Global Journal of Computer Science and Technology
Volume 12 Issue 6 Version 1.0 March 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Querying Capability Enhancement in Database Using Fuzzy Logic

By Amit Garg & Dr. Rahul Rishi
Maharishi Dayanand University, Rohtak

Abstract - We already know that Structured Query Language (SQL) is a very powerful tool. It handles data, which is crisp and precise in nature.but it is unable to satisfy the needs for data which is uncertain, imprecise, inapplicable and vague in nature. The goal of this work is to use Fuzzy techniques i.e linguistic expressions and degrees of truth whose result are presented in this paper. For this purpose we have developed the fuzzy generalized logical condition for the WHERE part of SQL. In this way, fuzzy queries are accessing relational databases in the same way as with SQL. These queries with linguistic hedges are converted into Crisp Query, by deploying an application layer over the Structured Query Language.

Keywords : *Fuzzy Query, Linguistic variable, Membership value, Fuzzy Logic, Fuzzy Sets, Fuzzy Relational Databases, Fuzzy SQL*

GJCST Classification: *I.5.1*

Strictly as per the compliance and regulations of:

Querying Capability Enhancement in Database Using Fuzzy Logic

Amit Garg [α] & Dr. Rahul Rishi [σ]

Abstract - We already know that Structured Query Language (SQL) is a very powerful tool. It handles data, which is crisp and precise in nature.but it is unable to satisfy the needs for data which is uncertain, imprecise, inapplicable and vague in nature. The goal of this work is to use Fuzzy techniques i.e linguistic expressions and degrees of truth whose result are presented in this paper. For this purpose we have developed the fuzzy generalized logical condition for the WHERE part of SQL. In this way, fuzzy queries are accessing relational databases in the same way as with SQL. These queries with linguistic hedges are converted into Crisp Query, by deploying an application layer over the Structured Query Language.

Keywords : Fuzzy Query, Linguistic variable, Membership value, Fuzzy Logic, Fuzzy Sets, Fuzzy Relational Databases, Fuzzy SQL.

I. Introduction

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 develop a front end that will allow fuzzy querying to the database. Here the underlying database will always be crisp. This paper is organized as follows: Fuzziness in database is presented in section 2. Section 3 explains fuzzy logic concepts. Section 4 analyses SQL limitations. Section 5 discusses the proposed framework for FSQL.Section 6 gives the implementation detail of fuzzy querying and in next section conclusion and future scopes are drawn

Author α : Indus Institute of Engg. & Tech., Kinana (Jind).
E-mail : amit.indus86@gmail.com
Author σ : Maharishi Dayanand University, Rohtak .
E-mail : rahulrishi@rediffmail.com

II. Fuzziness In Database

A database is an ordered collection of related data elements intended to meet the information needs of an organization and designed to be shared by multiple users.

If a regular or classical database is a structured collection of records or data stored in a computer, a fuzzy database is a database, which is able to deal with uncertain or incomplete information using fuzzy logic. Basically, a fuzzy database is a database with fuzzy attributes, which may be defined as attributes of an item, row, or object in a database, which allows storing fuzzy information.. The following is a brief definition of the characteristics of imperfect data:

- *Uncertain data* - The uncertainty is related to the degree of truth of its attribute value, and it means that we can apportion some, but not all, of our belief to a given value or a group of values.
- *Vague data* - Lack of definite or sharp distinctions.
- *Imprecise data* - The imprecision and vagueness are relevant to the content of an attribute value, and it means that a choice must be made from a given range (interval or set) of values but we do not know exactly which one to choose at present [8].
- *Inapplicable data* - There may be some entities for which a piece of data relating to one of its properties cannot be acquired due to a lack of the property.

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 full names of the Young terrorists who were involved in the recent bomb blast that occurred in the region in and around GUJRAAT". 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.

III. Fuzzy Logic: Fuzzy sets and Basic Concepts

Fuzzy set theory is the base of fuzzy logic. In this logic, the truth-value of a sentence (or satisfaction degree) is in the real interval [0, 1]. The value 0 represents completely false, and 1 is completely true. The truth-value of a sentence "s" will be denoted as $\mu(s)$.Fuzzy logic was developed as a mean to do reasoning under uncertainty. Many new approaches and theories treating imprecision and uncertainty have been proposed since fuzzy set was introduced by Zadeh [4].

Its characteristics are

1. Fuzzy truth values are expressed in linguistic terms.
2. Imprecise truth tables.

Fuzzy Logic is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster. Fuzziness can be defined as the vagueness concerning the semantic meaning of events, phenomenon or statements themselves. It is particularly frequent in all areas in which human judgment, evaluation and decisions are important [10].

A fuzzy set is almost any condition for which we have words: short men, tall women, hot day, cold climate, new building, ripe bananas, high intelligence, low speed, overweight, etc., where the condition can be given a value between 0 and 1. Fuzzy set 'A' over a universe of discourse X (a finite or infinite interval) within which the fuzzy set can take a value) is a set of pairs:

$$A = \{\mu A(x) / x : x \in X, \mu A(x) \in [0, 1] \in \Re\}$$

Where, $\mu A(x)$ is called the membership degree of the element x to the fuzzy set A. This degree ranges between the extremes 0 and 1 of the dominion of the real numbers: $\mu A(x) = 0$ indicates that x in no way belongs to the fuzzy set A, and $\mu A(x) = 1$ indicates that x completely belongs to the fuzzy set A. Note that $\mu A(x) = 0.5$ is the greatest uncertainty point [5].

a) Fuzzy Set Operators

For crisp sets, the basic operations are, namely,

- Union, OR
- Intersection, AND
- Complement, NOT

As an analogy, for fuzzy sets we define fuzzy operators that allow us to manipulate the fuzzy sets. We similarly have fuzzy complements, intersection and union operators but they are not uniquely defined i.e. as membership functions, they are also context – dependent [5].However an important dissimilarity exists there between traditional set / logic and fuzzy set theory. Traditionally there is a distinction between a union operation of sets and OR of logic as is the case with intersection and AND also. But in fuzzy theory there is no such distinction between the logical and set Operators

Fuzzy sets allow operations of union, intersection, and complement. These operations can be used when linguistic hedges, such as "very" or "not very," are used [6].

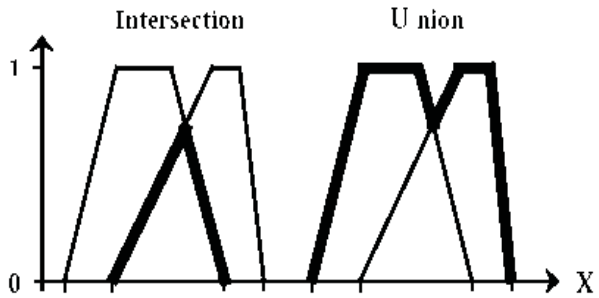

Figure 3.1: Intersection (minimum) and union (maximum)

Fuzzy union = Fuzzy OR
Fuzzy intersection = Fuzzy AND
Fuzzy complement = Fuzzy NOT
We define some standard fuzzy operations as:

- **Fuzzy Complement,**
  $$\sim A(x) = 1 - A(x)$$
- **Fuzzy Union,**
  $$(A \cup B)(x) = \max [A(x), B(x)]$$
- **Fuzzy Intersection,**
  $$(A \cap B)(x) = \min [A(x), B(x)]$$

b) 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

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.

Example : The "Temperature" is a linguistic variable. We can define four linguistic labels, like "Very_Cold," "Cold," "Hot," and "Very_Hot," using the membership functions depicted in the diagram as:

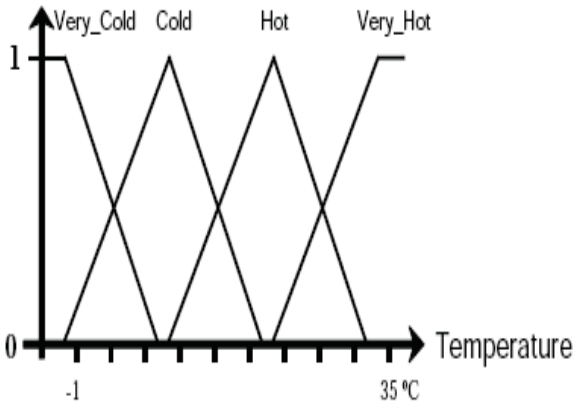

Fig 3.2: Frame of Cognition with four linguistic labels for temperature

IV. SQL and Its Limitation

Users search databases in order to obtain data needed for analysis, decision making or to satisfy their curiosity. The SQL is a standard query language for relational databases. The simply SQL query is as follows:

```
select attribute_1,…,attribute_n
from T
where attribute_p > P and attribute_r < R
```

The result of the query is shown in graphical mode in figure4 1. Values P and R delimit the space of interesting data. Small squares in the graph show database records. In the graph it is obviously shown that three records are very close to meet the query criterion. These records could be potential customers and direct marketing could attract them or municipalities which almost meet the criterion for some financial support for example

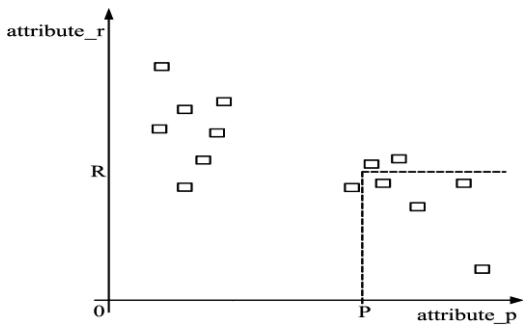

Fig. 4.1: The result of the classical query

The SQL uses the crisp logic in querying process that causes crisp selection. It means that the record would have not been selected even if it is extremely close to the intent of the query criterion. As the criterion becomes more and more complex, the set of records selected by the WHERE statement becomes more and more crisp. If the classical SQL is used for solving this problem, the SQL relaxation would have to be used in the following way:

```
select attribute_1,…,attribute_n
from T
where ateribute_p > P-p and attribute_r < R+r
```

where p and r are used to expand the initial query criteria to select records that almost meet the query criteria. This approach has two disadvantages [3].

First, the meaning of the initial query is diluted in order to capture adjacent records. The meaning of a query: "where attribute_p is more than P" is changed and adjacent records satisfy a query in the same way as initial ones. More precisely, the difference between original and adjacent data (caught records along the "edge" of interesting space) does not exist. Secondly problem rises from the question: what about records that are very close to satisfy the new expanded query and it is useful to make another expanding of a query. In this way more data from the database is selected, but the user has lost the accuracy of his query.

V. Proposed Framework

An application layer is placed over the SQL and it wil 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.

a) FSQL Architecture

SQL is the most influential commercially marketed database query language. It uses a combination of relational algebra and relational calculus constructs to retrieve desired data from a database. FSQL is SQL that can handle fuzzy attribute values. The main difference between SQL and FSQL is that SQL returns a subset of the database as the query result. When attributes with fuzzy values appear in the query, it is transformed into a query that can be handled by SQL and finally results obtained from the SQL query are then post processed in order to obtain the desired information.

The components of FSQL architecture are as:

- **Traditional Database** : They are data from our relations with a special format to store the fuzzy attribute values. The fuzzy attributes are classified

by the system in different data types as we explained above.

- **Fuzzy Meta Knowledge Base (FMB) :** It stores information about the fuzzy relational database (FRDB) in a relational format. It stores attributes which admit fuzzy treatment and different information for each one of them, depending on their fuzzy type.
- **FSQL Server :** It has been programmed entirely in SQL in Oracle PL/SQL (Galindo, 1999) and PostgreSQL (Maraboli & Abarzua, 2006) and it includes three kinds of functions:
- **Translation Function (FSQL2SQL) :** It carries out a lexical, syntactic and semantic analysis of the FSQL query and translates it into a classic SQL sentence. The resulting SQL sentence includes reference to the following kinds of functions.
- **Representation Functions** : used to show the fuzzy attributes in a comprehensible way for the user and not in the internally used format.
- **Fuzzy Comparison Functions :** used to compare the fuzzy values and to calculate the compatibility degrees (CDEG function).
- **FSQL Client :** It is a simple and independent program that serves as an interface between the user and the FSQL Server. The user introduces the FSQL query and the client program communicates with the server and the database in order to obtain the final results.

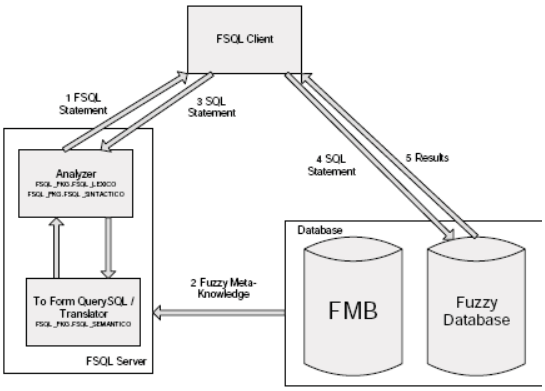

Figure 5.1: FSQL Implementation architecture

VI. Fuzzy Querying

"Fuzzy querying is similar to the process of ordinary querying, but more complexes. ".
Classical relational databases suffer from a lack of flexibility in query. The given selection condition and the contents of the relations are all crisp. A query is flexible if the following conditions can be satisfied

1. A qualitative distinction between the selected tuples is allowed.
2. Imprecise conditions inside queries are introduced when the user cannot define his/her needs in a definite way, or when a pre specified number of responses is desired and therefore a margin is allowed to interpret the query.

The crucial difference between fuzzy queries and exact queries is the number of records brought into the memory. A large number of tuples will be selected by fuzzy condition in comparison to the crisp one. If a record consists of a fuzzy attribute, say height, a query such as retrieve all tall people" will cause a considerable portion of the database being brought into the memory. Fuzzy querying allows one to express vague predicates represented by fuzzy sets. Therefore, access paths of the existing index structures cannot be used directly since fuzzy has also other differences from crisp querying. One of the distinguishing features of fuzzy querying is the concept of a matching degree belonging to the [0, 1] interval. The fuzzy query evaluation against a crisp database may be considered as a special case of a more general and complex case of fuzzy, possibility based databases. In the latter case we deal with imperfect information both in the query and in the database. Namely, the query may contain linguistic terms represented by the fuzzy sets and the values of attributes in the database may be represented by possibility distributions. Then, a simple query condition may be expressed as the requirement that a numeric attribute value, represented by the possibility distribution ¼(u), matches a soft constraint represented by the fuzzy set P. The matching degree is evaluated using two measures:

$$\text{Possibility of matching: } \Pi(P) = \sup_{u \in U} \min(\pi(u), \mu_P(u))$$

$$\text{Necessity of matching: } N(P) = \inf_{u \in U} \max(1-\pi(u), \mu_P(u))$$

In case of a crisp database the possibility distribution is replaced by a single value u0 what corresponds to the following possibility distribution: $\pi(u0) = 1$ and $\pi(u) = 0$ V $u = u0$ Then, both (1) and (2) reduce to $\Pi(P) = N(P) = \mu P(u0)$.

In general, we have two feasible ways to incorporate fuzziness in databases :

1. Making fuzzy queries to the classical databases
2. Adding fuzzy information to the system: The tables that are required are, Meta_Information Table, 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 Linguistic_Hedges table contains the linguistic hedges 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 column Membership Value refers to the degree of membership of the attribute in 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 [7].

. Let trapezoidal map of AGE (young) for an instance Is as:

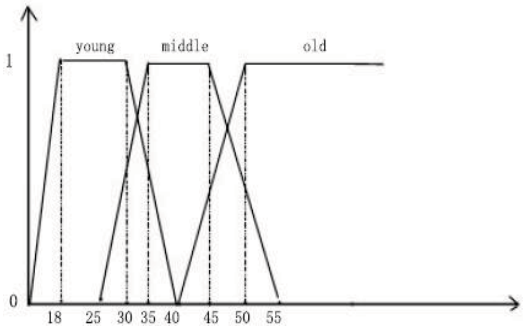

Fig 6.2: Trapezoidal Function of AGE

Membership function of AGE (young, middle, old) is defined as following:

$$\mu_A(\text{young}, x) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \le x < b \\ 1 & \text{if } b \le x < c \\ (d-x)/(d-c) & \text{if } c \le x < d \\ 0 & \text{if } x \ge d \end{cases}$$

Here, a = 0; b = 18; c = 30; d = 40

$$\mu_A(\text{middle}, x) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \le x < b \\ 1 & \text{if } b \le x < c \\ (d-x)/(d-c) & \text{if } c \le x < d \\ 0 & \text{if } x \ge d \end{cases}$$

Here, a = 25; b = 35; c = 45; d = 55

$$\mu_A(\text{old}, x) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \le x < b \\ 1 & \text{if } x > b \end{cases}$$

Here, a = 40; b = 50

Let trapezoidal map of SALARY for an instanceis as:

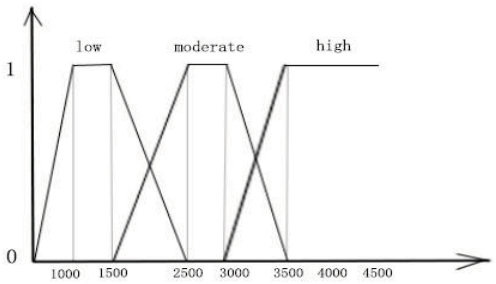

Figure 6.3 : Trapezoidal Function of Salary

Membership function of SALARY (low, moderate, high) is defined as following:

$$\mu_s(\text{low}, x) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \le x < b \\ 1 & \text{if } b \le x < c \\ (d-x)/(d-c) & \text{if } c \le x < d \\ 0 & \text{if } x \ge d \end{cases}$$

Here, a = 0; b = 1000; c = 1500; d = 2500

$$\mu_s(\text{moderate}, x) = \begin{cases} 0, & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \le x < b \\ 1 & \text{if } b \le x < c \\ (d-x)/(d-c) & \text{if } c \le x < d \\ 0 & \text{if } x \ge d. \end{cases}$$

Here, a = 1500; b = 2500; c = 3000; d = 3500

$$\mu_s(\text{high}, x) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \le x < b, \\ 1 & \text{if } x > b \end{cases}$$

Here, a = 3000; b = 3500

In FSQL linguistic terms may appear as fuzzy values, relations, and quantifiers (associated with aggregation operators) in the WHERE clause and other clauses.. This is called by Bosc et al. the vertical quantification in contrast to the horizontal quantification when a quantifier plays the role of an aggregation operator and replaces the AND or OR connectives in a condition. All the operations of the relational algebra (implicitly or explicitly used in SQL's SELECT instruction) are redefined to properly process fuzzy relations that

appear when parts of a fuzzy query are processed. The syntax of FSQL query.

SELECT atributeList
FROM tableNameList
[WHERE conditionList]
[GROUP BY atributeList
HAVING conditionList]
[THRESHOLD number]

Query (extremely old, salary is a little high). The ideal SQL commands could be:

SELECT * FROM queryanalyzer.mdb;

WHERE age="extremely old" AND salary="a little high"

But in fact, it is could not be realized directly. Therefore, it must be transformed equivalence precise conditions linked with the fuzzy membership function values. The first step is calculating individual membership value; the second is by the formula For example the query:

SELECT id, name, department, age, salary FROM;

employee, agemember, salarymember WHERE;

employee.id = agemember.id and;

employee.id = salarymember.id and;

agemember.old >= 0.5+z and;

salarymember.high >= 0.5+z;

The value of variable: z could be adjusted automatically through the selection of the object: combobox. The codes of this part are about:

The user's query is finished by input query words or α-threshold into the text boxes; the fuzzy operator is by clicking the combo boxes. The programmer could judge from these inputs.

do case

case s="extremely", z=0.4.

case s="very", z=0.2.

case s="a little", z=-0.2

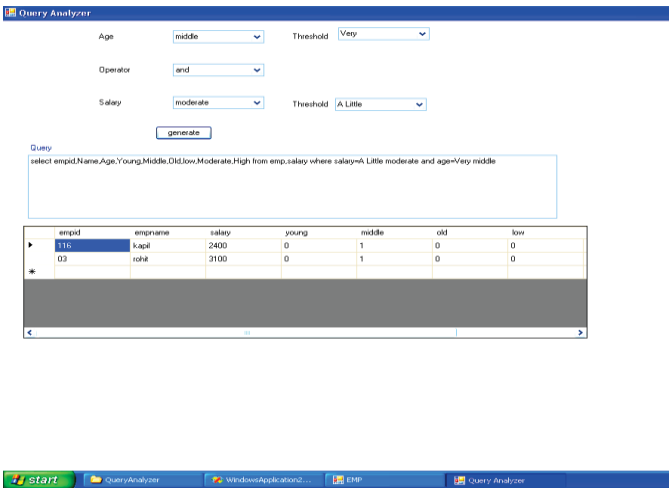

Fig.6.4 : Implementation of FSQL

VII. Conclusion and Future Trends

This paper presents an introduction to fuzzy logic and fuzzy databases. The fuzzy SQL is in this approach an independent module and it can be used when the user wants to use a linguistic expression in queries. 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. Various fuzzy database models, including relational and object-oriented databases, have been proposed over the past thirty years and tremendous gain is hereby accomplished

References Références Referencias

1. Aleksandar Takači, Srđan Škrbić ; How to Implement FSQL and Priority Queries.
2. Bosc P., Galibourg M., Hamon G. (1988): "Fuzzy Querying with SQL: Extensions and Implementation Aspects". Fuzzy sets and systems, Bosc P., Pivert O. (1995): "SQLf: A Relational Database Language for Fuzzy Querying". IEEE Transactions on Fuzzy Systems.
3. Galindo, J., Urrutia, A., Piattini, M., Fuzzy Databases: Modeling, Design and Implementation, Idea Group Publishing, Hershey, USA. (2006)
4. Klir & Yuan. Fuzzy Logic & Fuzzy sets.
5. L.A. Zadeh, "Fuzzy Sets", *Inf. Contr.,* 8, pp. 338- 353, 1965.
6. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, Part 1: 8:199-249; Part 2: 2:301-357; Part 3: 9:43-80, 1975.
7. 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.
8. M. Black, "Vagueness", Philosophy of Science, 4, pp. 427-455, (1937).
9. P. C. Saxena and B. K. Tyagi, "Fuzzy functional dependencies and independencies in extended fuzzy relational database models," Fuzzy Sets and Systems, Vol. 69, 1995, pp. 65-89.
10. Timothi J.Ross Fuzzy Logic with Engineering Applications, McGraw-Hill Book Co – Singapore,1997.
11. Z. M. M and LI YAN, "A Literature Overview of Fuzzy Database Models*", Journal Of Information Science And Engineering 24, 189-202 (2008).

12. Zadeh L. A. “A new direction in AI : Toward a computationa ltheory ofperceptions”, Technologies for Constructing Intelligent Systems I,Physica-Verlag Heidelberg New York, pp 3-20 (2002)
13. Zimmerman J. [2001], “Fuzzy Set Theory – And It’s Applications”, KluwerAcademic Publishers, Norwell, Massachusetts, U.S.A.

This page is intentionally left blank