 148000	6	3	8	27	31	23	21
148001 - 236000	6	4	16	22	20	10	9
236001 - 324000	9	16	18	5	4	3	1
324001 - 412000	5	1	1	1			
412001 - 500000	1						

Figure 4. AIM displaying No. of cars with Price Vs Mileage for Hyundai Santro Cars



MAKE Hyundai TYPE Hatch Back MODEL Hyundai Santro

PREFERENCE :

ROW Price COLUMN Mileage CONTENT Count(CarName)

RANGE 5 RANGE 7

CarName	Price	Mileage	Year	Colour
Hyundai Santro GLS	315000.00	11500	2009	Grey
Hyundai Santro Xing	300000.00	10000	2009	Black
Hyundai Santro Xing 2007 XO l	275000.00	5500	2007	White
Hyundai Santro Xing 2009 top	310000.00	11500	2009	Golden
Hyundai Santro Xing GL	310000.00	14500	2010	Maroon
Hyundai Santro Xing gl	320000.00	4000	2009	Beige
Hyundai Santro Xing GLS	280000.00	17000	2009	Silver
Hyundai Santro Xing GLS	320000.00	3000	2009	Grey
Hyundai Santro Xing xp	250000.00	5000	2004	Black

Figure 5. AIM displaying Car details for the selected Price & Mileage

5. Conclusion

Since different users will surely have different purpose for browsing the database and they may want to compare the result set based on their preferences, The Abstract Information model reduces information overload of database query results that cannot be ordered using standard sorting technique. This method helps to compare the information available in the database and lead them towards the details they expect in minimum steps. This AIM is surely a useful, easily understandable and user friendly model for users to get the overall view of the data. Also, this model provides options to get either abstract or detail at user's discretion. This will be helpful in decision support systems.

In this AIM for presenting database query results, the PMI and FMI requires more study to decide on how it can be generalized to accommodate all such databases. Also, a more sophisticated approach is to be devised to find the user preferences. And the presentation of Abstract information may also be changed to include multiple result types, so that the user can choose from the available results, instead of displaying single preference at a time.

6. References

[1] Aditya Telang, Chengkai Li and Sharma Chakravarthy. (2009) 'One size does not fit all: Towards user & query dependent ranking for web databases', Technical Report CSE-2009

[2] Agrawal, S. Chaudhuri, S. Das, G. and Gionis, A. (2003) 'Automated Ranking of Database Query Results', Proceedings of the 2003 CIDR Conference.

[3] Amit P. Sawant and Christopher G. Healey. (2008) 'Visualizing Multidimensional Query Results Using Animation', Visualization and Data Analysis 2008. Proceedings of the SPIE, Volume 6809, pp. 680904-680904-12.

[4] Beygelzimer, S. Kakade and Langford, J. (2006) 'Cover trees for nearest neighbor', In ICML, pages 97–104.

[5] Chaudhuri, S. Das, G. Hristidis, V. and Weikum. G. (2006) 'Probabilistic information retrieval approach for ranking of database query results', ACM Trans. Database Syst., 31(3):1134-1168.

[6] Gautam Das, Vagelis Hristidis, Nishant Kapoor, and Sudarshan, S. (2006) 'Ordering the Attributes of Query Results', SIGMOD 2006, June 27–29.

[7] Kaushik Chakrabarti, Surajit Chaudhuri and Seung-won Hwang. (2004) 'Automatic Categorization of Query Results', SIGMOD 2004, June 13–18, Paris, France.

[8] Kimberlee, A. Kemble et al. (2006) 'Efficient presentation of database query results through audio user interfaces'.

[9] Kostas Stefanidis, Marina Drosou and Evaggelia Pitoura (2009) '"You May Also Like" Results in Relational Databases', In ACM, 2009.

[10] Liu, B. and Jagadish, H. V. (2009) 'Using Trees to Depict a Forest', In VLDB.

[11] Sumathi, M. and Kalaikumaran, T. (2012) 'Abstract Information Model for Presenting Database Query Results', International Conference on Computer Communication and Informatics (ICCCI-2012), IEEE Catalog Number: CFP1208R-PRT, ISBN: 978-1-4577-1581-5, Vol.1, pp.253-258.

[12] Zhiyuan Chen and Tao Li. (2007) 'Addressing Diverse User Preferences in SQL-Query-Result Navigation', In SIGMOD, pages 641-652.