A Comparative Study of Existing Fuzzy Query Systems of Database

Mama Rachid and Machkour Mustapha

Abstract Relational Database Management Systems (RDBMS) have become, without a doubt, the core of any computer system. Besides, in many cases information is found to be naturally fuzzy or imprecise, that's why fuzzy query systems have become indispensable to represent and manage this information and especially facilitate interrogation to a non-expert user, but the problem is that Boolean queries do not allow the user to use vague and imprecise language terms in the qualification criteria of the searched data or to express preferences between these criteria, which is often a request legitimate end user. Nowadays, there are many proposals that allow users to make fuzzy queries on relational databases. In this chapter, we will briefly review the main attempts to find a perfect solution to this problem, highlighting their advantages, disadvantages and difficulties encountered.

Keywords Database management systems · Fuzzy query · Flexible queries · Fuzzy logic

1 Introduction

Data plays an important role in our daily lives, such as bank accounts, games, social networks, videos, etc. However Relational Database Management Systems (RDBMS) have become the core of any computer system. In many cases, the information is naturally fuzzy or inaccurate because the knowledge that humans have about the world is almost never perfect. Thus, the knowledge on which human reasoning is based, are almost always tainted with uncertainties and inaccuracies. In fact, these imperfections emanate from the very nature of man and the world. Traditional interrogation systems are unable to deal with uncertainty and vagueness.

M. Rachid (✉) · M. Mustapha
Information Systems and Vision Laboratory, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
e-mail: mama.rachid@edu.uiz.ac.ma; m.machkour@uiz.ac.ma

M. Elhoseny et al. (eds.), *Distributed Sensing and Intelligent Systems*, Studies in Distributed Intelligence, https://doi.org/10.1007/978-3-030-64258-7_40

In addition, a relational database management system supports a structured query language (SQL) for data processing.

This language is based on standards based on Boolean interpretations that prevent database experts from processing fuzzy information. To illustrate this problem, consider a user who consults by the Internet a database of car rental offers. The user wants to rent a new car, cheaper and with reduced fuel consumption, this request can be expressed as:

```
SELECT * FROM tbl_cars
WHERE prod_year=" new" AND Price="cheap"
AND fuel_consumption="small"
```

with tbl_carsis a table that contains numeric data. The problem is that traditional interrogation systems are unable to handle these kinds of fuzzy terms such as "new," "cheap," and "small."

Several works have been proposed in the literature to introduce flexibility in database querying. Most of these works have used the fuzzy sets and fuzzy logic formalism to model linguistic terms such as ("new," "small") and to evaluate predicates with such terms. The main idea of this work is to extend the SQL language and add an additional layer of a classical DBMS to evaluate fuzzy predicates [1–3].

In this chapter, we present a comparative study of the most relevant fuzzy Query systems of database, along with the advantages and the drawbacks of each one, and finally a conclusion.

2 Background

The problem of the representation and processing of "imprecise" information has been widely studied by several authors [2, 4]. However, all the models published to give the solution to this problem have their advantages, disadvantages and their limitations. The problem is not trivial, it is necessary to modify the structure of the relations and, with these, the operations defined on them. To help store imprecise information and to consult it in a flexible way, this information requires the study of a multitude of cases that do not occur in the classical model.

The first models are mainly theoretical models of fuzzy relational databases, among them the model of Bukles-Petry [5] proposed by Buckles and Petry in 1980, This is the first model that uses the similarity relations in the relational model, he defined a fuzzy representation for relational databases, in which non-fuzzy databases are a special case of this model. The structure for representing imprecise information that has been defined in this model differs from ordinary relational databases in two important ways: n-tuplet components do not need to be unique values and a similarity relation is required for each set of domains in the database. A fuzzy relation R is defined as a subset of the Cartesian product $2^{D_1} \times 2^{D_2} \times \cdots \times 2^{D_n}$ where 2^{D_i} is any not null member of the domain base set D_i.the domains are either discrete scalars, or discrete numbers from a finite or infinite set.

The values of a particular tuple can be simple scalars or numbers (including nulls) or a finite set of scalars or numbers. .for example (Table person):

NAME	APTITUDE	AGE
{SMITH}	{AVERAGE,GOOD}	{21,22,23}

The resemblance relation that exists on each of the domains serves to represent and direct the imprecision. It establishes a measure of similarity s (x, y) between the different values of the domain on which it is defined. It is defined by the user and the resemblance values are between 0 and 1. (0: Completely different; 1: Completely similar). In a query, the user asks about tuples that satisfy a given condition for a given similarity threshold. For example:

```
PROJECT( PERSONNE :APTITUDE ,AGE)  WITH  THRES(APTITUDE≥0.5) ,
THRES(AGE)≥0.75
```

The tuples of the relationship are grouped into equivalence class, according to the relations and the thresholds of similarities defined on them. The main disadvantages of this model are:

- it does not model well all the fuzzy aspects of information (for example fuzzy modifiers, fuzzy quantifiers ...).
- atomicity is not guaranteed in the representation of information.
- the integrity of the database is not guaranteed.
- the result having several interpretations.

However, we can mention some advantages such as the use of resemblance relationships is an appropriate and intuitive tool for representing imprecision, and the use of different thresholds for each of the attributes.

Umano et.al. [6] proposed in 1980 a model based on the theory of possibilities. It is one of the first models of fuzzy relational databases. He uses as value:

- Possibility distributions
- Undefined: $\pi_{A(x)}(d) = 0$, $\forall\ d \in D$, Unknown: $\pi_{A(x)}(d) = 1$, $\forall\ d \in D$ And Null = {1/Unknown,1/Undefined}with D is the discourse universe of $A(x)$ and $\pi_{A(x)}(d)$ is the measure of possibility.

The consultation in this module returns three subsets: the tuples that clearly satisfying the consultation, tuples that approximately satisfy the consultation and tuples that do not clearly satisfy the consultation. Among the advantages of this model is that it can assign a degree of belonging to each tuple of the relationship, as well as it can store possibilities distributions. But there are the limits like does not handle non-scalar data, do not support the similarity relationship, and also does not model well all the fuzzy aspects of the information (for example Fuzzy modifiers, Fuzzy quantifiers ...).

In 1984, Henri Prade and Claudette Testemale [7] proposed a model based on the distribution of the possibility introduced by Zadeh [8, 9] to represent and process

partial, uncertain or fuzzy data and to take into account vague queries. It is an approach that generalizes the representations of Buckles-Perty and Umano-Fukami and describes an extended relational algebra. The value of the attributes and the vague predicates are represented by means of possibility distributions evaluated by [0,1]. The data structure is similar to that used in the Umano-Fukami model. It uses measures of possibility and necessity to satisfy the conditions established in the consultation. The allowed domains in this model are:

- Finished set of scalar. Example $D = \{\text{red, blond, brown}\}$
- Finished set of number. Example $D = \{21,22,23\}$
- Set of fuzzy numbers or fuzzy labels. Example $D = \{\text{small, medium, big}\}$

The possible values for these domains are:

- Precise values. Example 25
- Interval values. Example [30,34]
- Fuzzy values. Example "good," "about-10," "bad-to-very-bad"
- Null values "Unknown" and "does-not-apply"
- A distribution of possibility. Example {1/M, 0.6/D}

For example, the relation person, can correspond to a table such as:

Name	Age	Family-situation[a]
David	25	Unknown
Tom	[30,34]	U
Paul	Young	{1/W,1/D}
Jean	About-50	{1/M,0.6/D}

[a]*M* married, *U* unmarried, *D* divorced, *W* widow(er)

{1/M, 0.6/D} means that there is a possibility equal to 1 that the person is married, and possibility equal to 0.6 that he is divorced, and a zero possibility for the others.

Although this module has defined an acceptable generalization for the representation of uncertain and incomplete information. There are still some disadvantages like:

- It does not model well all the fuzzy aspects of the information (for example Fuzzy modifiers, Fuzzy quantifiers, Fuzzy group by ...)
- Do not support the similarity relationship
- Does not support multivalued attributes such as spoken-language (David) = English and Arabic
- Do not model values that are related to each other

In 1985, Maria ZAMANKOVA and Abraham KENDEL [10] proposed another fuzzy relational database model, this model is based on research in relational data and theories of fuzzy sets and the possibility. It allows to recover the information desirable by the application the rules of linguistics fuzzy terms of the query. Among the advantages of this module is that it takes into account individualization. A user

can define specific functions or rules that can be added to the system vocabulary. For example, a definition of a fuzzy set AGE may differ from one user to another even though the Age data is the same in the database. This model consists of three parts:

- A database of values (VDB) that store the actual data values.
- An explanatory database (EDB) that stores definitions for fuzzy subsets and fuzzy relationships is one part that reflects a user's knowledge profile.
- A set of translation rules that are used to manipulate adjectives.

The allowed domains in this model are:

- Set of discrete scalars. Example color = {red, blond, brown}
- Set of discrete or continuous numbers
- The unit interval [0, 1]

And the possible values for these domains are:

- Simple scalars or number
- A possibility distribution
- A real number in the interval [0,1] which is the value of the membership function or distribution of possibility
- Null value

For example, the person relationship may represent as:

Name	Age	Hair color	Smart
David	25	0.8/black + 0.3/brown	0.5
Tom	30	0.6/red + 0.7/blond	0.4
Paul	82	1/black	0.9

Among the disadvantages of this module is that do not support fuzzy quantifiers, fuzzy grouping, and fuzzy join. Also, the dependency with the relational model, which is not treated by this model. In addition, it does not allow the user to specify the accuracy with which the conditions involved in a query are met.

The most generalized model is the Generalized model for fuzzy relational database (GEFRED) which was proposed in 1994 by Medina, Pons and Villa [11]. it constitutes an eclectic synthesis of the various published models to treat the problem of the representation and the treatment of the fuzzy information by means of the relational databases. it is based on the Generalized fuzzy Domain (D) and on the Generalized fuzzy Relationship (R), one of the main advantages of this model is that it consists of a general abstraction that makes it possible to treat different approaches, even those that may seem very disparate. The possible data in the GEFRED model can be consulted in [11].

Based on the theoretical GEFRED model and the resources of the classical relational model, Medina and .al have developed a module called Fuzzy Interface for RelationalSystems (FIRST) to extend the capabilities of a classic DBMS so that it can represent and manipulate imprecise information. It is based on the client-

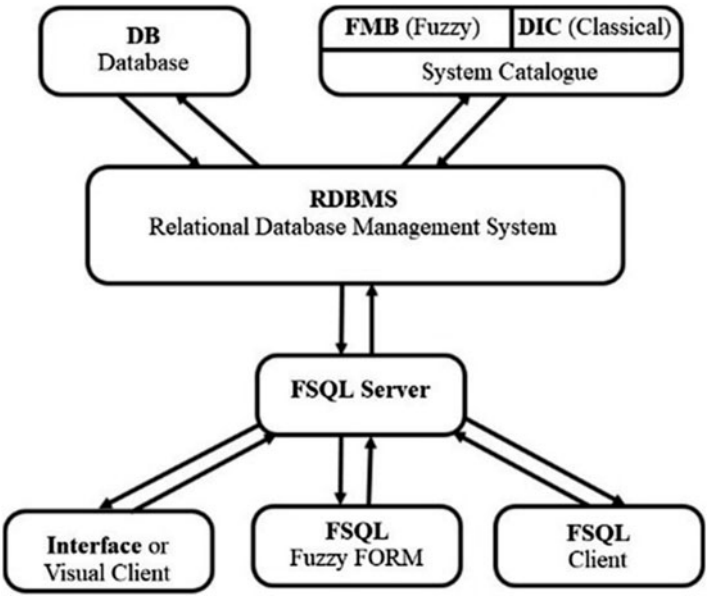

Fig. 1 FIRST architecture

server RDBMS architecture provided by Oracle. It adds new components (Fuzzy Meta Knowledge Base "FMB," FSQL Server, etc.) to the existing structure to handle imprecise information. Figure 1 shows the general architecture of this model.

This model uses a specific query language called Fuzzy SQL (FSQL),it is an extension of SQL to allow flexible queries. It already extends the existing commands in SQL, but it also incorporates novelties like fuzzy attributes, fuzzy constants, fuzzy comparators, fuzzy quantifiers ...

For example, if we consider a table person and we want to find young person (with a threshold of 0.4) who live in New York and have a salary greater than or equal to the trapezoidal distribution [100,300,500,800] .the FSQL query is written:

```
SELECT name ,CDEG(age),salary FROM person WHERE age FEQ $young 0.4
AND salary FGEQ  $[100,300,500,800] AND address = 'new york'
```

The FMB component deals with the storage of attributes that allow fuzzy processing and the information of each of them according to their type in a relational format .while the FSQL server's role is to extract the queries written with the FSQL language and translate them into SQL language using the information contained in the FMB. (see Fig. 2).

Although this model has several advantages in the representation and processing of information fuzzy, there are the weaknesses, such as:

- The problem concerns the choice of the type of the attribute (FTYPE1, FTYPE2, or FTYPE3), because an attribute can be in FTYPE1 cases and in other cases FTYPE2.

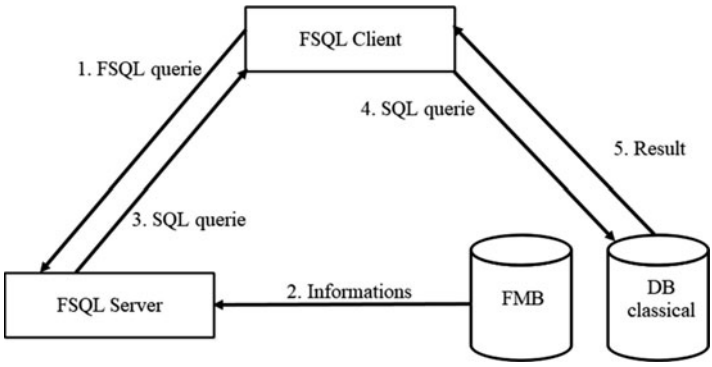

Fig. 2 FSQL Server operations

- The GIFRED model theoretically defines some features that have not yet been implemented in this module(Example Fuzzy group by).
- The approach uses a parser/translator to check and convert an FSQL query to SQL respecting the definitions of any fuzzy terms or operators stored in another database (FBM). This will slow down the query process.
- This module requires a good description of the different operations to be done at the database level and at the FBM level. This operation becomes heavier and trickier if the database becomes very large.
- The SQL language remains unusable by a non-expert user.
- This model does not allow the user to describe the fuzzy database (FDB) schema or manipulate its FDB.

Several approaches have been proposed to improve the FIRST model like that of José Galindo [12] which introduced the second version (FIRST 2) which contains new comparators, new fuzzy attributes, new fuzzy constants, and new feature in executionthresholds ... etc. and Martinez [13] who introduced an approach to extend non-scalar attribute management using ontology, he presents a new system (see Fig. 3) that combines fuzzy logic and ontology for get an answer as complete as possible.

Another fuzzy query language was proposed by Patrick Bosc and Olivier Pivert [14] in 1995 called SQLF for the purpose of remedying the problems posed by the SQL language in flexible queries. The structure of the SQL base block is kept in SQLF:

```
SELECT   [distinct ]   [n| t | n, t]   <attributes>
FROM   <relations>    WHERE   <fuzzy condition>
```

The "FROM" clause does not undergo any change, the changes concern two points: the calibration of the result and the nature of the authorized conditions which may contain Boolean or gradual conditions, or both connected by connectors. In SQLF, a multi-relation block combines projection, restrictions, and algebraic or fuzzy joins:

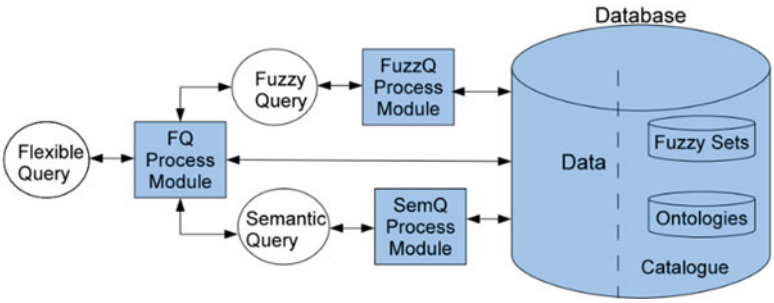

Fig. 3 System architecture proposed by Martinez

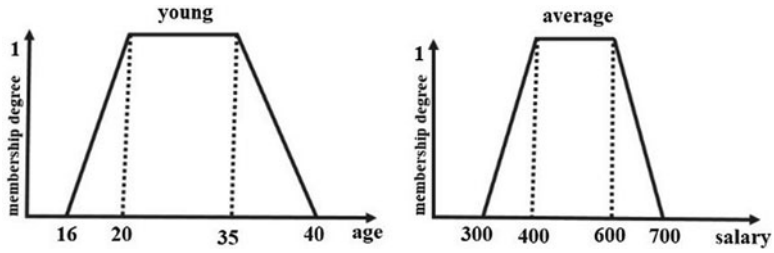

Fig. 4 Definition of fuzzy sets 'young' and 'average'

SELECT distinct R.A, S.B.
FROM R, S WHERE f_{cR} *and* f_{cS} *and (R.C* θ *S.D).*

where (R.C θ S.D)is the fuzzy join condition for example (R.C "roughlyequal to" S.D) and f_{cR} (resp. f_{cS}) it's a selection expression on R (resp S). The relational and set operations used in Boolean queries have been extended to take into account fuzzy predicates and return fuzzy relationships. An example of a fuzzy query addressed to the person table using the predicates shown in Fig. 4 would be:

```
SELECT name ,age,salary
FROM person  WHERE age = 'young' AND salary = ' average'
```

All valid queries in SQL are always valid in SQLF, it is an important point for optimizing the evaluation of the query. Among the disadvantages of SQLF is that do not manage non-scalar data, and do not support the similarity relationship.

SQLF has undergone several enhancements like the one made by Kacprzyk and Zadrozny [15], as part of their FQUERY package for Access, to increase the efficiency of the fuzzy query engine.

3 Results

We presented a brief study on proposed models. The tables (extract from [13]) below show a summary of our study (Tables 1 and 2).

Table 1 Comparison of most relevant characteristic in fuzzy query systems (part 1)

Model	Medina [11]	Bosc [14]	Zemankova [10]	Prade [7]	Martinez [13]	Umano [6]	Kacprzky [15]	Buckles [5]
Manage scalar data	.	.	.	.	.	.	.	.
Manage non-scalar data	.		.	.	.			.
Similarity relationship	.		.		.			.
Possibility distributions	.	.	.	.	.		.	
Degree in attributes level	.			.	.	.		.

Table 2 Comparison of most relevant characteristic in fuzzy query systems (part 2)

Model	Medina [11]	Bosc [14]	Zemankova [10]	Prade [7]	Martinez [13]	Umano [6]	Kacprzky [15]	Buckles [5]
Degree in tuple level	.	.	.	.	.	.	.	
Fuzzy modifiers		.	.					
Fuzzy quantifiers	.	.					.	
Fuzzy comparison operators	.	.	.	.	.	.	.	.
Fuzzy group by		.					.	
Fuzzy joins	.	.		.		.		
Nesting	.	.						
Store fuzzy data	.		.	.		.		.
Fuzzy queries	.	.	.	.	.	.	.	.
Extension SQL language	.	.			.	.	.	

We conclude that none of the proposals is complete, most of them give a partial version of the representation and processing of imprecise information and the implemented proposals also depend on the platform.

4 The Intended Model

The implementation of any system requires a detailed study. This study must consider the needs of users. The desired system must be flexible, able to provide the appropriate mechanisms for the representation, processing and retrieval of fuzzy information in all its forms, in addition it must be considerably collaborate with the commercial DBMS in an efficient way to obtain better performances. And, must be regardless of the platform.

Our system must be complete, having all the features and operators to recover and process fuzzy information. The previous systems are incomplete; the model that offers more functionality is the GEFRED model that has been proposed by Medina et al. [11].

Our system must be able to consider the fact that the user can be non-expert for example do not know the schema of the database. So, the system must have a friendly graphical interface to facilitate the tasks. For example, help the user to define his own linguistic terms. And allow users to compose their questions in natural language and receive the answer in natural language.

5 Conclusion

To sum up, even though many models have been proposed, either by using theories of possibility or fuzzy logic, the problem of implementing a flexible fuzzy query system of database is still persisting. Therefore, improvements need to be made to provide flexible and user-friendly interfaces to RDBMSs.

References

1. Zemankova-Leech, M., & Kandel, A. (1984). *Fuzzy relational databases—A key to expert systems*. Verlag TUV Rheinland GmbH. B. Simpson et al. Title of paper goes here if known. unpublished.
2. Galindo, J., Urrutia, A., & Piattini, M. (2006). *Fuzzy databases: Modeling, design and implementation*. Idea Group Publishing.
3. Mama, R., & Machkour, M. (2019). A study on fuzzy interrogation systems of database. In *International conference of computer science and renewable energies (ICCSRE)* (pp. 1–6).
4. Kacprzyk, J., Zadrozny, S., & De Tré, G. (2015). Fuzziness in database management systems. *Fuzzy Sets and Systems, 281*(C), 300–307.

5. Buckles, B. P., & Petry, F. E. (1982). A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems, 7*(3), 213–226.
6. Umano, M., Hatono, I., & Tamura, H. (1995). Fuzzy database systems. In *Proceedings of the FUZZIEEE/IFES'95 workshop on fuzzy database systems and information retrieval* (pp. 53–36).
7. Prade, H., & Testemale, C. (1984). Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences, 34*(2), 115–143.
8. Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems, 1*(1), 3–28.
9. Zadeh, L. A. (1981). Test-score semantics for natural languages and meaning representation via PRUF. In B. B. Rieger (Ed.), *Empirical semantics* (Vol. 1, pp. 281–349). Bochum Brockmeyer.
10. Zemankova, M., & Kandel, A. (1985). Implementing imprecision in information systems. *Information Sciences, 37*(1–3), 107–141.
11. Medina, J. M., Pons, O., Vila, M. A., & GEFRED. (1994). A generalized model of fuzzy relational databases. *Information Sciences, 76*(1–2), 87–109.
12. Galindo, J. (2005). New characteristics in FSQL: A fuzzy SQL for fuzzy databases. *Computer Journal of WSEAS Transactions on Information Science and Applications, 2*(2), 161–169.
13. Martinez-Cruz, C., Noguera, J. M., & Vila, M. A. (2016). Flexible queries on relational databases using fuzzy logic and ontologies, information sciences. *Vol., 366*(20), 150–164.
14. Bosc, P., & Pivert, O. (1995). SQLf: A relational database language for fuzzy querying. *IEEE Transactions on Fuzzy Systems, 3*(1), 1–17.
15. Kacprzyk, J., & Zadrożny, S. (2001). SQLf and FQUERY for access. In *Proceedings of the IFSA World Congress and 20th NAFIPS international conference* (Vol. 4, pp. 2464–2469).