

# A Framework for Computing Linguistic Hedges in Fuzzy Queries

V.Balamurugan<sup>1</sup> and K.Senthamarai Kannan<sup>2</sup>

<sup>1</sup>Department of Computer Science & Engineering, SCAD College of Engineering & Technology, Cheranmahadevi, Tirunelveli, India, e-mail: [bala\\_vm@yahoo.com](mailto:bala_vm@yahoo.com)

<sup>2</sup>Department of Statistics, Manonmaniam Sundaranar University, Tirunelveli, India, e-mail: [senkannan2002@gmail.com](mailto:senkannan2002@gmail.com)

## Abstract

*Most query languages are designed to retrieve information from databases containing precise and certain data using precisely specified commands. Due to the advancements in various kinds of data repositories in the recent years, there is a steep increase in complex queries. Most of the complex Queries are uncertain and vague. The existing Structured Query Language exhibits its inefficiency in handling these complex Queries. This paper proposes a framework to handle the complexities by using fuzzy set theory. The Fuzzy Query with linguistic hedges is converted into Crisp Query, by deploying an application layer over the Structured Query Language.*

**Key Words:** Fuzzy Query, Linguistic variable, Membership value, Alpha-cut, SQL

## 1 Introduction

In many query situations, the databases contain precise and certain data. The queries on such data are also need to be precisely and certainly expressed. However, in recent years, the development in various kinds of data repositories has paved the way to new potentials as to data querying. The application of fuzzy set theory to relational data models is one major shift in addressing the vagueness in the data and the query specification. The research in this direction includes extensions to SQL to facilitate vague queries on relational databases [1],[2],[7], functional dependencies in fuzzy relational data models [8], fuzzy extensions to relational calculus and relational algebra [4],[5],[6],[9],[12], and a logic based approach to the fuzzy relational databases to deal with various forms of fuzziness and a domain calculus based fuzzy query language [3],[11].

Complexity normally arises from uncertainty in the form of ambiguity. The computerized system is not capable of addressing complex and ambiguous issues. However, the human have the capacity to reason “approximately”. As a result, human, when interacting with the database, want to make complex queries that have a lot of vagueness present in it.

The traditional tools used for computing, are crisp, deterministic and certain in nature. Here certainty indicates that the structures and parameters of the model to be definitely known. But in real situations, these are not crisp and deterministic and therefore, cannot be described precisely.

The techniques based on the fuzzy set theory are very much useful while modeling the uncertainties especially, when the uncertainties are non-random in nature. The proposed framework will perform the necessary translation, by acting as a middleware. Main aim of this model is to exploit the standard facilities available in the modern DBMS. The easiest way to do this is, to use classical relational databases and formulate a front end that will allow fuzzy querying to the database. Here the underlying database will always be crisp. Therefore, the fuzziness is incorporated in the front end only.

## 2 Fuzziness in Database

Until recently, most data processed by computers, were numeric or alphanumeric data concerning the system. Now it is possible to store graphs, pictures or even voice and movies in the database. The data stored in the database is normally crisp in nature. But the request for the required information may be of fuzzy nature. The fuzziness may be classified into two categories viz. Impreciseness and Vagueness. In the real time situation, people express their ideas using the natural languages. Normally natural language has a lot of vagueness and ambiguity. However, while applying one's thoughts as a query in terms of natural languages into the database, a lot of problems are experienced due to the inefficiency of RDBMS to handle such queries. Consider the query "Give me the names of the Young terrorists who were involved in the recent bomb blast that occurred in the region in and around Jammu Kashmir". This query cannot be processed directly by the SQL, since it contains a lot of vagueness like Young terrorists, recent bomb blast and in and around Jammu Kashmir. The best remedy for modeling the above situation is by the use of Fuzzy Sets.

### 2.1 Linguistic Variables And Hedges

Natural language consists of fundamental terms called "atomic terms". Examples of some atomic terms are "medium", "young" and "beautiful", etc. Collection of atomic terms are called composite terms. Examples of composite terms are "Very slow car", "Slightly Young student", "fairly beautiful lady", etc. The atomic terms are called linguistic variable in Fuzzy set theory [10]. A linguistic variable differs from a numerical variable in that, its values are not numbers but words or sentences in Natural languages. The purpose of using the linguistic Variable is to provide a means of approximate characterization of phenomena that is not defined properly. Linguistic variables can be characterized by the use of trapezoidal shaped possibility distribution. In linguistics, fundamental atomic terms are often modified with adjectives (noun) or adverbs (Verbs) like very, low, slightly, more-or-less, fairly, almost, roughly, etc. These modifiers are called linguistic hedges. When a fuzzy set is used for interpretation, the linguistic hedges have the effect of modifying the membership function for a basic atomic term.

## 3 Proposed Framework

An application layer is placed over the SQL and it will perform the necessary translation by acting as a middleware. It is assumed that the underlying database will be crisp. Therefore the fuzziness is incorporated in the front end only. At the front end, initially the Fuzzy sets / Linguistic Variables on the necessary domains are defined. For example, in Table.2, the fuzzy sets *Young*, *Middle* and *OLD* are defined on the attribute *AGE*. Similarly, the attribute *Area\_Of\_Operation* can be associated with fuzzy sets *near*, *far away*, etc.

### 3.1 Pre-Processing Stage

The framework is divided into Pre-Processing stage as shown in Figure.1 and Query Implementation stage. The preprocessing works involve collecting the information from different data repositories, bringing it to the standard format and storing it at single location. The second step is to identify the fuzziness that can be associated with the different attributes. In the third step, we form different fuzzy sets. In the fourth step, a table that contains the membership values is formed. Finally, defuzzification process is carried out by finding  $\alpha$ -Cut [10]. In the pre-processing stage, additional information may be needed based on the nature of the attributes. For example, the attribute *AreaOfOperation* in the Table.1, needs the information like distance, direction, etc.

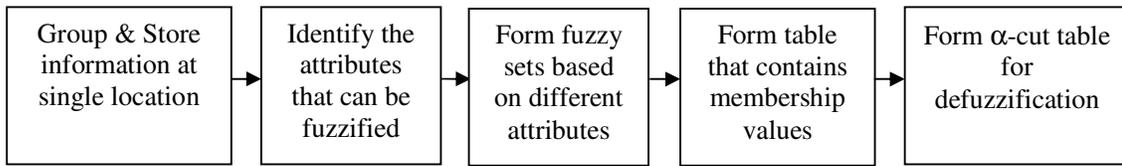


Figure 1: Block diagram of Pre-Processing Stage.

Table.1: Information about Terrorists.

Terrorist Name	Age	Cases Involved	AreaOfOperation
Anna	22.50	1	Srilanka
Bala	25.00	1	Srilanka
Daniel	21.50	2	England
Abdul	40.00	8	Pakistan
Mohanto	55.00	3	Assam
Quirola	58.00	3	Assam
Nasar	62.00	2	J&K
Ahmed	77.00	2	J&K

### 3.2 Information Requirements

The tables that are required for the framework are, *Meta\_Information Table (Table.2)*, *Linguistic\_Hedges Table*, *Membership Table* and *Alpha\_cut table*. Meta-Information table contains all the information related to fuzzification of the different attributes of different tables. The description of the *Meta\_Information* table is as follows.

- Linguistic\_Term : used to store the name of the fuzzy set.
- Table\_name : used to refer the name of the table in which the attribute associated with the fuzzy set is available.
- Column\_name : used to refer the attribute associated with the fuzzy set
- Alpha ( $\alpha$ ) : Lower range of the SUPPORT [10] of the fuzzy set.
- Beta ( $\beta$ ) : Lower range of the CORE [10] of the fuzzy set.
- Delta ( $\delta$ ) : Upper range of the CORE of the fuzzy set.
- Gamma ( $\gamma$ ) : Upper range of the SUPPORT of the fuzzy set.

The *Linguistic\_Hedges* table contains the linguistic hedges [10] and the computation formula for computing new membership values called manipulated membership values. Membership table has three columns viz., Col\_Name, Membership\_Value and Manipulated\_Membership\_Value. The column Col\_Name refers to the value of the corresponding attribute on which the membership value is to be computed. The values of the Col\_Name are calculated during the run time. The column Membership\_Value refers to the degree of membership of the attribute in the fuzzy set. The column Manipulated\_Membership\_Value refers to the degree of membership of the attribute based on the linguistic hedges present in the query.

### 3.3 Computation of Membership and Manipulated Membership Value

Membership Value Assignments: To find the membership value we divide the fuzzy set into three regions [10]. These are Lower Boundary region, Core region and Upper boundary region. The membership value for the core region is always equal to 1. The membership value for the Lower Boundary and Upper Boundary are calculated as  $(x - \alpha)/(\beta - \alpha)$  and  $(\delta - x)/(\delta - \gamma)$  respectively where x is the value of the attribute that can be brought from the concerned table.

Once the membership value is computed the next step is to find the Manipulated Membership Values for corresponding attribute. If the fuzzy query contains linguistic hedges [10] then we need to compute the appropriate membership value depending on the linguistic hedges. This value is called Manipulated membership value. By knowing the corresponding membership function, we can find the manipulated membership values. As an example, let us look at the basic linguistic atom,  $\alpha$  and subject it to some hedges. Define  $\alpha = \int_Y \mu_\alpha(Y)/Y$ , then

- i. Very  $\alpha = \alpha^2$
- ii. Very Very  $\alpha = \alpha^4$
- iii. Plus  $\alpha = \alpha^{1.25}$
- iv. Slightly  $\alpha = \sqrt{\alpha}$
- v. Minus  $\alpha = \alpha^{0.75}$

### 3.4 Fuzzy to Crisp Conversion

The defuzzification of a fuzzy set is the process of rounding it off from its location in the unit hypercube to the nearest vertex. Let us consider a fuzzy set  $A$ . The lambda cut set [10] is defined as  $A_\lambda$ , where  $0 \leq \lambda \leq 1$ . The set  $A_\lambda$  is a crisp set called the lambda ( $\lambda$ ) cut set or Alpha ( $\alpha$ ) cut set of the fuzzy set A where  $A_\lambda = \{x | \mu_A(x) \geq \lambda\}$ . Infinite number of lambda cut set can be made from a fuzzy set  $A$ . After completing the manipulated membership values, the next step is to derive lambda cut at a threshold value 0.5. This threshold value can be varied depending on the situation. After finding the lambda cut, minimum and maximum range for the corresponding attributes are found.

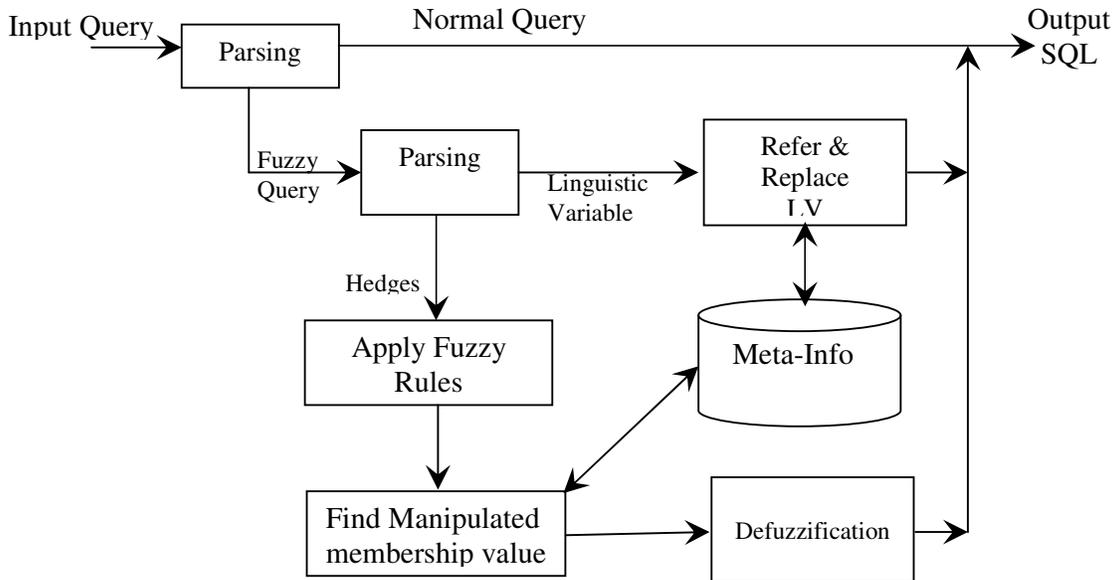


Figure 2: Block Diagram of the Query Implementation Stage

### 3.5 Query Implementation Stage

The Query implementation stage is illustrated in Figure 2. In the first step the query is checked for its type i.e whether it is a Selection query or Deletion query or Update query. In case of Insert query no action is needed. In the second step, the input query is checked for the presence of linguistic variables. If it contains any linguistic terms, the next step will carry out a check for the presence of linguistic hedges. In case, if it contains linguistic hedges the membership value is calculated and it is manipulated based on the Hedges. Once this operation is carried out, the fuzziness is removed by applying the defuzzification methods. The defuzzification is carried out by finding the  $\alpha$ -cut and by calculating the maximum and minimum range. Once the minimum and maximum range is calculated then the fuzzy terms with linguistic hedges are removed by using these values.

### 4 Experimental Results and Snap Shot of Database

Some sample queries have been tested and the evaluated results are produced in this section.

Table 2: Meta-Information Table

LinguisticTerm	Alpha	Beta	Delta	Gamma	Tablename	columnname
Aged	60	62	80	82	Student	age
Best	55	60	75	80	student	Percentage
Middle	30	32	40	42	Student	Age
Old	40	42	60	62	Student	age
Young	18	20	30	32	Student	age

Table 3. Student Information Table

Name	Age	Percentage
Anna	22.2	80.00
Bala	25.4	85.00
Charlie	20.5	75.00
Ela	19.6	87.00
Francis	30.8	88.00
Gopal	18.7	65.00
Inba Tamilan	16.3	66.00
Janakiraman	37.6	99.00
Kalidas	35.7	45.00
Narayan	48.7	19.00
Om Prakash	51.4	24.00
Prasad	55.8	50.00
Quirola	58.4	55.00
Raman	60.0	56.00
Baby	30.0	55.55
Titus	43.3	88.40

The information from which the below results were obtained is given in Table.2 & Table.3. When an input query “select \* from student where age = young” is applied, the linguistic term “young” is replaced with the range of values 18 and 32. So the result will be the query “select \* from student where age between 18 and 32”. When an input query “select \* from student where age Very Young” is applied, the linguistic term “young” with hedge “Very” is replaced with the range of values 19.6 and 30.0. So the result will be the query “select \* from student where age between 19.6 and 30.0”.

## 5 Conclusions

The proposed framework is successfully implemented and the translation of fuzzy query into SQL in relational databases is carried out. The fuzziness is in the form of approximate values or linguistic variables, which can be applied only in queries. Though the fuzziness can be incorporated by storing the fuzzy value inside the database, it may not be the efficient method in the real time. Fuzzy databases are still not popular among the people because they are reluctant to replace their crisp data by fuzzy data before they are convinced. Therefore, the proposed framework scores over the second approach as it can be used with crisp data and also it is making use of the power of fuzzy set theory.

## Acknowledgements

I would like to thank Sadhguru Jaggi Vasudev for leading me in every walk of my life. I am grateful to Prof.K.Senthamarai Kannan for his valuable guidance and support in my research activities. I thank my family members for their cooperation. Thanks are extended to Dr. Cletus Babu, Dr.K.Arulmozhi, Dr.K.Sadasivam, Prof. Parthiban and the staff members of the Department of Statistics, Manonmaniam Sundaranar University, Tirunelveli for their moral support and motivation.

## Reference:

- [1] Bolloju.N. Modelling of Imprecise and Uncertain Information, in Prakash, N.(Ed.) Current Trends in Management of Data, Tata - McGraw Hill: New Delhi, 1990.
- [2] Bosc.P. and Pivert.O. and Farquhar.K. Integrating Fuzzy Queries into an Existing Database Management System: An Example, International Journal of Intelligent Systems, 1994.
- [3] Gogolla.M. and Hohenstein.U. Towards a Semantic View of an Extended Entity-Relationship Model, ACM Transactions on Database Systems, 16, 369-416, 1991.
- [4] Lee.D., Kim.M.H., Lee-Kwang.H., and Lee.Y.H. A Fuzzyfication of the Relational Data Model in: Moon,S.and Ikeda H. (Eds) Database Systems for Advanced Applications '93, World Scientific: Singapore, 360- 367, 1993.
- [5] Lee.D.H, and Kim.M.H., Accommodating Subjective Vagueness Through a Fuzzy Extension to the Relational Data Model, Information Systems, 18, 6, 363-374, 1993.
- [6] Lee.D.H, and Kim.M.H., Extending Semantics of Relational Operators for Vague Queries, Microprocessing and Micro- programming, 39, 165-168, 1993.
- [7] Nan-Chen Hsieh, Answers to Database Queries Concerning Imprecise Information in Logical Fuzzy Relational Databases, Tamkang Journal of Science & Engineering, Vol. 7, No 3.pp.149 -150, 2004.
- [8] Raju, K.V.S.V.N. and Majumdar, A.K. Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database System, ACM Transactions on Database Systems, 13, 2, 129-166, 1988.
- [9] Takahashi, Y. Fuzzy Database Query Languages and Their Relational Completeness Theorem, IEEE Transactions on Knowledge and Data Engineering, 5, 1,122-125, 1993

- [10] Timothy J. Ross Fuzzy Logic with Engineering Applications, McGraw-Hill Book Co – Singapore, 1997.
- [11] Villa, M. A., Cubero, J. C., Medina, J. M. and Pons, O. A Logic Approach to Fuzzy Relational Databases, International Journal of Intelligent Systems, 9, 449-460, 1994.
- [12] Wilma Penzo, Rewriting Rules to Permeate Complex Similarity and Fuzzy Queries within a Relational Database System, IEEE Transactions on Knowledge and Data Engineering, VOL 17, NO 2, 255-270, February 2005.

## Authors



**Kaliaperumal Senthamarai Kannan** is currently working as a Professor & Head, in the department of Statistics, at Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India. He has more than 18 years of teaching experience a post-graduate level. He has published more than 30 research papers in international and national journals. He authored four books and visited Turkey, Singapore and Malaysia. He has been awarded TNCST Young Scientist Fellowship and SERC Visiting Fellowship. His area of specialization is ‘Stochastic Process and their Applications’. His other research interests include stochastic modeling in the analysis of birth intervals in human fertility bio-informatics, data mining and precipitation analysis.



**Vanniappan Balamurugan** is an Assistant Professor & Head in the Department of Computer Science and Engineering, SCAD College of Engineering & Technology, Cheranmahadevi Tirunelveli, India. He has received his M.Tech in Computer and Information Technology from Manonmaniam Sundaranar University, Tirunelveli, India. Presently he is working on Ph.D under the guidance of Prof.K.Senthamaraikannan, Manonmaniam Sundaranar University, Tirunelveli. He possesses 22 years of experience, out of which 16 years in programming and 6 years in teaching. His research interests include Time Series Data Mining, Content Based Retrieval, Similarity Search and Image Processing.