www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 8 Aug 2015, Page No. 13887-13891

FRDMS For Fuzzy Querying Based On GEFRED Model

Annu Rani[1] , Sandeep Jaglan[2]
Computer Science and Engineering, N.C. College of Engineering, Israna
Panipat, India
[1]rohalannu9991@gmail.com
[2]sandeep@ncce.edu

Abstract **- When users uses the usual software tools they must have to change many valued logical thinking (approximate reasoning) within the two-valued computer logic. Although the Structured Query Language (SQL) is a very influential tool, it is not easy to satisfy needs for data selection based on linguistic terms and degrees of truth. In this paper, we are attentive in flexible querying which is based on fuzzy set theory. Medina et al. have developed a server named fuzzy SQL, associate flexible queries and based on a theoretic model called GEFRED. For modelling the flexible queries alongwith the concept of fuzzy attributes, an addition of the SQL language named fuzzy SQL has been defined. The FRDB has already been well-defined by the user. In this paper, we prolong the work of medina et al. to implement a new architecture of fuzzy DBMS based on the GEFRED model. The architecture is built on the theory in which we handle the weak coupling with the DBMS SQL Server.**

Keyword- FSQL, Possibility Model, GEFRED

INTRODUCTION

Databases are a very significant component in computer systems. Because of their rising number and volume, good and precise ease of access to a database becomes even more important. Organizations work with very giant data collections mostly stored in relational databases. Linguistic terms are attractive for data mining, analysis, propagation and decision making. The research area of fuzziness in Data Base Management Systems (DBMS) has resulted in a number of models intended at the representation of defective information in Data Bases, or at enabling non-particular queries (often called flexible queries) on predictable database schemas. Though, few works have been done from a convenient point of view.

The works of Medina *et al*. has been emphasized who introduced the GEFRED model in 1994 and its related language named FSQL. This language represents new concepts such as comparators, attributes, constants, etc. all comes with fuzzy. The thesis proposal is to execute a new structural design of the Fuzzy Relational DBMS (FRDBMS) based on the GEFRED model. This structural design is based on the weak coupling standard with the RDBMS SQL Server.

The research area of fuzziness in Data Base Management Systems (DBMS) has resulted in a number of models aimed at the representation of faulty information in DataBases (DB), or at enabling non-particular queries (often called flexible queries) on conventional database schemas [1, 2]. Though, few works have been done from a convenient point of view. The mainstream of these works used the fuzzy sets formalism to model the linguistic terms as “moderate”, “means” and to value the predicates as well as such terms. The important idea in these works consists in extending the SQL language and adding a additional layer to the relational DBMS to calculate the fuzzy predicates [3]. The use of this language is throughout a software named Fuzzy Query (FQ) [10]. Even though it solved a number of problems linked to the flexible queries modeling, FQ presents a number of limits: (1) it allows only the flexible querying of FRDB, (2) the FRDB is believed already implemented underneath Oracle, (3) the implementation of the DB is completed manually by the user, (4) FQ is not appropriate in practice for FRDB made up of more than ten tables. In this review paper, we propose a different design of the Fuzzy Relational DBMS (FRDBMS) based on the GEFRED model. This design is based on the weak coupling standard with the RDBMS Oracle. This FRDBMS offers all functionalities of a standard DBMS, in particular the depiction, the management and the querying of FRDB. Further this beginning, this paper includes different phases. Phase 1 represents the basic concepts of FRDB. Phase 2 represents the architectures previously used for the flexible querying modeling. Section 3 represents the architecture type of FRDBMS. Section 4 represents our new architecture of the FRDBMS as well as its accomplishment. Section 5 makes an estimation of this work and gives some potential perspectives of it.

General Definitions

- Database: A typical database is organised with collection of information (records or data) stored in a computer.
- Fuzzy Database: This database is a database which is capable to deal with uncertain or incomplete information using fuzzy logic.

- Fuzzy Logic: Fuzzy logic is derived from fuzzy set theory by Zadeh (1965) dealing with analysis that is estimated rather than correctly deduced from standard predicate logic. It can be consideration of as the application side of fuzzy set theory dealing with well attention out real world expert values for a compound problem.

- FRDB: FRDB is an addition of the relational database. This extension introduces fuzzy predicates under shapes of linguistic terms that, over the time of a flexible querying, allows to have a range of answers (each one with a membership degree) in order to offer to the user all intermediate variations between the completely satisfactory answers and those completely dissatisfactory .

- FRDBMS: It is an extension of the relational DBMS in order to treat, store and interrogate imprecise data.

- FRDB Models: **:** Two broad approaches are possibilistic model and the similarity relation based model. These models are considered in a very simple shape and consist in adding a degree, usually in the intermission [0, 1], to each tuple. They permits retaining the homogeneity of the data in DB. The main models under both approaches are Prade-Testemale, Umano-Fukami, Buckles-Petry, Zemankova-Kaendel and GEFRED of Medina *et al*... This last model constitutes an eclectic synthesis of the various models published so far with the aim of dealing with the problem of representation and treatment of fuzzy information by using relational DB.

Possibility Models

Under this category, the models which using possibility theory to signify fuzziness are included. The greatest important models that comes under this group are Prade-Testemale model, Umano-Fukami model, and GEFRED model. The GEFRED model consists of a general abstraction that permits for the use of different approaches, nevertheless of how dissimilar they might look.

1) **Prade-Testemale Model***:* Prade and Testemale issued a fuzzy relational database (FRDB) model that permits the integration of what they call Imperfect or uncertain data in the possibility theory sphere. An attribute A, having a D domain, is measured. All the existing knowledge about the value occupied by A for an x entity can be characterized by a possibility distribution πA(x) about D ∪ {e}, where e is a distinct element signifying the case in which A is not applied to x.

2) **Umano-Fukami Model:** *This* proposal also uses the possibility distributions in command to model information knowledge. In this, if D is the discourse universe of A(x), πA(x) (d) signifies the possibility that A(x) takings the value d∈D. The following knowledge may be modeled: unidentified and applicable information, the non-applicable information (undefined), and the total unawareness (we do not know if it is applicable or non-applicable):
   Unknown = πA(x)(d) = 1, ∀ *d* ∈ D
   Undefined = πA(x)(d) = 0, ∀ *d* ∈ D
   Null = {1/Unknown, 1/Undefined}

3) **GEFRED Model***:* The GEneralised model Fuzzy heart Relational Database (GEFRED) has been proposed in 1994 by Medina *et al*. .One of the chief benefits of this model is that it contains of a overall abstraction that permits for the use of several methods, irrespective of how dissimilar they might look. In fact, it is created by the generalized fuzzy domain and the generalized fuzzy relation, which comprise individually classic domains and classic relations. It constitutes an eclectic synthesis of the various models published so far with the aim of dealing with the problem of representation and treatment of fuzzy information by using relational DB.

Fuzzy Attributes in GEFRED Model

In order to model fuzzy attributes we distinguish between two classes of fuzzy attributes.

1) **Fuzzy Sets as Fuzzy Values:** These fuzzy attributes may be classified in four data types. In all of them the values unknown, undefined, and null are included:

- Fuzzy Attributes Type 1 (FTYPE1): These are attributes with “precise data”, classic or crisp (traditional) . However, they can have linguistic labels defined over them, which allow us to make the query conditions for these attributes more flexible.

- Fuzzy Attributes Type 2 (FTYPE2): These attributes admit together crisp and fuzzy data, in the form of possibility distributions above an underlying well-ordered domain (fuzzy sets). Table 2 shows the kinds of values defined in these attributes.

- Fuzzy Attributes Type 3 (FTYPE3): These are attributes over “data of discrete unordered dominion with analogy” .In these attributes some **l**abels are well-defined (“blond”, “red”, “brown”, *etc*.) that are scalars with a resemblance relationship defined over them, so that this relationship shows each pair of labels be similar to each other to what extent.

- Fuzzy Attributes Type 4 (FTYPE4): These attributes are defined in the same way as Type 3 attributes, without it being essential for a resemblance relationship to occur among the labels.

2) **Fuzzy Degrees as Fuzzy Values:** The domain of these degrees can be found in the interval [0,1], while other values are also acceptable, like a possibility distribution . The significance of these degrees is varied and depends on their use. The most imperative possible meanings of the degrees used by some authors are: fulfillment degree, Uncertainty degree, Possibility degree and Importance degree. The ways of using these fuzzy degrees are classified in two families: associated degrees (type 5, type 6, type 7) and non-associated degrees (type 8).

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 [α,β,γ,δ]	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

Table 1 Kind of values of fuzzy attributes type 2

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 permits fuzzy values.

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 :These stores the depiction of the parameters which describe the data and which depend on the type of value (FT), the name of these attributes is designed by the concatenation of numbers 1, 2, 3 and 4 in the name of the attribute to which they are belongging .

The fuzzy attributes type 3 is represented by a variable number of traditional attributes according to the form.

FT: is similar to FT used in FTYPE2 attribute.

(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.

Fuzzy attributes type 4 is represented just like type 3. Fuzzy degrees (types 5, 6, 7 and 8) are represented using a classic numeric attribute, because their domain is the interval [0, 1].

The FSQL Language

The FSQL language is an authentic extension of SQL language to model fuzzy queries.

- Linguistic Labels: If an attribute is capable of fuzzy dealing then linguistic labels can be well-defined on it. These labels will be headed with the symbol $ to extricate them easily. Every label has an associated trapezoidal possibility distribution (for fuzzy attributes type 1 and 2) or a scalar (for fuzzy attributes type 3 and 4). Valid statements in SQL are also valid in FSQL.

- Fuzzy Comparators: Moreover the classic comparators (=,>, *etc*.), FSQL includes fuzzy comparators.
- Function CDEG: the function CDEG (compatibility degree) can be used with an attribute in the argument to compute. It calculates the satisfaction degree of the condition of the query for the attribute defined in its argument.
- Fulfillment Thresholds: For each simple condition, a fulfillment threshold τ may be well-known (default is 1) with the format: <condition> THOLD τ showing that the condition must be fulfilled with minimum degree τ ∈□[0,1] to be considered.
- Fuzzy constants: Moreover the typical constants (numbers, NULL…), FSQL involved many constants such as fuzzy trapezoidal $[a, b, c, d], approximate values #n, $label, [n, m], UNKNOWN, UNDEFINED, etc.
- Fuzzy Quantifiers: There are of two types: absolute and relative. They permits us to use expressions like "most", "almost all", "many", "very few", etc.

New Architecture of the FRDBMS

We advise the weak coupling method with DBMS. The perception of weak coupling is shown in Figure 1.The FRDBMS suggested respects the GEFERD model. The language of explanation and management of the data is therefore FSQL. Seen that the FSQL language is an extension of the SQL language, a FRDBMS can model a RDB (described in SQL language) or a FRDB (described in FSQL language). The standard of this coupling is the description of a software layer that permits the conversion of the command written by the user in FSQL language in their correspondent written in SQL.

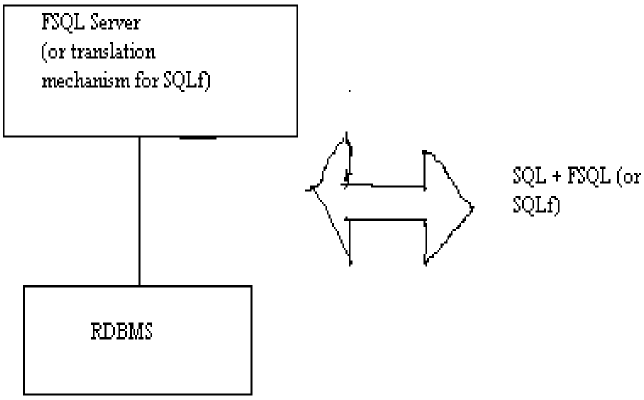

Figure 1 Weak Coupling Concept

In order to implement a system which characterize and deploy "imprecise" information, Medina *et al.* have developed FIRST architecture (a fuzzy Interface for relational systems). It is built on RDBMS Client-Server architecture provided by Oracle. It prolongs the existing structure and adds some new components to handle fuzzy information. The main essential component added to this architecture is the FSQL Server which declares the translation of flexible queries written in FSQL in a comprehensible language by the DBMS (SQL).The installation of this architecture is described in Figure 2.

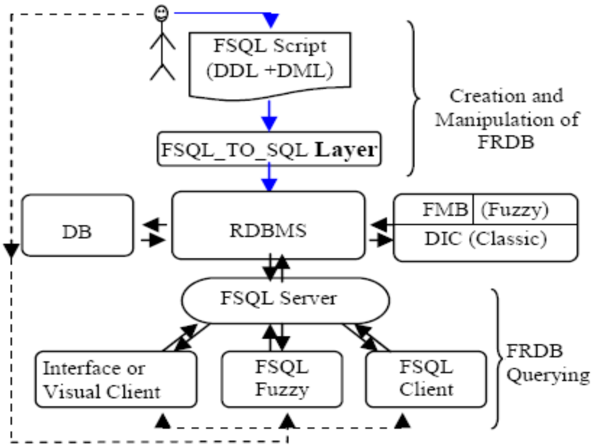

Figure 2 the FRDMS Architecture

Conclusion

Fuzzy relational data bases have been broadly studied in a academic level. The majority of these works used the fuzzy sets formalism to model the linguistic expressions as "moderate", "means" and to value the establishes including such terms. Medina et al. have developed a server named fuzzy SQL, associate flexible queries and based on a theoretic model called GEFRED. This server has been planned in PL/SQL language under Oracle database managing systems. To model the flexible queries and the concept of fuzzy attributes, an extension of the SQL language named fuzzy SQL has been well-defined. The FSQL language prolongs the SQL language, to maintenance the flexible queries, with many fuzzy perceptions. The FRDB is supposed has already been well-defined by the user. In this tender, we extend the work of medina et al. to implement a software layer which will translate FSQL queries to SQL queries. This architecture is built on the concept of weak coupling with the DBMS. This will enable the user a powerful and easy to use data mining tool which permits him to query data from databases by using linguistic expressions in order to recover the quality of selection process.

Future Scope

The proposed architecture of FRDBMS based on the GEFRED model makes use of weak coupling concept with the DBMS. As a future work a new architecture supporting the concept of strong coupling with DBMS can be developed. As futures perspectives of this work, we also mention the automatic mapping of existing relational DB to FRDB. This point is theoretically done but not implemented yet, so we think that it will contribute to make easier the use of the FRDB in real applications.

References

[1]Amel Grissa T. and Mohamed Hassine , "New Architecture of Fuzzy Database Management Systems," The International Arab Journal of Information Technology, Vol. 6, No. 3, July 2009 213

[2] Bosc P. and Pivert O., "SQLf: A Relational Database Languageb5n6for Fuzzy Querying," Computer Journal of IEEE Transactions on Fuzzy Systems, vol. 3, no. 1, pp. 80-89, 1995.

[3] Bosc P., Lietard L., and Pivert O., "Bases de Donnees et Flexibilite: Les Requetes Graduelles," Computer Journal of Techniques et Sciences Informatiques, vol. 7, no. 3, pp. 355-378, 1998.

[4] Bosc P. and Pivert O., "SQLf Query Functionality on Top of a Regular Relational Database Management," in Proceedings of Knowledge Management in Fuzzy Databases, Heidelberg, pp. 171-190, 2000.

[5] Buckles P. and Petry E., "A Fuzzy Representation of Data for Relational Databases," in Proceedings of Fuzzy Sets and Systems, USA, pp. 213-226, 1982.

[6] Galindo J., Medina M., Pons O., and Cubero J., "A Server for Fuzzy SQL Queries," in Proceedings of Lecture Notes in Artificial Intelligence (LNAI), USA, pp. 165-175, 1998.

[7] Galindo J., "Tratamiento de la Imprecision en Bases de Datos Relacionales: Extension Del Modelo y Adaptacion de los SGBD Actuales," Doctoral Thesis, Universidad de Granada, 1999.

[8] Galindo J., Aranda M., Caro L., Guevara A., and Aguayo A., "Applying Fuzzy Databases and FSQL to the Management of Rural Accommodation," Tourist Management Journal, vol. 23, no. 6, pp. 623-629, 2002.

[9] Galindo J., Urrutia A., and Piattini M., Representation of Fuzzy Knowledge in Relational Databases, IEEE Computer Society, 2004

[10] Galindo J. "New Characteristics in FSQL: A Fuzzy SQL for Fuzzy Databases," Computer Journal of WSEAS Transactions on Information Science and Applications, vol. 2, no. 2, pp. 161- 169, 2005.

[11] Galindo J., "Le Serveur FSQL and FQ," http: //www.lcc.uma.es/personal/ppgg/FSQL.html, 2008.

[12] Galindo J., Urrutia A., and Piattini M., Fuzzy Databases: Modeling, Design and Implementation, Idea Group Publishing, Hershey, 200

[13] Grissa A., Ben Hassine A., and Ounelli H., "Extended_FSQL_Server: A Server for the Description and the Manipulation of FRDB," in Proceedings of the 4th International Multi Conference on Computer Science and Information Technology (CSIT), Jordan, pp. 454- 464, 2006.

[14] Medina M., Pons O., and Vila A., "An Elemental Processor of Fuzzy SQL," Computer Journal of Math Ware and Soft Computing, vol. 1, no. 3, pp. 285-295, 1994.