FUZZY QUERY PROCESSING IN DISTRIBUTED DATABASES

[1]ROHAN BADLANI, [2]SUREKHA BHANOT, [3]ADITYA MANGLA

Birla Institute of Technology and Science, Pilani, India
E-mail: rohan.badlani@gmail.com, surekha@pilani.bits-pilani.ac.in, aditya21494@gmail.com

Abstract- The problem of evolving databases to make them more intuitive, user-friendly and to be able to answer vague human queries with separate needs for each user has become a popular research topic. The solution to this problem in part has been proposed via databases that aim at inserting fuzzy data into databases hence handling vague human like queries. It has been suggested in many research papers that fuzziness may be applied to databases. However, this approach is infeasible and inefficient for real time processing. In the past 30 years of research, fuzzy databases are still not popular in industry because of unwillingness of companies to replace crisp data with fuzzy data in their databases due to excessive precomputation and possible chances of data inconsistency. Having fuzzy databases also places severe constraints on the database as it will become very difficult to run crisp queries on fuzzy databases. This problem becomes even more complex with the advent of "Big Data". This paper proposes a three pronged fuzzy logic based technique as a layer of computation above traditional query processing to solve such queries in real time. This fuzzy logic based approach to querying in distributed databases can be used to solve ambiguous queries, incorporating the preferences of each user in the current scenario of excessive data. The results obtained using the fuzzy logic approach are compared with those obtained using traditional approach in terms of accuracy, time taken for each approach and closeness of the results to users requirements.

Keywords- Fuzzy logic, Fuzzy Inference System, Big Data, Distributed Query Processing (DQP), Distributed Databases, Fuzzy SQL, Yager's intersection, Mamdani's implication, Fuzzy Rule Base (FRB), Fuzzy Expert System(FES).

I. INTRODUCTION

Database Management Systems find intensive applications in domains like finance, banking, web-based applications, personnel records and population surveys. Information retrieval is one of the core field of Computer Science and a lot of research has been going on in the field since 1950[1]. Today's computers store wide variety of information and in huge amount. Effectively querying databases to obtain required information, its analysis and identifying patterns is the quintessential purpose behind maintaining database systems[2, 3].

Querying is of 2 types – relational algebra querying (range or equality based) and vague, uncertain, human language based querying. The traditional approaches to querying on databases is crisp in the sense that the data required to be retrieved follows a given crisp condition. This approach is very limited and insufficient in modern real world applications. In modern applications we need to run dynamic uncertain queries like "Who is the best employee of a firm with 10,000 employees?" or "Who is the best all-rounder student in a batch of students?" or "Which is the best second hand car to buy in a collection of second hand cars?"

These kind of vague queries with terms like best and worst can be solved by adopting two techniques -- traditional approach and fuzzy logic based approach[4]. There have been many attempts to extend SQL on relational databases with support for fuzzy querying.[5]

This paper proposes an approach to retrieve results for flexible and vague queries on regular databases by supporting user's preferences for each attribute. The traditional and fuzzy logic based approach are compared based on the accuracy and time taken to resolve the query. The fuzzy expert system approach suggested by the paper has been implemented to solve this kind of vague query on a second hand car database[17] to obtain the best car for a particular user depending on his/her preferences. Finally, the results obtained using this approach and the traditional approach in terms of accuracy, feasibility, response and processing time and user-friendliness are compared[6].

II. TRADITIONAL APPROACH

A traditional approach to solve the above mentioned kind of user specific uncertain queries would involve the following steps.

1. Setup the database and the user interface.
2. Take user requirements for each attribute.
3. Query the database for each attribute requirement.
4. Aggregate the results for each attribute.

The above mentioned general approach to solve fuzzy queries would involve a selection part and an ordering part. Selection is done based on the selection part and ordering of the results based on the ordering constraint. But, we have the selection criterion valued over [0, 1] as a membership grade function. One major problem is then to find out an appropriate fuzzy set expression representing the solution satisfying criterion.
Figure 1 shows the different types of query processing. The focus of this paper is to deal with vague, fuzzy and flexible queries in regular databases.

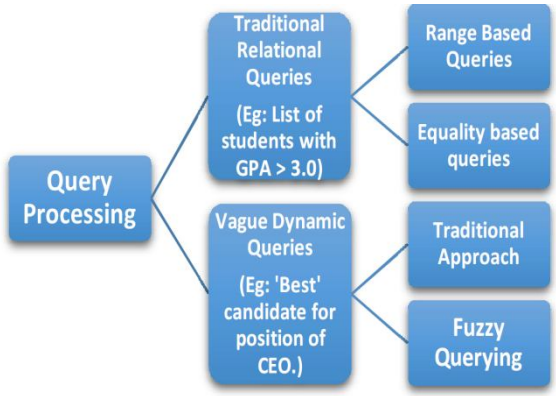

Fig 1: Two types of query processing techniques

An example of a fuzzy query is presented to illustrate the problems of traditional ways of solving these kinds of queries. Given a database of second hand cars with the following 5-attributes - price, year, kilometres travelled, body type, engine capacity. Suppose a user wants to find the best car according to his/her preferences, needs and budget. But any traditional approach can sort the database entries using a single attribute at a time. So, in order to deal with user requirements on multiple attributes aggregation of the results obtained using each of the attributes is to be done which is computationally very expensive.

Moreover, it might so happen that none of the database entries in the database satisfy the user needs for a single attribute. In such cases, the result for that particular attribute cannot be considered valid for aggregating with other results. This may lead to results that do not entirely comply with the user's requirements. Also, this approach does not give the user the flexibility of giving different weights to different attributes. These issues indicate the need for a system to incorporate the shortcomings of the traditional query approach.

Non fuzzy set based approaches aiming at discriminated answers discussed in [7], [8], [9], [10] shows that in each case queries are expressible in the context of fuzzy sets. Each of these approaches proposes only one or more aggregation mechanism and it is then clear that fuzzy sets provide a much more general framework where the user can choose the appropriate aggregation mechanism.

III. FUZZY LOGIC BASED QUERY PROCESSING

The crisp set theory proposes the idea that an element can either belong to set or not. However, with the increasing complexity of the user requirements, it has become very difficult to formulate accurate and relevant results. The natural language in which people communicate involves a lot of vagueness that in order to be accommodated in the computations can be done through the use the fuzzy set theory. L. A. Zadeh, the father of Fuzzy Set theory initiated a revolution in the field of Mathematics through his work on Fuzzy set and Fuzzy Logic. [11]

Fuzzy set theory defines the extent of belongingness of an element in the set using a membership grade function (mgf). Since the occurrence of an element in the set is defined in a new way, the logical operations like AND, OR, NOT, all form a separate meaning in the Fuzzy Logic. Various definitions exist for each of these operations and it depends on the situation, which definition perfectly suits the requirement of the system.

A fuzzy expert system is an expert system that uses a collection of fuzzy based rules (FRB) and membership functions to derive the conclusions. A fuzzy expert system (FES) uses fuzzy logic instead of traditional Boolean logic in order to derive inferences. [12]

Fuzzy Rule Base (FRB):

A collection of fuzzy based rules is called a Fuzzy Rule Base. These rules are generally in the IF ==> THEN format. The "If" side of the rule is called the Antecedent (premise) and the "Then" side of the rule is called the Consequent (conclusion). The input variables can be numeric or fuzzy in nature.[13]

The fuzzy expert system (FES) accepts a combination of inputs from the user, performs certain computations and generates an output value. This process involves the following 4 major steps:

1. **Fuzzification:**

The given membership functions for different input variables are applied to their crisp values to determine the degree with which they belong to a fuzzy set. This degree is referred to as the rule's α value or Degree of Fulfilment (DoF). If a rules premise has a non-zero α value, then the rule is said to be fired.

2. **Inference System:**

In the inference process α value of each rule is applied to the consequent of that rule. This results in one fuzzy set being assigned to each output variable for each rule.

3. **Aggregation:**

Aggregation involves unification of all outputs obtained. The membership function of all rule consequents are combined into a single fuzzy set. The method used for aggregation may vary from system to system.

4. **Defuzzification:**

A crisp value is assigned to each output variable based on the fuzzy set s arrived after the aggregation process.

Figure 2 illustrates the sequence of steps performed in a typical Fuzzy Expert System.

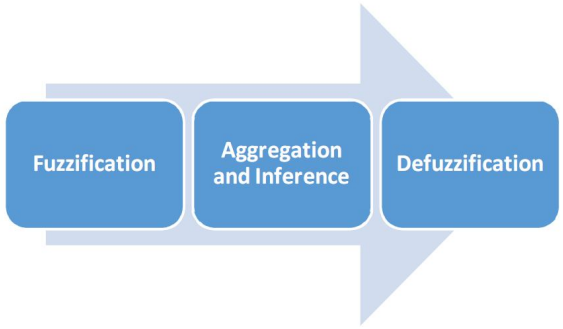

Fig 2: Fuzzy Expert System (FES).

IV. PROPOSED APPROACH TO FUZZY QUERY PROCESSING

Any query in today's world is specific to the user's requirements and involves some amount of vagueness and fuzziness. Many times, though the user is certain about the range of his requirements but due to the multiplicity of the records that satisfy the user's specification it is very difficult for the user to find the most suitable record for him/her.

The proposed approach discussed below incorporates fuzzy queries in regular databases.

I. Filtering Dataset:

This involves the user to specify his/her requirements as a range based or equality based query.[14] The query runs in the form of a relational algebra query and a filtered set of records is returned.

II. Fuzzification:

For each attribute of tuple, the membership grade value for each attribute value is calculated. Figure 3 shows a possible membership grade function for an attribute. Now using the fuzzified value of each attribute a rule from the fuzzy rule base (FRB) is fired. Figure 4 shows a sample fuzzy rule base.

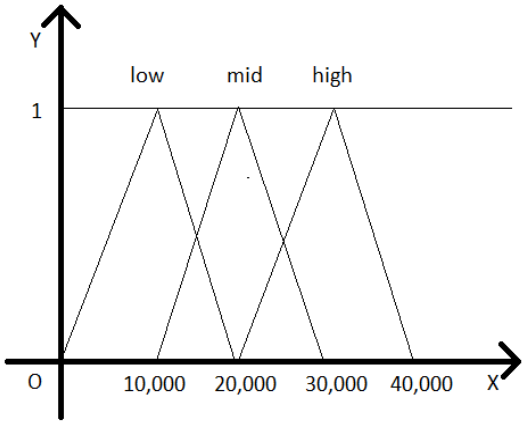

Fig 3: Membership grade function for low, mid and high linguistic values for attribute A.

III. Inference Evaluation:

For each rule fired, the truth value of the rule is calculated using Yager's intersection formula. The priorities for each attribute are incorporated by using the priority value (between 0 and 1) as a coefficient to each of the terms corresponding to an attribute in the Yager's Formula. These priorities are to be specified by the user along with the query. Thus the truth value is obtained after taking into account the user's preferences.

A \ B	low	mid	high
low	low	low	med
mid	low	med	high
high	med	high	high

Fig 4: Fuzzy Rule Base generated for combinations of each of the fuzzified attribute values.

Yager's intersection formula

Truth value =
$1 - \min (1, (p1*(1-a1)^w + p2*(1-a2)^w + \ldots + pn*(1-an)^w)^{1/w})$

Where, p1, p2, ..pn are the priorities of attributes a1, a2, .. an respectively and w is the Norm chosen.

Our key contribution suggested in this paper depends on the modification of the general Yager's intersection formula to the one mentioned above which includes the priorities of each of the attributes.

Using Mamdani's implication we find the suitable region for each rule.

IV. Aggregation:

Using the weighted centre of gravity method and the area of the region obtained from each rule we obtain the final score value of each record.
Aggregated Value =

$$\frac{\sum_{i=1}^{n} Ci * Ai * Wi}{\sum_{i=1}^{n} Ai * Wi}$$

V. Defuzzification:

The final obtained Centre of Gravity gives us the final score value. The final score obtained using the Centre of Gravity method for each of the records lets us obtain a complete order on the records and hence we have been able to handle vague queries fired on the database. Figure 5 shows the output transformation of the truth value obtained.

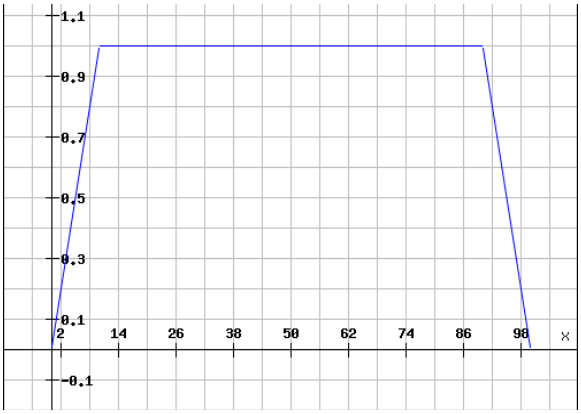

Fig 5: Membership grade function of the output.

V. DISTRIBUTED QUERY PROCESSING

Query processing in general is essentially of 2 types - central and distributed querying[15]. In the centralized query model all the data is stored on a single mainframe which increases the load, high communication costs and mainframe failure risks. In the flood of ever increasing data (Big Data), this model is very limited and fails.[16] So we propose the use of Distributed Query Processing (DQP) in distributed system that aims to optimize operating costs and response time that are associated with the query. DQP works in 3 phases – local processing phase, reduction phase and final processing phase over each node above which exists the fuzzy based computation layer processing the input data and obtaining precise results for the ambiguous query.

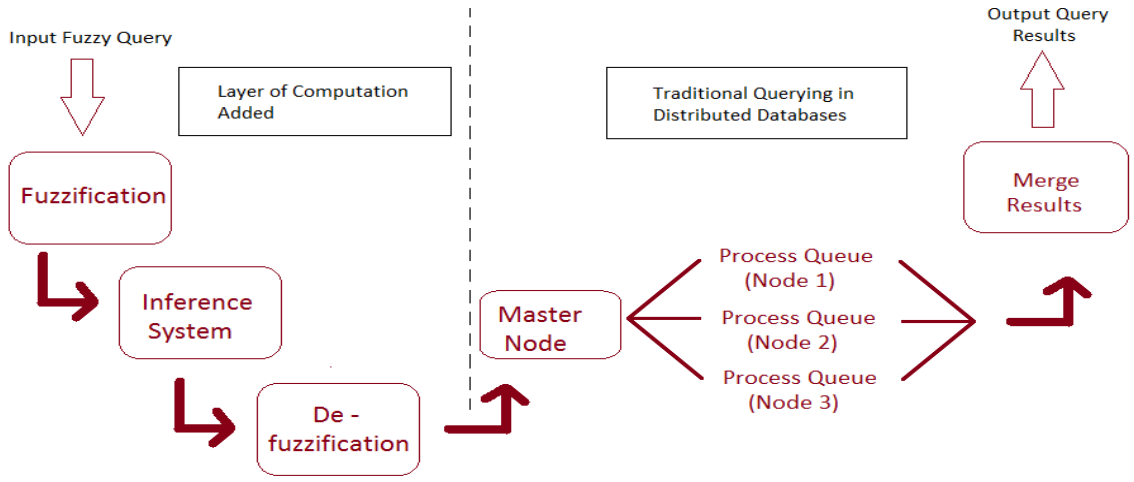

Figure 6: Illustrating the Proposed Appraoch to Fuzzy Query Processing In Distributed Databases

Fuzzy querying on distributed databases containing crisp data thus provide appropriate results for vague human queries exploiting the advantages of distributed databases.

Figure 6 illustrates the proposed model for processing fuzzy queries on a distributed database with crisp data values.

VI. RESULTS & DISCUSSION

The above mentioned fuzzy based approach is applied on a classical web based searching query. Users today use the Internet in order to access most of the information in the world. Consider a dataset of second hand cars. The aim is to find the best car according to the user's requirements. Table I shows a subset of original dataset used for analysis[17]. Consider that the strict (crisp) filtering conditions on the dataset and the preferences for each attribute are provided by the user. Following section discusses the traditional approach of solving this problem and then compares the results with the proposed approach.

Traditional Approach:

The traditional approach involves sorting the results according to each of the attributes one at a time and ranking the results. In the above specification, we can sort the cars according to the price and rank the results. Table 2, 3 shows the results of sorting the cars according to least price code and least kilometres travelled respectively.

In order to aggregate the results so obtained, one of the process that can be used is to sum the ranks obtained for each individual car and then sort the cars according to their overall rank. Table 4 shows the results so obtained.

ID	MAKE	YEAR	PRICE	KM USED	CYLINDER VOLUME
1	TOYOTA	2007	410000	38000	1300
2	NISSAN	2007	325000	50000	1500
3	HONDA	2005	385000	59000	1500
4	TOYOTA	2007	360000	59000	1300
5	TOYOTA	1989	50000	62665	1300
6	TOYOTA	2008	615000	67000	1300
7	TOYOTA	2008	575000	69000	1500
8	TOYOTA	2006	550000	73000	1500
9	TOYOTA	2006	450000	82000	1490
10	TOYOTA	2006	325000	85000	1000
11	TOYOTA	2000	325000	113000	1500
12	TOYOTA	2000	218000	129000	1500
13	NISSAN	2001	195000	145000	1500

Table I: Subset of Second Hand Car Dataset for Evaluation Purpose.

ID	MAKE	YEAR	PRICE	KM USED	CYLINDER VOLUME	PRICE RAN
6	TOYOTA	2008	615000	67000	1300	1
7	TOYOTA	2008	575000	69000	1500	2
8	TOYOTA	2006	550000	73000	1500	3
9	TOYOTA	2006	450000	82000	1490	4
1	TOYOTA	2007	410000	38000	1300	5
3	HONDA	2005	385000	59000	1500	6
4	TOYOTA	2007	360000	59000	1300	7
2	NISSAN	2007	325000	50000	1500	8
10	TOYOTA	2006	325000	85000	1000	9
11	TOYOTA	2000	325000	113000	1500	10
12	TOYOTA	2000	218000	129000	1500	11
13	NISSAN	2001	195000	145000	1500	12
5	TOYOTA	1989	50000	62665	1300	13

Table II: Second hand car dataset sorted on PRICE attribute.

ID	MAKE	YEAR	PRICE	KM USED	CYLINDER VOLUME	KM RANK
1	TOYOTA	2007	410000	38000	1300	1
2	NISSAN	2007	325000	50000	1500	2
3	HONDA	2005	385000	59000	1500	3
4	TOYOTA	2007	360000	59000	1300	4
5	TOYOTA	1989	50000	62665	1300	5
6	TOYOTA	2008	615000	67000	1300	6
7	TOYOTA	2008	575000	69000	1500	7
8	TOYOTA	2006	550000	73000	1500	8
9	TOYOTA	2006	450000	82000	1490	9
10	TOYOTA	2006	325000	85000	1000	10
11	TOYOTA	2000	325000	113000	1500	11
12	TOYOTA	2000	218000	129000	1500	12
13	NISSAN	2001	195000	145000	1500	13

Table III: Second hand car dataset sorted on KM travelled attribute.

Fuzzy Logic Based Approach:

The fuzzy logic approach takes into account all the attribute preferences of the user at the same time in order to generate the score. This score generated for each of the cars is used to sort the results and hence obtain the 'best' car for the user specification.

The fuzzy based approach involves a sequence of steps. The first step involves the fuzzification process. This is done by categorizing each of the attributes and assigning a membership function for each of the attribute value.

ID	MAKE	YEAR	PRICE	KM USED	CYLINDER VOLUME	AGG RANK
1	TOYOTA	2007	410000	38000	1300	6
6	TOYOTA	2008	615000	67000	1300	7
3	HONDA	2005	385000	59000	1500	9
7	TOYOTA	2008	575000	69000	1500	9
2	NISSAN	2007	325000	50000	1500	10
4	TOYOTA	2007	360000	59000	1300	11
8	TOYOTA	2006	550000	73000	1500	11
9	TOYOTA	2006	450000	82000	1490	13
5	TOYOTA	1989	50000	62665	1300	18
10	TOYOTA	2006	325000	85000	1000	19
11	TOYOTA	2000	325000	113000	1500	21
12	TOYOTA	2000	218000	129000	1500	23
13	NISSAN	2001	195000	145000	1500	25

Table IV: Aggregate Rank obtained for Traditional Approach

The second step involves the inference system. In order to deduce the output we need a fuzzy rule base. This rule base can be developed in the following 4 steps:[18]

1. Divide the attribute into a fixed number of categorical values and assign linguistic meanings to each range of values.
2. Generate fuzzy rules using all the linguistic terms assigned to each of the attribute value.
3. Count the rules with the highest number of conflicting rules. The rules with maximum number of count are kept in the system.
4. The fuzzy rule base contains all the required results.

The next step is to take the users preference for each of the attributes. User can express the preference by a number between 0 - 10. The transformation matrix specifies the preferences of each of the user. A null field indicates that the particular attribute does not affect the user. Table 5 shows the transformation matrix illustrating the preferences of different users. For example, user id 1 gives equal preference to all attributed while user id 2 gives less preference to cylinder volume.

USER ID	PRICE	KM TRAVE	YEAR	CYLINDER VOLUME
1	10	10	10	10
2	8	8	8	5
3	6	6	4	0
4	4	4		
5	1			

Table V: User priorities for each attribute.

In order to rank all the cars according to the preferences of user, the state matrix of each car is to be prepared. This state matrix shows the final score of each car. The final score is obtained by using the fuzzy rule base and giving different priorities to each of the attributes and aggregating the centre of gravities of each of the rules fired using the Yager's intersection and the Mamdani's implication formulas described above. Using the priorities for User Id 1(equal for all), the following results were obtained using the fuzzy-based approach.

ID	FUZZY RANK	AGG SCORE	YEAR RANK	PRICE RANK	KM RANK	CYL RANK
1	1	11	3	9	1	9
4	2	23	4	7	3	10
5	2	24	13	1	5	11
3	3	29	9	8	4	2
12	3	33	11	3	12	6
2	4	33	5	4	2	1
11	4	32	12	6	11	5
6	5	34	1	13	6	12
7	6	22	2	12	7	3
8	7	25	6	11	8	4
13	8	30	10	2	13	7
9	8	32	7	10	9	8
10	9	36	8	5	10	13

Table VI: Fuzzy Rank and individual attribute ranks for subset of second hand car dataset.

COMPARISON OF RESULTS

In fuzzy based approach we incorporate all the requirements of the user by taking in the user's preferences for each of the attribute. On the other hand the results obtained using the traditional approach do not take into account the users preferences, but just a ranking process, which may not lead to the best outcomes.

Table VII compares the fuzzy results with the traditional approach and analyses the reasons for the success of the Fuzzy Based Approach. The numbers in the last row indicate the ratio of number of times fuzzy based approach succeeds in incorporating user's needs to the number of times it fails. For

example, the fuzzy based approach is able to reflect the results of the user's preference 10 times as the aggregate rank does and fails only 3 times.

ID	FUZZY RANK	AGG RANK	YEAR RANK	PRICE RANK	KM RANK	CYL RANK
1	1	1	3	9	1	9
4	2	3	4	7	3	10
5	2	4	13	1	5	11
3	3	6	9	8	4	2
12	3	9	11	3	12	6
2	4	9	5	4	2	1
11	4	8	12	6	11	5
6	5	10	1	13	6	12
7	6	2	2	12	7	3
8	7	5	6	11	8	4
13	8	6	10	2	13	7
9	8	8	7	10	9	8
10	9	11	8	5	10	13
		10:03	08:05	08:03	11:01	07:06

Table VII: Traditional Aggregate Ranking, Individual Attribute Ranking and Fuzzy Based Approach Ranking of cars.

It is evident that the proposed fuzzy based approach not only captures all the individual attributes but also the aggregate rank (using equal priority for each of the attributes.) Hence, it is established experimentally that the results obtained from the fuzzy logic approach better accommodate the user specification and provide better results for vague queries.

CONCLUSIONS

There are three primary advantages of the proposed Fuzzy Logic Approach when used instead of the Traditional approach.

1. Ability to incorporate user preferences in the querying process.
2. Less computation cost: In the traditional approach, records need to be ordered based on all attributes one by one and requires multiple aggregations and hence a lot of processing is required as compared to the fuzzy based approach.
3. Although the traditional approach seems a little better in terms of user experience as he/she does not have to specific his/her own set of preferences but the results obtained by this approach are not the best possible results (as illustrated in the example in our paper). Adding a Fuzzy Logic Approach to the traditional search scheme on databases provides more satisfactory results.

It can be clearly seen that Fuzzy based approach is the best approach for processing human-like vague queries. This can also be used in a variety of applications like websites that suggest second hand cars, mobile phones, or even developing intuitive user friendly applications like finding the best employee of a company.

ACKNOWLEDGEMENTS

Rohan Badlani is very grateful to the academicians of BITS, Pilani for allowing him to conduct this study on the application of fuzzy logic in Databases. This work was supported by Computer Science, Electronics and the Mathematics Department, BITS, Pilani. The authors express their deep gratitude to all the professors for providing their valuable guidance, encouragement, support and advising technical points from time to time in the course of preparation of this paper.

REFERENCES

[1]. Singhal Amit; "Modern Information Retrieval: A Brief Overview". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24, No. 4. (2001), pp. 35-42.
[2]. Garcia-Molina, Hector; Ullman, Jeffrey D.; Widom, Jennifer (2009). "The Worlds of Database Systems".
[3]. Database systems: the complete book (2nd ed.). Upper Saddle River, N.J.: Pearson Prentice Hall. ISBN 978-0131873254.
[4]. SS Singh, Rishi Sayal and Venkat Rao "Analysis and Usage of Fuzzy Logic for optimized Evaluation of Database Queries", International Journal of Computer Applications, Feb 2011.
[5]. P. Bosc & O. Pivert "Fuzzy Queries And Relational Databases"
[6]. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database System", Fifth Edition, Pearson Education, Second Impression, 2009. Book.
[7]. Lacroix M. & Lavency P., Preferences : putting more knowledge into queries, Proc. 13th Conference VLDB, Brighton, GB, September 1987, pp 217-225.
[8]. Morro A., VAGUE : a user interface to relational databases that permits vague queries, ACM Transactions on Office Information Systems. 6(3), 1988, pp 187-214.
[9]. Ichikawa T. & Hixakawa M., ARES : a relational database with the capability of performing flexible interpretation of queries, IEEE Transactions on Software En~ineerine. 12(5), 1986, pp 624-634.
[10]. Chang C.L., Decision support in an imperfect world, Research regort RJ3421. IBM San JosE, CA, USA, 1982.
[11]. M. Ganesh: Introduction to Fuzzy sets and Fuzzy Logic, PHI, 2006. Book.
[12]. Anjali Baghel, Tilotma Sharma, "Survey on Fuzzy Expert System", International Journal of Emerging Technology and Advanced Engineering. (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 12, December 2013.
[13]. Les M Sztandera, "Generation of Fuzzy IF..AND..THEN rules", Philadelphia College of Textile.
[14]. Vaclav Bezdek, "Possible Use of Fuzzy Logic in databases", Faculty of Management & Economics, Tomas Bata University in Zlen, Czech Republic.
[15]. Abhijeet Raipurkar and G.R. Bamnote, "Query Processing in Distributed Database Through Data Distribution", International of Advance Research in Computer and Communications Engineering, Feb 2013.
[16]. Thomas H Davenport and Jill Dyche, "Big Data in Big Companies", International Institute of Analytics, May 2013.
[17]. Sameerchand Pudaruth, "Predicting the Price of Used Cars using Machine Learning Techniques". International Journal of Information & Computation Technology. ISSN 0974-2239.
[18]. L.X. Wang and J. M. Mendel, "Generating Fuzzy Rules from numerical Data with Applications", USC SIPI Rep No 169, University of Southern California, Los Angeles, 1991.

★★★