RESEARCH ARTICLE OPEN ACCESS

Fuzzy Database Data Models and Its Framework

Ms. Poonam
Banasthali Vidyapith, Rajasthan
India

ABSTRACT

In today's information-oriented society, the importance of accumulating and accessing information efficiently has become widely accepted. In response to this need, many database systems have been developed. A major limitation of these systems, however, is that many kinds of imprecise data exist which are not easily processed. To address this problem, the concept of a fuzzy database system has been developed. This paper mainly focuses on the Fuzzy database model, framework and Sql Evaluation

Keywords:- Fuzzy Relational Model, Database Model, Framework, SQL

I. INTRODUCTION

A database-management system (DBMS) consists of a collection of interrelated data and a set of programs to access those data. The primary goal of a DBMS is to provide an environment that is both convenient and efficient for people to use in retrieving and storing information. A number of operations are performed on a DBMS. Searching is an important operation among those. A significant amount of time is needed for searching data from a DBMS. As the size of a DB increases, the searching time also increases. A number of algorithms have been developed to improve the performance of searching using query. But those algorithms have been developed only for classical DB. The traditional DBMS cannot manipulate incomplete, imprecise and vague data such as very high, about 30, etc. properly. To overcome this problem, FDBMS (Fuzzy Database Management System) has been introduced. The primary focus of fuzzy logic is on natural language, where reasoning with imprecise propositions approximates is rather typical. As the size of DB is increasing day by day, programmer are intending to reduce the time complexity to access data from a large database [12]. The performance of a query is influenced by the structure of data and size of a DB. Large database may consist of millions of data and it costs significant amount of time to find any particular record from that database. The search time may be reduced by indexing database through the B-tree algorithm.

Databases are used to store lots of data and retrieve needed data from them efficiently. In databases of the real world, however, we encounter several kinds of fuzziness, that is, data themselves, e.g.

"Taro is ***young,"*** association between data, e.g., "Taro is young is ***more OT less true"*** and words in queries, e.g., "Get names of persons who are ***young*** and ***rich."*** We demonstrate a fuzzy database system of possibility-distribution fuzzy-relational model proposed by authors that covers these three kinds of fuzziness. And we demonstrate two data manipulation languages based on fuzzy relational algebra and SQL (Structured Query Language), which are standard in ordinary database systems for theoretical issue and practical use, respectively.

II. A FUZZY DATABASE MODEL

A fuzzy database is defined as an enhanced relational database that allows fuzzy attribute values and fuzzy truth values; both of these are expressed as fuzzy sets.

For representing fuzzy data, we propose a possibility-***distribution-fuzzy-relational model*** (pd-fr model, for short), in which fuzzy data are represented by fuzzy relations whose grades of membership and attribute values can be possibility distributions, where the fuzziness of an attribute value is represented by a possibility distribution and its association by a grade of membership. This model is an extension of the relational model of data by Codd [l]. We shall have a fuzzy relation RA whose attribute and membership values are possibility distribution

(Create RA <NAME, AGE> Cl/<Jack, 27>, 0.6/<John, 35>, {1/O.8, 0.6/0.7}/<Judy, <0.6/23, 1/24)>, 0.8/<Mike, {0.6/47, 1/48, 0.5/49}>, {l / l, 0.7/0.9)/<Richard, about25>})

where the function **creates** defines a relation schema and creates an instance of fuzzy relation with setting entities. We have possibility distributions **as** attribute values such **as** (0.6/23, 1/24} and (0.6/47, 1/48, 0.149) and **as** membership values such **as** (1/0.8, 0.6/0.7} and {1/1,0.7/0.9}. We have a name of possibility distributions such **as about25,** which have been defined **as (term (about 25 {0.6/24, 1/25,**

0.6/26}))

2.1 Fuzzy Data Model

A fuzzy database consists of relations: a is a relation relation $R(t_1........,t_n)$ a Cartesian product P_1 xP_2 x ... domains x P_n, of *Pi;* each *P,* is a set of fuzzy sets *t,* over an attribute domain D_i,*(1*<=i<=n ***).*** It is assumed that key attributes take ordinary nonfuzzy values. For the notational convenience, fuzzy sets are identified with their representative membership functions; for example, *t;* also denotes a membership function.

2.2 *Fuzzy Attribute Value*

Attribute values such as age have nonfuzzy values such as fuzzy predicates such as 20 in relational database; attribute values are defined as fuzzy predicate such as "young" and "about forty" in the fuzzy database. For example, a fuzzy attribute value of "age of Dr. *x* is is expressed as a possibility distribution

p (age of *x)* = **YOUNG; YOUNG** denotes a **fuzzy** set that represent the fuzzy predicate "young". Thus attribute values are identified with fuzzy sets such as YOUNG.

2.2 Fuzzy Truth Values

Truth values of any tuples are either 1(=true) or 0(=false) in the relational database; truth values of any tuples are defined as fuzzy predicates such as **"0.7"** and "completely true" in the fuzzy database. Consider, for example, a tuple *t* that asserts fuzzy proposition: "It is completely true that Dr. X is very much older than twenty."The truth values of t is expressed as a possibility distribution P[T(t)]=N;T(t) denotes a truth value of t and N denotes a fuzzy set that represents the fuzzy predicate "completely true." Thus the truth values T(t) are identified with fuzzy sets such as N over z ϵ[0,1];the values z ϵ [0,1] has the following meaning.

1)z=0 means the tuple is completely false

2)0<z<1 means that the tuple t is true to degree expressed by the real number z

3)z=1 means that the tuple t is completely true

D. Fuzzy Set theory

Fuzzy set theory provides a formal mathematical framework for a systematic treatment of fuzzy data. **A** fuzzy subset **A** of X is defined as:

A = { (m_A(x)/x) I x ϵ X} and m_A(x) → [0,1] where mA(x) is the membership function.

2.3 THE GEFRED MODEL

The GEFRED (Generalized Fuzzy Relational Database) model was published in 1994 by Medina-Pons-Vila[11]. It is developed in possibilistic framework , so fuzzy domains are considered. It also includes the case where the underlying domain is not fuzzy i.e. numeric. Various data types given in GEFRED are:

1. A single scalar (e.g., Age = Young, represented by the possibility distribution 1/Young).
2. A single number (e.g., Height = 160, represented by the possibility distribution 1/160).
3. A set of mutually exclusive possible scalars (e.g., Age = {Young, Old}, represented by {1/Young, 1/Old}).
4. A set of mutually exclusive possible numbers (e.g., Age = {14, 50}, represented by {1/14, 1/50}).
5. A possibility distribution in a scalar domain (e.g., Age = {0.6/Young, 1.0/Middle}).
6. A possibility distribution in a numeric domain (e.g., Age = {0.5/23, 1.0/26, 0.8/24}, fuzzy numbers or linguistic labels). It includes Umano-Fukami models data types UNKNOWN ,UNDEFINED and NULL also.
7. An Unknown value with possibility distribution: Unknown = {1/x : x ∈ D}
8. An Undefined value with possibility distribution: Undefined = {0/x : x ∈ D}
9. A NULL value given by: NULL = {1/Unknown,1/Undefined}

As GEFRED suggests, Generalized fuzzy domain (D) and generalized fuzzy relations (□)) concepts are being included. Generalized fuzzy domain (D) includes classic domain, with the possibility distributions defined for this domain and the NULL value also. If X is the universe of discourse and P (X) is set of all possibility distributions including unknown, undefined types and Null type. Then, generalized fuzzy domain (D) can be represented as D □ P(X)□□NULL.

Generalized fuzzy relations are relations whose attributes are having generalized fuzzy domain. Fuzzy attributes may have associated with compatibility attribute where compatible degree can be stored. Generalized fuzzy relations are given by two sets: Head H and Body B. The head includes the name of each one of the attributes, their domains, and their compatibility attributes (which are optional). The body includes the values of the m tuples:

$$R = \begin{cases} H= \{(A1:D1[,C1], ..., An [,Cn])\} \\ B = \{(A1:di1[,ci1], ..., An:din[,cin])\} \ i= 1,..., m \end{cases}$$

GEFRED model also defines fuzzy comparators which are based on the classic comparators (<, >, =etc). When these fuzzy comparators are used on non-fuzzy values, meanings of comparators does not change i.e. classical output should come during crisp values.

2.4 Representation of Fuzzy Attributes

This representation is different according to the fuzzy attributes. Fuzzy attributes of type 1 are represented as usual attributes, because they do not allow fuzzy values. Fuzzy attributes type 2 need five classic attributes as shown in table 3.

Table 1 Kind of values of fuzzy attributes type 2

Kind of values	Attributes in the DB for each fuzzy attribute Type 2				
	FT	F1	F2	F3	F4
UNKNOWN	0	NULL	NULL	NULL	NULL
UNDEFINED	1	NULL	NULL	NULL	NULL
NULL	2	NULL	NULL	NULL	NULL
CRISP	3	d	NULL	NULL	NULL
LABEL	4	FUZZY_ID	NULL	NULL	NULL
INTERVAL	5	n	NULL	NULL	m
APPROXIMATELY	6	d	d-marge	d+marge	marge
TRAPEZE [$\alpha,\beta,\gamma,\delta$]	7	α	β	γ	δ
APPROXIMATE VALUE $d \pm m$	8	d	$d - m$	$d + m$	m
POSSIBILITY DISTRIBUTION-2	9	p1	d1	p2	d2
POSSIBILITY DISTRIBUTION-4	10	d1	d2	d3	d4

• FT: stores the kind of value which the attribute in question can take (0 for UNKNOWN, 1 for UNDEFINED, *etc*). The letter T is concatenated the name of the attribute.

• F1, F2, F3 et F4 : stores the description of the parameters which define the data and which depend on the type of value (FT), the name of these attributes .

Fuzzy Attribute Type 3(FTYPE3):They are attribute over "data of discrete non-ordered dominion with analogy". In these attributes some labels are defamed("blond","red","brown",etc)that are scalars with a similarity(or proximity)relationship defined over them, so that this relationship indicates to what extent each pair of labels be similar to each other. The fuzzy attribute type 3 is represented by a variable number of traditional attributes according to the form described in Table below--

- FT: is similar to FT used in FTYPE2 attribute.

Table 2. Kind of values of fuzzy attributes type 3

- (FP1, F1),…, (FPn,Fn): in these attributes, we store data of the distribution of possibility. For example, in a value of the SIMPLE type, only first couple is used and value of possibility will be 1 (to be standardized).

Kind of Values	FT	FP1	F1		FPn	Fn
UNKNOWN	0	NULL	NULL	...	NULL	NULL
UNDEFINED	1	NULL	NULL	...	NULL	NULL
NULL	2	NULL	NULL	...	NULL	NULL
SIMPLE	3	P	d	…	NULL	NULL
POSS. DISTR.	4	P1	d1	…	pn	dn

Fuzzy Attributes Type 4 (FTYPE4): These attributes are defined in the same way as Type 3 attributes, without it being necessary for a similarity relationship to exist between the labels.

III. FRAMEWORK FOR FUZZY DATABASE

Database systems have evolved over a number of years. In this section, a review on the development of the DBMS is given. It is hoped that this review will help to elucidate the limitations and advantages of the DBMS at each development phase. DBMS's have undergone four major phases of change .

Phase 1

Traditional Indexed File System: is the predatabase period where data used by an organization are often stored in many independent indexed files. **A** major limitation of Indexed File Database Systems lies in its lack of support for automatic linkage of files.

Phase 2

Hierarchical and Network Database System: in hierarchical model, record types are linked as an ordered set of treelike structure and the network data structure can be considered as a more general form of the hierarchical one , where arbitrary linkages of record types can be formed. These systems realize the sharing of an integrated database among multiple users. Traversing the tree structure carries out data retrievals. The key problem of these Database Systems lies in data dependence and tedious navigation in accessing the database.

Phase **3**

Relational Database Systems [41; the elegance of the Relational Model lies in its simplicity and its sound mathematical foundation. The essence of RDBS is its designation of **a** data structure **as** relations, giving rise to easy access of data. RDBS is characterized by declarative query with support for data independence. However, as new applications like CAD, CASE tools and multi-media system emerge, RDBS becomes inadequate, the reason being that these applications require monatomic, abstract data types which are not easily handled by the RDBS. Therefore, a new database system, which supports data independence as well as a richer set of data structure, should be constructed. A number of approaches have been taken. One approach is to extend the Relational Model to include user-defined data types.

Phase 4

Another approach uses the Object-Oriented Model. The final phase, Object-Oriented Database Systems [10], deals with OODBS. OODBS is based on the concepts of object-oriented paradigm. It supports multiple users and is characterized by a richer set of data types and database facilities. A higher level of modeling construct such **as** entities, relationships and inheritances accompanies the move from relational to OODBS. Furthermore, OODBS support the notion of data encapsulation that can be viewed as an integration of the structural and behavioral aspect of data.

IV. SQL-TYPE LANGUAGE

Recently the SQL language has rapidly become a standard data manipulation language for relational databases. We have formulated how to interpret fuzzy queries in the SQLtype language and implemented it in a Common Lisp version of Fuzzy-Set Manipulation System **[11].** We demonstrate it using the same query in the previous section. **[Query Sl]** Get the NAME in the relation R whose AGE is young. >(predicate (young (X)*(z* x 25 35))) (%young) >(select NAME from R

where (AGE = young)) {0.92/jack, {0.6/0.7,1/0.8)/Judy, {0.7/0.9,0.6/0.98,1/1}/Richard} Where the function (z x….)means standard function by Zadeh.

[Query S2] Get pairs of NAMEs in R whose AGES are approximately equal. >(select <X.NAME, Y.NAME>

from ((X RA) (Y RA)) where (and (not (equal X.NAME Y.NAME)) X.AGE Y.AGE))))

(approximately-equal

*the result is the same as above Query where the **X** and Y are alias names for **R.**

V. CONCLUSION

We have discussed here about fuzzy databse,its framework and its access using SQL. We will extend this work by using fuzzy queries with fuzzy quantifiers, such as **very**, **very, very** in future. By using the fuzzy query on fuzzy database concepts, we will extend traditional database management system which has some intelligence. Power of fuzzy database can be extended by using oodb approach, which provides multiuser access.

REFERENCES

[1] E.F. Codd (1970): “A Relational Model of Data for Large Shared Data Banks,” Communications of the AGM, Vo1.13, pp.377-387.

[2] Zongmin Ma, 2005, Fuzzy Database Modeling with XML, Springer. [3] Jose Galindo, Angelica Urrutia, Mario Piattini, 2006. Fuzzy Databases: Modeling, Design and Implementation, Idea GroupPublishing

[3] Medina, J. M., Pons, O., & Vila, M. A., GEFRED: A generalized model of fuzzy relational databases. Information Sciences, 76(1/2), 1994,87-109.

[4] C. J. Date. An Introduction to Database Systems. Addison-Wesley 1990.

[5] R. Elmasri and S. B. Navathe. Fundamentals of Database Systems. BenjamidCummings 1989.

[6] Peter Gray, K. G. Kulkarni and N. W. Paton. ObjecOriented Databases -- A Semantic Data Model Approach. Prentice-Hall, 1992.

[7] Paul Harmon. "Object-Oriented Database Management Systems," Object-Oriented Strategies Vol 11, No. 1, 1992.

[8] Angela Goh and Hui Siu-Cheung. "Object-Oriented Database Systems: the next Generation DBMS “, proceeding the XI Conference of SEARCC 1992.

[9] Handbook of Research on Fuzzy Information Processing in Databases by José Galindo, Volume 1

[10] [10]Won Kim. Introduction to Object-Oriented Databases, MIT, 1990.

[11] M. Umano (1987): “Fuzzy-Set Manipulation System in Lisp,” Second International Fuzzy Systems Association Congress (Tokyo, Japan, July 20 - 25, 1987), pp.84&843.

[12] Seymour Lipschutz, Theory and Problem of Data Structures, McGraw Hill Education, 199