Relaxing the Universal Quantifier of the Division in Fuzzy Relational Databases

José Galindo,[1,*] Juan M. Medina,[2,†] Juan C. Cubero,[2,†]
M. Teresa García[3,‡]
[1] *Departamento Lenguajes y Ciencias de la Computación, Universidad de Málaga, 29071 Málaga, Spain*
[2] *Departamento Ciencias de la Computación e I.A., Universidad de Granada, 18071 Granada, Spain*
[3] *Departamento Lenguajes y Sistemas Informáticos, Universidad de Cádiz, 11003 Cádiz, Spain*

In a previous paper, we presented an approach to calculate relational division in fuzzy databases, starting with the GEFRED model. This work centered on dealing with fuzzy attributes and fuzzy values and only the universal quantifier was taken into account since it is the inherent quantifier in classical relational division. In this paper, we present an extension of that division to relax the universal quantifier. With this new system we can use both absolute quantifiers and relative quantifiers irrespective of how the function of the fuzzy quantifier is defined. We also include a comparison with other fuzzy division approaches to relax the universal quantifier that have been published. Furthermore, in this paper we have extended the fuzzy SQL language to express any kind of fuzzy division. © 2001 John Wiley & Sons, Inc.

1. INTRODUCTION

On a theoretical level, there are many fuzzy relational database (FRDB) models, which are based on the relational model and which can be extended to allow the storage and/or treatment of vague and uncertain information. The FRDB models are based on the concept of fuzzy relation. However, there are several ways of representing and handling imprecise or uncertain information in these fuzzy relations. Moreover, these models can be mixed to allow a greater flexibility.

Ref. 1 lists a compendium of FRDB models and their main characteristics. In Refs. 2 and 3 a brief summary of the most important ways of introducing fuzzy information into the fuzzy relations was presented. What the best method

* Author to whom correspondence should be addressed; e-mail: ppgg@lcc.uma.es.
† e-mail: {Medina,carlos}@decsai.ugr.es.
‡ e-mail: mayte.garcia@uca.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 16, 713–742 (2001)
© 2001 John Wiley & Sons, Inc.

is will depend on the circumstances. The two most used ways are:

(1) **Fuzzy values in the attributes:** It is possible to store fuzzy values as attribute values in a relation and to operate with them. The fuzzy values are mainly fuzzy numbers, possibility distributions, or different labels (scalars) with a similarity relation between each two labels. This type of fuzzy relations is used in Refs. 3–13. In particular, the GEFRED model[7] represents a synthesis among the different models which have appeared to deal with the problem of the representation and management of fuzzy information in relational databases and this model allows all data types in Table I to be represented. The unknown, undefined, and NULL types are defined in Umano[11] and Fukami et al.[14]

(2) **Grade for every value of an attribute or for a tuple:** This implies that every value of every attribute or the whole tuple can have an associated grade (or degree), generally in the interval [0, 1], that measures the level of fuzziness of this value. The domain of these grades is usually limited to the interval [0, 1], but other values can be allowed, as, for example, possibility distributions in [0, 1], since it may be difficult to know all the degrees precisely and therefore this grade is pervaded with uncertainty and imprecision.[15] The semantics of these degrees can vary. Therefore, the most important **meanings** of these grades may be **fulfillment degree** (of a property or condition),[7,15,16] **membership degree** (measuring the level of membership of an object to a set),[8,9,12,17,18] and **importance degree** (of every object).[8,15,16] In some contexts, fulfillment and membership degree may be considered to be the same thing, since the membership degree to a set S may measure to what extent the property S is fulfilled.

It is usual to mix some of these ways or several variations (such as associating two values per tuple with the meaning of necessity and possibility[9]). However, if some of these ways are used, although a greater flexibility is achieved, the database semantic becomes very difficult to understand. In this paper we will focus on the use of fuzzy relations, considering fuzzy values in the

Table I. Data types for GEFRED FRDB model.

1.	A single scalar (e.g., Size = Big, represented by the possibility of distribution 1/Big).
2.	A single number (e.g., Age = 28, represented by the possibility of distribution 1/28).
3.	A set of mutually exclusive possible scalar assignations (e.g., Behavior = {Bad,Good}, represented by {1/Bad,1/Good}).
4.	A set of mutually exclusive possible numeric assignations (e.g., Age = $\{20, 21\}$, represented by $\{1/20, 1/21\}$).
5.	A possibility distribution in a scalar domain (with a similarity relation) (e.g., Behavior = {0.6/Bad,1.0/Average}).
6.	A possibility distribution in a numeric domain (e.g., Age = $\{0.4/23, 1.0/24, 0.8/25\}$, fuzzy numbers or linguistic labels).
7.	A real number belonging to $[0, 1]$, referring to the degree of matching (e.g., Quality = 0.9).
8.	An **unknown** value with possibility distribution **unknown** = $\{1/u\colon u \in U\}$ on domain U, considered.
9.	An **undefined** value with possibility distribution **undefined** = $\{0/u\colon u \in U\}$ on domain U, considered.
10.	A **null** value given by **null** = {1/Unknown,1/Undefined}.

attributes and an optional grade for each attribute with the meaning of fulfillment degree (or compatibility degree) of a concrete condition.

Before studying fuzzy division, let us look at a definition, in a classical sense, for the operator of relational algebra, the so-called relational division:

DEFINITION 1. *Let R and R' be relations with headers (A, B) and (B), respectively, where A and B are simple attributes or sets of attributes. Then, the* ***relational division*** *of R by R', denoted by $R \div R'$, is a relation with header (A) whose body is formed by all the tuples $(A : a)$ so that a tuple $(A : a, B : b)$ exists in R for every tuple $(B : b)$ in R'.*

So, we may say that tuples in relation $R \div R'$ comply with the division requirements.

Sometimes, the former definition is extended to take into account the case when the relation R' also has attributes which are not common to R. In this case, if (B, C) is the header of R', the division results are tuples $(A : a, C : c)$ so that a tuple $(A : a, B : b)$ exists in R for all tuples $(B : b, C : c)$ in R'. This result is equivalent to computing the division of R by $R'[B]$ (the projection of R' onto B) and afterwards by computing the cartesian product of the resulting relation and $R'[C]$ (the projection of R' onto C), denoted by $(R \div R'[B]) \times R'[C]$.

We can therefore give the following definition for any kind of division:

DEFINITION 2. *Let R and R' be relations with headers (A, B, C) and (B, D), respectively, where A, B, C, and D are simple attributes or sets of attributes denoted by, for example, $A = \{A_1, \dots, A_n\}$. We can therefore define the* ***general relational division*** *of R by R':*

$$R \div_{A,B,X} R' = (R[A, B] \div R'[B]) \times R'[X] \tag{1}$$

where X is a set of R' attributes: $X \subseteq \{B \cup D\}$.

If C, D, and X are empty sets, then Eq. 1 is the usual division. If C is not empty, then we select the important R attributes (A) and we ignore the other attributes (C). This allows us to perform the division using any set of attributes and not only those which are not common to R'. If X is not empty, then we select some R' attributes (X) to obtain how A values are related with them, through R and R' (using the division semantic).

Of course, B implies an implicit matching between R and R' attributes with the same domain, whatever their attribute names. General relational division allows division between any two relations with the only requirement being that they have at least one attribute with the same domain.

Relational division uses the universal quantifier (for all, ∀), selecting tuples of the first relation which are related, in some way, to “all” tuples in the second relation. In this article, we provide a solution by using fuzzy quantifiers in classical or fuzzy databases.

Fuzzy quantifiers[19] may be absolute (like "many", "very few", "approximately x"...) or relative (like "most", "almost all", "approximately less than half"...). Absolute quantifiers are defined as fuzzy sets over nonnegative integers, while relative quantifiers are fuzzy sets over the real interval $[0, 1]$.

Linguistic quantifiers have been widely used in database contexts to process flexible queries: for example, the method presented by Zadeh,[19] the method using ordered weighted averaging (OWA) operators[20] presented by Yager[21] or the method presented by Vila et al.[22] (in this paper there is also a comparative study of Zadeh's and Yager's methods). These studies are useful when computing the fulfillment degree (or truth degree) of sentences which include linguistic quantifiers (absolute or relative) and properties (vague or not). For example, they compute the fulfillment degree of the following kind of sentences: "*Most students are good at mathematics*." A survey of methods for evaluating quantified sentences and some new methods are shown in Refs. 23 and 24.

However, these methods only return a fulfillment degree and they do not return the set of tuples which comply with the sentence and the fulfillment degree of each one. In this paper, we formalize a method to carry out fuzzy division using any fuzzy quantifier (absolute or relative) whatever the definition of its function is, i.e., irrespective of how the function of the fuzzy quantifier is defined. This system returns the set of tuples which comply with the sentence as well as with the fulfillment degree of each one. This will allow us to set a threshold, u, to select only those tuples whose fulfillment degrees are greater than or equal to u.

This system is based on the generalized fuzzy division method,[2,3,25] summarized in the following section. Refs. 2 and 3 include a comparison with other fuzzy division approaches that have been published[8,12,16]. We then explain how fuzzy quantifiers may be used to relax the universal quantifier of the division. We include a comparison with other approaches[13,15,18] which use fuzzy quantifiers. We also define a new syntax based on the fuzzy SQL (FSQL) language,[2,26,27] to express any kind of fuzzy division. Finally, we offer some conclusions and future lines of work.

2. GENERALIZED FUZZY DIVISION WITH THE CLASSICAL UNIVERSAL QUANTIFIER

In Ref. 3, generalized fuzzy division was presented. This is a method used to calculate relational division in fuzzy databases, starting with the GEFRED model.[7] To define this generalized fuzzy division, two new operators are defined. It has been shown that these two operators are useful in other applications, providing solutions for questions other than those of fuzzy division.

Below, we will offer a brief explanation of these two operators and how they are used in generalized fuzzy division.

2.1. Qualified Fuzzy Intersection: $\cap_Q$

DEFINITION 3. *Let R and R' be two fuzzy relations such that both are base relations of the database and both have the same attributes $\{A_1, \ldots, A_n\}$ (compatible with respect to the union). Tuples of R are $\{\tilde{d}_{i1}, \ldots, \tilde{d}_{in}\}$ with $i = 1, \ldots, m$; tuples of R' are $\{\tilde{d}'_{k1}, \ldots, \tilde{d}'_{kn}\}$ with $k = 1, \ldots, m'$, with m and m' being the cardinalities of R and R', respectively.*

Then, the ***quantified fuzzy intersection*** *of R by R', denoted by $R \cap_Q R'$, will be equal to R, adding a compatibility degree to its attributes K_i with $i = 1, \ldots, m$ for each tuple in R.*

The value K_i is the possibility degree of tuple i in R existing in R'. The values K_i, with $i = 1, \ldots, m$, are computed individually in the following way: $\forall i = 1, \ldots, m$

$$K_i = \max_{w=1,\ldots,m'} \left\{ \min_{c=1,\ldots,n} \left\{ \Theta^{=} \left(\tilde{d}_{ic}, \tilde{d}'_{wc} \right) \right\} \right\} \tag{2}$$

where $\Theta^{=}$ is a fuzzy comparator representing the "fuzzy equal" or "possibly equal".

Some observations are:

- The operator $\Theta^{=}$ will be used to measure the equality of two values. One definition of this comparator, for possibility distributions, may be, for example,

$$\Theta^{=}(\tilde{p}, \tilde{p}') = \sup_{d \in U} \left\{ \min\left(\pi_{\tilde{p}}(d), \pi_{\tilde{p}'}(d) \right) \right\} \tag{3}$$

  where $\tilde{p}, \tilde{p}'$ are fuzzy values, and their associated possibility distributions are $\pi_{\tilde{p}}$ and $\pi_{\tilde{p}'}$, respectively. U is the discourse domain underlying the values. The definition of this comparator may be changed, as we will see below.

- If some attributes exist in R and R' with crisp domains, it is possible to improve the efficiency of computing K_i (Eq. 2). This is shown in Ref. 3.
- If we apply a threshold $u \in [0,1]$ to the K_i values in the resulting relation from the qualified fuzzy intersection, then we obtain the R tuples which belong to R' with a possibility greater than or equal to u.
- The qualified fuzzy intersection does not observe the commutative property. The operation $R \cap_Q R'$ returns the possibility of tuples in R belonging to R', and the operation $R' \cap_Q R$ returns the possibility of tuples in R' belonging to R. This operator is somewhat like an intersection in only one direction.

2.2. Fuzzy Projection with Group Functions $\mathscr{F}$: $\mathscr{P}^{\mathscr{F}}$

DEFINITION 4. *Let the following four elements be defined as follows:*

(1) *A fuzzy relation R, with attributes $\{A_1, \ldots, A_n\}$ and tuples $\{\tilde{d}_{r1}, \ldots, \tilde{d}_{rn}\}$ with $r = 1, \ldots, m$.*
(2) *A list X of R attributes:*

$$X = \{x_1, \ldots, x_\alpha\}: \ x_i \in \{A_1, \ldots, A_n\}, \qquad \forall i = 1, \ldots, \alpha$$

It is possible that there are elements repeated in X but in different positions.

(3) *A subset X' of R attributes with crisp domains*:

$$X' \subseteq \{A_1, \ldots, A_n\}:\ X' = \{x'_1, \ldots, x'_\beta\}$$

In X', like any set, repeated elements are not accepted.

(4) *A group function list, $\mathcal{F}$, defined over the elements of X. The list $\mathcal{F}$ has α elements, where each element is a group function (not necessarily distinct) defined over each attribute in X, respectively*:

$$\mathcal{F} = \{\mathrm{gfunc}_1, \ldots, \mathrm{gfunc}_\alpha\}$$

We use the name "group functions" to refer to those functions which operate on a finite group of values returning only one value. The most typical examples are those used to count the number of elements in the group (Lcount), *compute the maximum value* (max), *the minimum* (min), *the sum* (sum), *the average* (avg), *the variance* (variance), *and the standard deviation* (stddev). *All these functions are defined in SQL language.*

We now perform the fuzzy projection of R onto X': $R' = \mathcal{P}(R; X')$ with tuples $\{\tilde{d}'_{t1}, \ldots, \tilde{d}'_{t\beta}\}$, with $t = 1, \ldots, m'$, m' being the projection number of tuples. This projection does not cause any problems since the X' attributes have crisp domains.

Using the former elements, we then define the ***fuzzy projection of*** *R* ***onto*** *X',* ***with group functions $\mathcal{F}$ onto*** *X, as the fuzzy relation $\mathcal{P}^{\mathcal{F}}(R; X'; X)$, with all attributes in X' and X*:

- *The values in X' attributes are the same as the values of R'. Then $\mathcal{P}^{\mathcal{F}}(R; X'; X)$ also has m' tuples.*
- *The values in X attributes are denoted by $\tilde{d}^{\mathcal{F}}_{ti}$, $\forall t = 1, \ldots, m'$, $\forall i = 1, \ldots, \alpha$, and they are computed using the following equation*:

$$\tilde{d}^{\mathcal{F}}_{ti} = \underset{R'.\tilde{d}'_{tj} = R.\tilde{d}'_{rj},\ \forall j = 1, \ldots, \beta}{\mathrm{gfunc}_i} \left\{R.\tilde{d}_{ri}:\ r = 1, \ldots, m\right\} \tag{4}$$

This equation computes the group function gfunc_i *on values of attribute $x_i \in X$ of R of tuples whose X' attributes are equal to X' attributes in R'. This equality is in the classic sense since X' elements are on crisp domains.*

In Ref. 3 an algorithm appears which implements Eq. 4 and it has been shown that the fuzzy projection with group functions power allows us to easily solve questions which would be more complicated with other methods. For example, it includes an example to solve the question of "*Which students are good (with a* 0.8 *minimum degree) in* 2 *or more subjects*?" Using the same technique, it is possible to solve more complicated questions such as "*Which students are good in* 2 *subjects and bad in* 3 *subjects*?"

In generalized fuzzy relational algebra, this fuzzy projection of R onto X', with group functions $\mathcal{F}$ onto X, $\mathcal{P}^{\mathcal{F}}(R; X'; X)$, models what in SQL is performed with the GROUP BY clause in a SELECT statement with group functions. In a query containing a GROUP BY clause, all elements of the SELECT list must be either expressions of the GROUP BY clause, expressions containing group functions, or constants. So, the projection $\mathcal{P}^{\mathcal{F}}(R; X'; X)$ may be easily translated to a SQL SELECT statement, operating each clause on the following elements:

- SELECT: The selected elements will be the set of X' attributes and each group function of $\mathcal{F}$ on each X attribute, respectively. Then, the SELECT list will have $\alpha + \beta$ elements.
- FROM: The relation R.
- GROUP BY: All the X' attributes will appear in this clause.

It should be noted that the X attributes may have fuzzy domains, and in such a case, the group functions $\mathcal{F}$ corresponding to those attributes must be defined over those domains. We will therefore be able to compute the minimum value of a fuzzy number group, the maximum, etc. Some other interesting considerations are included in Ref. 3.

2.3. Generalized Fuzzy Relational Division: ÷

We now generalize the relational division (Definition 1) for fuzzy databases. We can extend this definition in the same way as Definition 2.

DEFINITION 5. *Let R and R' be two fuzzy relations with attribute sets $\{A_1, \ldots, A_{n'}, \ldots, A_n\}$ and $\{A_{n'+1}, \ldots, A_n\}$, respectively. Tuples of R are $\{\tilde{d}_{i1}, \ldots, \tilde{d}_{in}\}$ with $i = 1, \ldots, m$, tuples of R' are $\{\tilde{d}'_{k(n'+1)}, \ldots, \tilde{d}'_{kn}\}$ with $k = 1, \ldots, m'$, m and m' are the cardinalities of R and R', respectively, and n and $n - n'$ are the respective degrees.*

Furthermore, $1 \leq n' < n$, and $\{A_1, \ldots, A_{n'}\}$ have crisp domains. In Ref. 3, some remarks appear if these attributes are not crisp.

We then define the ***generalized fuzzy relational division*** *of R by R', denoted by $R \div R'$, as another fuzzy relation obtained by the following three operations*:

(1) *We calculate a relation R'' by*:

$$R'' = R[A] \times R' \tag{5}$$

with $A = \{A_1, \ldots, A_{n'}\}$.

(2) *We calculate the qualified fuzzy intersection of R'' by R*:

$$R''' = R'' \cap_Q R \tag{6}$$

(3) *We calculate the Generalized Fuzzy Projection of R''' onto A, with Group Functions $\mathcal{F}$ onto C*:

$$R \div R' = \mathcal{P}^{\mathcal{F}}(R'''; A; C) \tag{7}$$

where C is the compatibility attribute computed in Eq. 6, and $\mathcal{F}$ is the minimum group function (min).

It should be noted that in Eq. 5, the projection onto A, $R[A]$, does not create any problems, since we have supposed that all A attributes have crisp domains. The operator $\times$ is the cartesian product. In Eq. 7, the projection with group functions does not create any problems, since both A and C are attribute sets with crisp domains.

The generalized fuzzy relational division is an extension of the classic relational division on crisp attributes and this is therefore included. In classic relations (without fuzzy attributes), both divisions obtain the same results (but in a different form). In other words, this is also another method to calculate relational division in classic databases. In Ref. 3 there is justification for generalized fuzzy division, a comparison with the classic relational division formula, some possible problems and their solutions, and a comparison with other fuzzy division approaches.[8,12,16]

3. USING FUZZY QUANTIFIERS

In the previous section we have shown a method for fuzzy division with fuzzy relations. In this method only the universal quantifier is considered. However, it is very restrictive and it is useful to perform the fuzzy division using both relative and absolute fuzzy quantifiers.[19]

Relative quantifiers depend on the number of existing tuples in the denominator relation (R'), and they are depicted by fuzzy sets over the real interval $[0, 1]$. Examples of relative quantifiers are "all" (universal quantifier, $\forall$), "almost all", "most" ("the majority"), "approximately half", and "the minority". Absolute quantifiers do not depend on that number and they are defined as fuzzy sets over the nonnegative integers. Examples of absolute quantifiers are "one or more" (existential quantifier, $\exists$), "many", "very few", "approximately 5", "more than 5", and "a lot more than 5".

DEFINITION 6. *Let R and R' be two fuzzy relations defined as in Definition 5 and let Q be a fuzzy quantifier. The fuzzy division using Q is therefore denoted by $R \div^{Q} R'$, and it selects tuples of the first relation which are related, in some way, to Q of tuples in the second relation. The* ***generalized fuzzy relational division with fuzzy quantifier*** *Q is the same as that shown in Definition 5, but we must change the definition of $\mathscr{F}$ in Equation 7 according to the fuzzy quantifier Q:*

(1) *Universal quantifier "all" ($Q = \forall$): $\mathscr{F} = \{\min\}$.*
(2) *Existential quantifier "one or more" ($Q = \exists$): $\mathscr{F} = \{\max\}$.*
(3) *Fuzzy absolute quantifier Q: The group function sum is applied and the quantifier Q is then applied on this value. This is represented by $\mathscr{F} = \{Q(\mathrm{sum})\}$.*
(4) *Fuzzy relative quantifier Q: The group function average is applied and the quantifier Q is then applied on this value. This is represented by $\mathscr{F} = \{Q(\mathrm{avg})\}$.*

Of course, this definition may be extended according to Definition 2 and, in this case, it is denoted by

$$R \div^{Q}_{A,B,X} R' \tag{8}$$

To evaluate the fuzzy quantifier, it is possible to use various methods to evaluate sentences with fuzzy quantifiers.[23,24] However, we have tested them and the results are not good enough, because fuzzy division has its own semantic.

3.1. Example

Example 1. Let us suppose that we have a fuzzy relational database about basketball players. A database relation may have the attributes (PLAYER, TEAM, HEIGHT, QUALITY, NUM_SHIRT...). The fields HEIGHT (where the player's height is stored) and QUALITY (where the player's quality is measured according to his average points per match) allow fuzzy values (type 6 in Table I). For the sake of the example, we will use the linguistic labels in Figure 1.

We have eliminated the labels "Very Short" and "Very Bad", since in our opinion, professional players with these characteristics do not exist.

In this context, we are going to find those basketball teams whose player types (in HEIGHT and QUALITY) match those of the team from Córdoba (using different fuzzy quantifiers).

To find these teams, we first take a projection of the previous relation onto the interesting attributes (TEAM, HEIGHT, and QUALITY), giving a relation R which can be seen in Table II. Furthermore, the second relation R' will be the projection onto the HEIGHT and QUALITY attributes after the selection

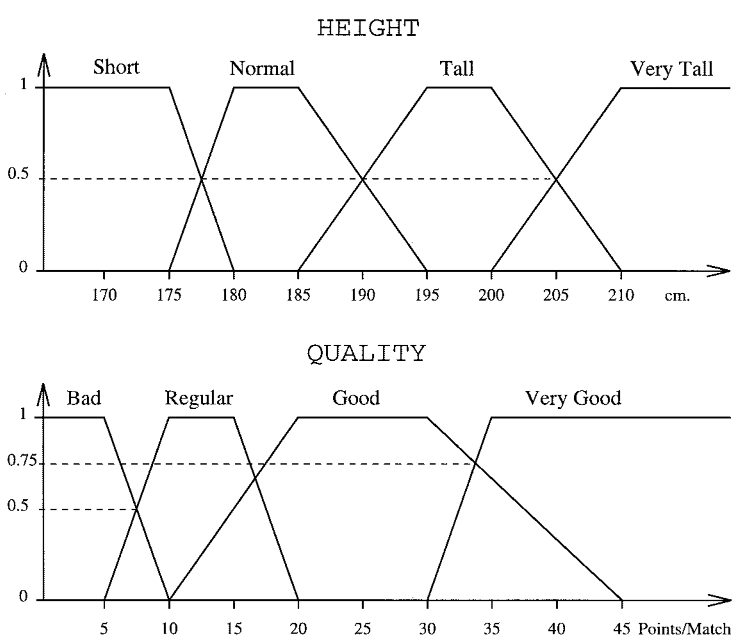

Figure 1. Example 1: definition of labels on HEIGHT and QUALITY attributes.

Table II. Example 1: relation R.

TEAM	HEIGHT	QUALITY
Córdoba	Short	Very Good
Córdoba	Very Tall	Bad
Granada	Short	Very Good
Granada	Very Tall	Bad
Granada	Tall	Regular
Málaga	Short	Very Good
Málaga	Tall	Bad
Málaga	Very Tall	Very Good
Sevilla	Short	Good
Sevilla	Very Tall	Bad
Sevilla	Normal	Good
Cádiz	Very Tall	Very Good
Cádiz	Short	Good
Almería	Tall	Very Good
Almería	Short	Regular

with the condition TEAM = Córdoba. In our example, R' has only two tuples. This relation is shown in Table III.

Thus, this fuzzy division may be expressed by using Eq. 8 and the definition given in Eq. 1:

$$R \div^{Q}_{\text{TEAM},\{\text{HEIGHT},\text{QUALITY}\},\varnothing} R' \tag{9}$$

where Q is the fuzzy quantifier and takes the following values in this example:

(1) Existential quantifier ($\exists$).
(2) Fuzzy absolute quantifier "approximately 2" defined by the following triangular function (2 ± 1):

$$Q(x) = \begin{cases} 0 & \text{if } x \leq 1 \text{ or } x \geq 3 \\ x - 1 & \text{if } 1 < x \leq 2 \\ 3 - x & \text{if } 2 < x < 3 \end{cases}$$

(3) Fuzzy relative quantifier "most", defined by: $Q(x) = x$.
(4) Fuzzy relative quantifier "almost all", defined in Figure 2.
(5) Universal quantifier ($\forall$).

Table III. Example 1: relation R'.

HEIGHT	QUALITY
Short	Very Good
Very Tall	Bad

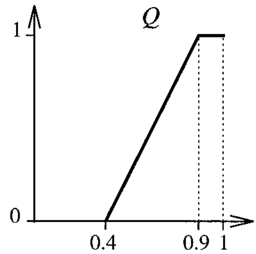

Figure 2. Fuzzy relative quantifier "almost all": $x \in [0.4, 0.9] \leftrightarrow y = 2(x - 0.4)$.

To calculate $R \div^{Q} R'$, we will compute the following equations:

(1) **Equation 5**. We calculate R'' by the following expression obtaining the relation shown in Table IV:

$$R'' = R[\text{TEAM}] \times R' \tag{10}$$

(2) **Equation 6**. We calculate the qualified fuzzy intersection, $R'' \cap_Q R$, and we obtain the fuzzy relation R''' in Table V. The C_{TEAM} values indicate the possibility ("compatibility degree") of tuples in R'' belonging to R. In this table, we have included those operations whereby the values were obtained by applying Eq. 2, avoiding the attribute TEAM because it is not a fuzzy attribute (see Ref. 3 for details). The values in min functions are the results of the fuzzy comparator $\Theta^{=}$, defined in Eq 3, applied on HEIGHT and QUALITY fuzzy attributes.

(3) **Equation 7**. Finally, following the fuzzy division general process, we perform the generalized fuzzy projection of R''' onto the TEAM attribute, with group function $\mathscr{F}$ onto C_{TEAM}:

$$R \div R' = \mathscr{P}^{\mathscr{F}}(R'''; \text{TEAM}; C_{\text{TEAM}}) \tag{11}$$

Table IV. Example 1: relation $R'' = R[\text{TEAM}] \times R'$.

TEAM	HEIGHT	QUALITY
Córdoba	Short	Very Good
Córdoba	Very Tall	Bad
Granada	Short	Very Good
Granada	Very Tall	Bad
Málaga	Short	Very Good
Málaga	Very Tall	Bad
Sevilla	Short	Very Good
Sevilla	Very Tall	Bad
Cádiz	Short	Very Good
Cádiz	Very Tall	Bad
Almería	Short	Very Good
Almería	Very Tall	Bad

Table V. Example 1: relation $R''' = R'' \cap_Q R$, with C_{TEAM} calculation.

TEAM	C_{TEAM}		(C_{TEAM} calculation)	HEIGHT	QUALITY
Córdoba	1	=	max{min{1,1}, min{0,0}}	Short	Very Good
Córdoba	1	=	max{min{0,0}, min{1,1}}	Very Tall	Bad
Granada	1	=	max{min{1,1}, min{0,0}, min{0,0}}	Short	Very Good
Granada	1	=	max{min{0,0}, min{1,1}, min{0.5,0.5}}	Very Tall	Bad
Málaga	1	=	max{min{1,1}, min{0,0}, min{0,1}}	Short	Very Good
Málaga	0.5	=	max{min{0,0}, min{0.5,1}, min{1,0}}	Very Tall	Bad
Sevilla	0.75	=	max{min{1,0.75}, min{0,0}, min{0.5,0.75}}	Short	Very Good
Sevilla	1	=	max{min{0,0}, min{1,1}, min{0,0}}	Very Tall	Bad
Cádiz	0.75	=	max{min{0,1}, min{1,0.75}}	Short	Very Good
Cádiz	0	=	max{min{1,0}, min{0,0}}	Very Tall	Bad
Almería	0	=	max{min{0,1}, min{1,0}}	Short	Very Good
Almería	0	=	max{min{0.5,0}, min{0,0.5}}	Very Tall	Bad

The final results from fuzzy division with all these quantifiers are shown in Table VI. For example, the values $C_{\text{TEAM}}^{\text{most}}$ indicate a degree which means at what level the associated TEAM has the same player types as "most" of the player types of the team from Córdoba.

It is possible to apply a threshold onto the relation resulting from generalized fuzzy division, such as $C_{\text{TEAM}}^{Q} \geq u$, where $u \in [0, 1]$.

It is easy to see that this method is useful for carrying out relational division with fuzzy quantifiers in classical databases (without fuzzy attributes).

4. COMPARISON WITH OTHER APPROACHES

In the references we can find various approaches for carrying out relational division in fuzzy databases. These approaches generally use a model of fuzzy relations or a meaning of the degrees that is different from the one we use (see Section 1). The methods in Refs. 8, 9, 12, and 16 do not consider fuzzy quantifiers and a comparison with Refs. 8, 12, and 16 is included in Ref. 3.

Table VI. Example 1: resulting relation from $R \div^{Q} R'$ with different values for Q.

TEAM	$C_{\text{TEAM}}^{\exists}$	$C_{\text{TEAM}}^{\text{approx. 2}}$	$C_{\text{TEAM}}^{\text{most}}$	$C_{\text{TEAM}}^{\text{almost all}}$	$C_{\text{TEAM}}^{\forall}$
Córdoba	1	1	1	1	1
Granada	1	1	1	1	1
Málaga	1	0.5	0.75	0.7	0.5
Sevilla	1	0.75	0.875	0.95	0.75
Cádiz	0.75	0	0.375	0	0
Almería	0	0	0	0	0

The method of Nakata[9] is the evolution of Refs. 4, 8, and 17. It is interesting because it considers fuzzy values in the attributes through possibility distributions (like Refs. 3 and 4) and two elements in some domains have a degree of resemblance to each other (like Refs. 3 and 8). Furthermore, a relation has a membership attribute for every tuple (like Ref. 17) and it is composed of two values: necessity and possibility measures. It uses the Dienes or Gödel implication over the membership degrees and the possibly and necessarily equal measures computed using the possibility distributions. However, it does not consider fuzzy quantifiers and it only allows one common attribute for both relations of the division.

Our approach may consider degrees, whatever their meanings, but these degrees are seen as fuzzy values themselves and the only necessary thing is for there to be a comparison function $\Theta^=$ for those types (which considers the meaning of the degrees). As we show below, this technique obtains very intuitive results.

The methods in Refs. 13, 15, and 18 use fuzzy quantifiers and in this section we analyze and compare them with our generalized fuzzy division.

4.1. The Yager Division

In Ref. 18, Yager showed a fuzzy division based on fuzzy relations with only one ownership degree for each tuple. This method does not allow fuzzy values in the attributes which are different from this ownership degree in the $[0,1]$ interval. This method allows fuzzy quantifiers to be used to relax the classical universal quantifier in relational division and it is based on OWA operators introduced by Yager himself in Ref. 20:

DEFINITION 7. *An OWA operator of dimension n is a function F,*

$$F\colon [0,1]^n \to [0,1] \tag{12}$$

that has a set of weights $w_i \in [0,1]$ *associated with it, with* $i = 1, \ldots, n$, *such that*

$$\sum_{i=1}^{n} w_i = 1$$

and where for any argument $(a_1, \ldots, a_n)$,

$$F(a_1, \ldots, a_n) = \sum_{i=1}^{n} b_i * w_i \tag{13}$$

with b_i *being the ith largest of the* a_i: $b_1 \geq b_2 \geq \cdots \geq b_n$.

To summarize, it is an intermediate method between taking the minimum and taking the maximum of $(a_1, \ldots, a_n)$. Thus, if the weights are $(w_1, \ldots, w_n) = (0, \ldots, 0, 1)$, then the function $F(a_1, \ldots, a_n)$ will take the minimum value of a_i, and if the weights are $(1, 0, \ldots, 0)$, then the function will take the maximum value.

We will show the Yager method for fuzzy division by means of the following example:

Example 2. Let R and S be the relations in Table VII and VIII, respectively. The fuzzy relation S stores the required manual dexterity. Thus, for each skill type α indicates the degree to which the skill requires manual dexterity.

Let the fuzzy relative quantifier "most" be defined simply as $Q(r) = r$, with $r \in [0, 1]$. Then, the query "*find the people who have most of the skills that require manual dexterity*" is solved through the division R by S:

(1) Find the people in R: this is a simple projection of R onto the "Name" attribute obtaining: {Jean, Barbara, Debbie, Tina, Patricia}.

Table VII. Example 2 of Yager division: relation R.

Name	Skill	α
Jean	I	1.0
Jean	II	0.7
Jean	III	0.5
Barbara	I	0.3
Barbara	II	0.6
Debbie	I	1.0
Debbie	II	0.7
Debbie	III	0.5
Debbie	IV	0.2
Tina	II	1.0
Patricia	I	1.0
Patricia	II	0.8
Patricia	III	0.2

Table VIII. Example 2 of Yager division: relation S.

Skill	α
I	1.0
II	0.8
III	0.2
IV	0.0

(2) For each element u obtained above find R_u^+ as a projection of R onto the attributes which are not in u.
(3) Evaluate the fulfillment degree of sentence "*Q Ss are R_u^+*", i.e., "*most of the elements in S are in R_u^+*":
   (a) We sort the degrees in S from the smallest to greatest and we add up all of them in d:

$$e_1 = 0,\ e_2 = 0.2,\ e_3 = 0.8,\ e_4 = 1, \text{ and } d = 2.$$

   (b) We compute values S_j, establishing that $S_0 = 0$ and

$$S_j = \frac{e_j}{d} + S_{j-1} \tag{14}$$

   We obtain $S_1 = 0$, $S_2 = 0.1$, $S_3 = 0.5$, and $S_4 = 1$.
   (c) We compute the weights:

$$w_j = Q(S_j) - Q(S_{j-1}) \tag{15}$$

   In this case, $w_j = e_j/d$ with $j \in \{1,2,3,4\}$: $w_1 = 0$, $w_2 = 0.1$, $w_3 = 0.4$, and $w_4 = 0.5$.
   (d) For each $u \in$ {Jean, Barbara, Debbie, Tina, Patricia}:
      (i) Compute C_i, with $i \in$ {I, II, III, IV}:

$$C_i = \max\{1 - \alpha_i, R_u^+(i)\} \tag{16}$$

      where α_i and $R_u^+(i)$ are the degrees of element i in S and R_u^+, respectively.
      (ii) Sort C_i from the greatest to the smallest: $b_1 \geq b_2 \geq b_3 \geq b_4$.
      (iii) Compute the fulfillment degree that we are looking for by

$$T(u) = \sum_{j=1}^{4} b_i * w_i \tag{17}$$

The resulting relation is shown in Table IX with α^{most}. The value for $\alpha^{\forall}$ indicates the results if we use the quantifier $\forall$, defined by $Q(r) = 0$ if $r \neq 1$ and $Q(1) = 1$. This last quantifier takes the minimum of C_i, and it coincides with the

Table IX. Examples 2 and 3: comparing Yager division, Dubois division, and our generalized fuzzy (G.F.) division.

	Yager Division		G.F. Division		Dubois Division	
Name	α^{most}	$\alpha^{\forall}$	$C^{\text{most}}_{\text{Name}}$	$C^{\forall}_{\text{Name}}$	α^{most}	$\alpha^{\forall}$
Jean	0.77	0.70	0.93	0.70	0.75	0.70
Barbara	0.47	0.30	0.48	0.00	0.50	0.00
Debbie	0.77	0.70	0.93	0.70	0.75	0.70
Tina	0.42	0.00	0.50	0.00	0.50	0.00
Patricia	0.82	0.80	1.00	1.00	1.00	1.00

necessity measure between the R values with respect to S values for a fixed "Name".

4.1.1. Analyzing the Yager Approach

The Yager division relaxes the universal quantifier, allowing fuzzy quantifiers such as "most" in the previous example to be used. This relaxation is carried out using the OWA operators. However, Yager only considers monotone quantifiers in Ref. 20 and in Ref. 18 only increasing quantifiers are studied. Even the existential quantifier ($\exists$), defined by $Q(r) = 1, \forall r$, cannot be used, because it obtains all weights equal to 0 (Eq. 15). In our fuzzy division approach, we can use any fuzzy quantifier, even nonmonotone quantifiers.

Yager uses an FRDB model which is very similar to the Mouaddib model,[8] and it only allows the representation of scalar values with an associated degree (α). This data type is also considered by the GEFRED model.[7] Then, to apply our approach to this data type, we have only to define the comparator $\Theta^{=}$ for that data type. For example, we can define it using the Gödel implication:

$$a \to b = \begin{cases} 1 & \text{if } a \leq b \\ b & \text{otherwise} \end{cases} \tag{18}$$

where a is the value α in S and b is the value α in R. The resulting relation is shown in Table IX.

Some results deserve a detailed survey because, for example, "Patricia" completely fulfills the requirements expressed in relation S although she only obtains $\alpha^{\forall} = 0.8$ and $\alpha^{\text{most}} = 0.82$. We think that these values are too small and they are not what we would intuitively expect. With the previous definition of the Gödel implication, our approach obtains the value 1 for "Patricia" with both quantifiers. "Barbara" obtains $\alpha^{\forall} = 0.3$, but she does not have skill III and if we use the quantifier "all" ($\forall$) we are looking for people who fulfill all requirements. Our approach obtains the value 0 for "Barbara" with quantifier "all". "Jean" and "Debbie" satisfy the requirements except for skill II. They therefore really satisfy most of the requirements, but they only obtain $\alpha^{\text{most}} = 0.77$. Our approach obtains the values 0.93.

Moreover, the Yager division only allows one common attribute to both relations, whereas in our approach we can have any number of common attributes for both relations (with different degree α).

If we apply this method to classic databases, then we obtain the same results as in the generalized fuzzy division.

4.2. The Dubois et al. Division

In Ref. 15 Dubois, Nakata, and Prade propose a method for fuzzy division which is similar to the method presented in Ref. 16 (analyzed in Ref. 3), but they relax the universal quantifier by using both absolute and relative fuzzy quantifiers. It is based on the use of a different kind of fuzzy implication, depending on the meaning of the degree:

(1) Fulfillment degree: They use R-implications (i.e., a residuated implication: $a \to b = \sup\{x \in [0,1]: a * x \le b\}$, where $*$ is a triangular norm§). Thus, they use Gödel,‖ Goguen,¶ Rescher-Gaines,** or Lukasiewicz†† implications.
(2) Importance degree: They use S-implications (i.e., an implication: $a \to b = (1 - a) \perp b$ where $\perp$ is a triangular co-norm‡‡). Thus, they use Dienes,§§ Reichenbach,‖‖ or Lukasiewicz implications.

In the fuzzy division $R(t,u) \div S(u)$ the degree of every value t in the solution is computed by

$$\alpha_{R \div S}(t) = \min_u \left(\alpha_S(u) \to \alpha_R(t,u) \right) \tag{19}$$

An absolute fuzzy quantifier Q must be increasing and $Q(m) = 1$, where m is the number of requirements (or tuples) in S. They then associate a fuzzy set I_Q defined by $I_Q(i) = 1 - Q(i-1)$. Thus, in the fuzzy division $R(t,u) \div_Q S(u)$ the degree of every value t in the solution is computed by

$$\alpha_{R \div_Q S}(t) = \min_i \max\left(\alpha_S(u) \to_{\sigma(i)} \alpha_R(t,u),\ 1 - I_Q(i) \right) \tag{20}$$

where $\alpha_S(u) \to_{\sigma(i)} \alpha_R(t,u)$ expresses a ranking such that $\alpha_S(u) \to_{\sigma(1)} \alpha_R(t,u) \ge \alpha_S(u) \to_{\sigma(2)} \alpha_R(t,u) \ge \cdots \alpha_S(u) \to_{\sigma(m)} \alpha_R(t,u)$.

If the quantifier is a relative one then

$$\alpha_{R \div_Q S}(t) = \min_i \max\left(\alpha_S(u) \to_{\sigma(i)} \alpha_R(t,u),\ Q((i-1)/m) \right) \tag{21}$$

Example 3. Following Example 2, let R and S be the relations in Tables VII and VIII, respectively. Then, the resulting relations using the Gödel implication and the quantifiers "most" and ∀ are shown in Table IX.

For example, for "Jean" (or "Debbie") we first compute the Gödel implication, obtaining values {1/I, 0.7/II, 1/III, 1/IV}. With these values and the quantifier "most", the computation of Eq. 21 is min{max(1, 0), max(1, 0.25),

§ A T-norm is a function $(a,b) \in [0,1]^2 \mapsto a * b \in [0,1]$ where $*$ is associative and symmetrical, increasing each argument in the broad sense w.r.t., and it satisfies the limit conditions $a * 1 = a$, $\forall a$ and $0 * 0 = 0$. Representative T-norms are $a * b = \min(a,b)$, $a * b = ab$, and $a * b = \max(0, a + b - 1)$.

‖ Gödel implication: $a \to b = 1$ if $a \le b$, and $a \to b = b$ if $a > b$.

¶ Goguen implication: $a \to b = \min(1, b/a)$ if $a \ne 0$, and $a \to b = 1$ if $a = 0$.

** Rescher-Gaines implication: $a \to b = 1$ if $a \le b$, and $a \to b = 0$ if $a > b$.

†† Lukasiewicz implication: $a \to b = \min(1, 1 - a + b)$.

‡‡ A T-conorm $\perp$ is associated with a T-norm $*$ by the duality relation $a \perp b = 1 - (1-a) * (1-b)$. The main T-conorms are $a \perp b = \max(a,b)$, $a \perp b = a + b - ab$, and $a \perp b = \min(1, a + b)$.

§§ Dienes implication: $a \to b = \max(1 - a, b)$.

‖‖ Reichenbach implication: $a \to b = 1 - a + ab$.

max(1, 0.5), max(0.7, 0.75)} = 0.75. For “Barbara” the computation is min{max(1, 0), max(0.6, 0.25), max(0.3, 0.5), max(0, 0.75)} = 0.5.

This method includes the case “where levels of importance are attached to the requirements expressing the satisfaction of a specified minimal level of fulfillment in the divisor” S. Namely, relation S may have two degrees per tuple: a fulfillment degree and an importance degree. Moreover, Dubois et al.[15] study the case when fulfillment degrees in relation R do not contain precise values and these values are pervaded with uncertainty and imprecision, allowing fuzzy valued degrees over the interval [0, 1].

4.2.1. Analyzing the Dubois et al. Approach

Some of the drawbacks of this method are that it only studies increasing quantifiers and that absolute quantifiers must be defined according to relation S, because it has the requisite $Q(m) = 1$.

Regarding the Yager approach, we can see that this method does not have the “Patricia” problem in Example 3. However, it maintains the “Jean” and “Debbie” problems, since they fulfill the requirements except for the skill II. Namely, they really satisfy “most” of the requirements, but they only obtain 0.75. With the quantifier ∀, this method obtains the same values as our method.

Moreover, in the same way as the Yager division, this approach only allows one common attribute to both relations and in classic databases they obtain the same results as in the generalized fuzzy division.

4.3. The Vila et al. Division

In Ref. 13, Vila et al. propose a new system for the fuzzy division $R(A, B) \div S(B)$ in FRDBs with the possibility-based model; i.e., it is possible to store possibility distributions as attributes values.

This method is based on the “compression” of relations R and S, such that the compression of R on B, $\delta_B(R)$, includes tuples with different values for A attributes and with all the values for B “compressed” in one value:

$$\forall t \in \delta_B(R), t[B] = \{r[B]: r \in R, r[A] = t[A]\}$$

Possibility distributions of B are compressed in one possibility distribution for each A value, taking the maximum value in all distributions for each value of the underlying domain. Relation S is compressed in the same way, and it will have only one tuple with the possibility distribution P.

Then, fuzzy division is carried out in the following way:

$$\pi_A(\sigma \Gamma^Q_{B \simeq P}(\delta_B(R))) \tag{22}$$

where π_A is the projection onto A attributes and $\sigma \Gamma^Q_{B \simeq P}$ is a generalized selection, the so-called $\simeq$-selection, depending on the fuzzy measure Γ^Q

(representing the accomplishment degree of the property to match with Q of objects described by P).

Briefly, for a nondecreasing fuzzy quantifier Q, a measure called α_Q is obtained. For example, Q may be expressed as:

$$Q(x) = \begin{cases} 0 & \text{if } 0 \le x \le a_Q \\ \dfrac{(x - a_Q)}{(b_Q - a_Q)} & \text{if } a_Q < x < b_Q \\ 1 & \text{if } b_Q \le x \le 1 \end{cases} \tag{23}$$

where $a_Q, b_Q \in [0,1]$, and $a_Q \le b_Q$. These two values represent the quantifier. Thus, the quantifier $\exists$ has the values $(0,0)$ and the quantifier $\forall$ has the values $(1,1)$. Then, α_Q is obtained by

$$\alpha_Q = (b_Q - a_Q)/2 + 1 - b_Q \tag{24}$$

With these data, they determine that the selection $\sigma\Gamma^{Q}_{B \simeq P}$ from the compressed relation $\delta_B(R)$ adds a degree which is computed by

$$\alpha_Q \Pi(B|P) + (1 - \alpha_Q) N(B|P) \tag{25}$$

where $\Pi(A|P)$ and $N(A|P)$ are possibility and necessity measures, respectively, with which B is matched with P.

Example 4. Let R be the relation in Table X, where $S\#$ is a fuzzy attribute. Relation R is compressed, obtaining the relation in Table XI. Let P be the possibility distribution obtained by compressing a relation S:

$$P = \{0.5/S1, 0.7/S2, 1/S3\}$$

Table X. Example 4 of Vila et al. division: relation R.

P#	S#
P1	0.8/S1, 0.5/S2
P1	0.7/S1, 0.6/S2
P1	0.8/S3, 0.7/S4
P2	1/S1
P2	1/S4
P3	0.6/S1, 0.9/S3
P3	1/S2
P4	0.7/S1, 0.2/S3
P4	1/S2
P4	1/S3
P4	0.6/S1, 0.5/S4
P5	1/S3
P6	0.5/S1, 0.7/S2, 1/S3

Table XI. Example 4 of Vila et al. division: compressed relation R.

P#	S#
P1	0.8/S1, 0.6/S2, 0.8/S3, 0.7/S4
P2	1/S1, 1/S4
P3	0.6/S1, 1/S2, 0.9/S3
P4	0.7/S1, 1/S2, 1/S3, 0.5/S4
P5	1/S3
P6	0.5/S1, 0.7/S2, 1/S3

For example, let us consider the following four quantifiers:

(1) Existential quantifier "one or more" ($\exists$) with its associated values $(0, 0)$ and $\alpha_\exists = 1$.
(2) "Most", with its values $(0, 1)$ and $\alpha_{\text{most}} = 0.5$.
(3) "Almost all" (Fig. 2), with its values $(0.4, 0.9)$ and $\alpha_{\text{almost all}} = 0.35$.
(4) The classical quantifier for the division "all" ($\forall$) with its values $(1, 1)$ and $\alpha_\forall = 0$.

The results for these quantifiers are shown in Table XII. For example, for P1 we compute the following values for each quantifier:

$$\Gamma^{\exists}(P1) = \max\{\min\{0.5, 0.8\}, \min\{0.7, 0.6\}, \min\{1, 0.8\}, \min\{0, 0.7\}\} = 0.8$$
$$\Gamma^{\forall}(P1) = \min\{\max\{1 - 0.5, 0.8\}, \max\{1 - 0.7, 0.6\}, \max\{1 - 1, 0.8\}, \max\{1 - 0, 0.7\}\} = 0.6$$
$$\Gamma^{\text{most}}(P1) = (0.5 * 0.8) + (1 - 0.5) * 0.6 = 0.7$$
$$\Gamma^{\text{almost all}}(P1) = (0.35 * 0.8) + (1 - 0.35) * 0.6 = 0.67$$

We use the T-conorm of the maximum and the T-norm of the minimum to compute the possibility and necessity measures.

Therefore, if we have the values for $\Gamma^{\exists}$ (possibility) and for $\Gamma^{\forall}$ (necessity), then it is easy to compute the values for any quantifier Q, if we have the value α_Q.

Table XII. Example 4: resulting relation of Vila et al. division.

P#	$\Gamma^{\exists}$	Γ^{most}	$\Gamma^{\text{almost all}}$	$\Gamma^{\forall}$
P1	0.80	0.70	0.67	0.670
P2	0.50	0.25	0.18	0.175
P3	0.90	0.75	0.71	0.705
P4	1.00	0.85	0.81	0.805
P5	1.00	0.65	0.55	0.545
P6	1.00	0.75	0.68	0.675

4.3.1. Analyzing the Vila et al. Approach

The Vila et al. division uses a possibility-based model for the FRDB, allowing possibility distribution, but it does not consider an important data type, scalars with a similarity relation between them. Furthermore, it allows the universal quantifier to be relaxed with any nondecreasing fuzzy quantifier Q, like "most" or "almost all" in the previous example. However, the unsolved problem is to compute α_Q for any fuzzy quantifier when it is not in the format of Equation 23.

Thus, the Vila et al. division only considers fuzzy quantifiers with a nondecreasing function Q and they must be in the format of Equation 23. This is too restrictive, because some fuzzy quantifiers are inherently decreasing (global or locally).

This approach does not define the method when there are more than one common attributes to both relations, but it includes an idea about this question. Our approach allows any number of common attributes to both relations.

In short, it obtains the set of those A elements which have "similarities" between its compressed B elements and P (compressed relation S). These "similarities" may be seen as a possibility or necessity measurement or a combination of these two measurements.

Some results deserve a detailed survey since, for example, "P6" completely fulfills the requirements expressed in P, i.e., $S\# = P$, yet it only obtains $\Gamma^{\forall} = 0.5$. We think that this value is too small with regard to the intuitively expected value.

In conclusion, the Vila et al. division is not really a relation division, but rather a "generalized selection", as they call it. This selection has a similar meaning with regard to relational division. Moreover, this approach is interesting because it contributes two operations, the compression and the generalized selection, which are very useful in flexible queries to FRDBs.

It is easy to see the difference between this method and the approach presented here in the following example:

Example 5. Let R be the relation in Table II (Example 1) and let R' be the relation in Table III, but we will only consider the QUALITY attribute.

All results for $R \div R'$, with our generalized fuzzy division and the Vila et al. division presented here, are included in Table XIII, using the four quantifiers used in Example 4: "one or more", "most", "almost all", and "all".

We can see that the results of generalized fuzzy division are closer to what we would intuitively expect. Teams from "Córdoba", "Granada", and "Málaga" only obtain $\Gamma^{\forall} = 0.5$ and they comply exactly with the division requirements. Moreover, generalized fuzzy division distinguishes among cases of "Sevilla", "Cádiz", and "Almería" teams, with different degrees in each team. These degrees naturally depend on the fuzzy quantifier. The values for the team from "Sevilla" with the quantifier "almost all" are interesting, since this team has one "Bad" player (exactly equal to one value in R') and two "Good" players (very similar to the other value in R', "Very Good", with a similarity degree of 0.75).

Table XIII. Example 5, comparison between our generalized fuzzy division and the Vila et al. division.

TEAM	Generalized Fuzzy Division $C^{\exists}_{\text{TEAM}}$	$C^{\text{most}}_{\text{TEAM}}$	$C^{\text{almost all}}_{\text{TEAM}}$	$C^{\forall}_{\text{TEAM}}$	Vila et al. Division $\Gamma^{\exists}$	Γ^{most}	$\Gamma^{\text{almost all}}$	$\Gamma^{\forall}$
Córdoba	1	1	1	1	1	0.75	0.675	0.5
Granada	1	1	1	1	1	0.75	0.675	0.5
Málaga	1	1	1	1	1	0.75	0.675	0.5
Sevilla	1	0.875	0.95	0.75	1	0.50	0.350	0
Cádiz	1	0.5	0.2	0	1	0.50	0.350	0
Almería	1	0.75	0.7	0.5	1	0.50	0.350	0

However, the Vila et al. division only obtains 0.35 whereas our approach obtains 0.95.

If we apply this method in classic databases, then we can obtain the following pairs of values for $(\Gamma^{\exists}, \Gamma^{\forall})$: $\{(1,1), (0,0), (1,0)\}$. Therefore, for any quantifier Q if we obtain the values $(1,1)$, then $\Gamma^{Q} = 1$. If we obtain the values $(0,0)$ then $\Gamma^{Q} = 0$. Finally, if we obtain the values $(1,0)$ then $\Gamma^{Q} = \alpha_Q$, and this value is not good, as we have seen in Example 5. We have shown that this case does not distinguish among different situations which may occur.

5. THE FSQL SYNTAX FOR FUZZY DIVISION

The FSQL language extends the SQL language to allow fuzzy databases to be managed. At present, we have an FSQL server for flexible queries available for Oracle databases, programmed in PL/SQL. This server allows us to query a fuzzy or classic relational database with the FSQL language. A detailed explanation of the FSQL language and server can be found in Refs. 26 and 27 and mainly in Ref. 2.

In this section, we will give a brief summary of FSQL language and will then suggest a new syntax to express the generalized fuzzy division (Definition 6), the qualified fuzzy intersection (Definition 3), and other interesting queries with fuzzy quantifiers.

5.1. Some Ideas about the FSQL Language

Briefly, FSQL queries are especially flexible since we can use, for example, the following important elements:

(1) **Fuzzy Comparators**: The available fuzzy comparators are shown in Table XIV, including two families (possibility and necessity). The definition of these fuzzy comparators is shown in Ref. 27. For example, FEQ is defined in Equation 3.
(2) **Thresholds**: To retrieve only the *most important* items, for each simple condition, a fulfillment threshold may be established (default is 1), with the format: $\langle$condition$\rangle$ THOLD γ, indicating that the fuzzy condition must be satisfied with a minimum degree $\gamma \in [0,1]$. The reserved word THOLD is optional and it can

Table XIV. The 14 fuzzy comparators for FSQL.

Possibility	Necessity	Significance
FEQ	NFEQ	Possibly/Necessarily fuzzy equal
FGT (FGEQ)	NFGT (NFGEQ)	Possibly/Necessarily fuzzy greater than (or equal to)
FLT (FLEQ)	NFLT (NFLEQ)	Possibly/Necessarily fuzzy less than (or equal to)
MGT (MLT)	NMGT (NMLT)	Possibly/Necessarily much greater (less) than

be substituted by a traditional crisp comparator ($=$, $\leq$, etc.), modifying the meaning of the query. The word THOLD is equivalent to the use of the crisp comparator $\geq$.

(3) **Fuzzy constants**: The right part of a simple condition may be a column or a fuzzy constant. All fuzzy constants types are shown in Table XV. The information about fuzzy attributes (labels definition, margin in approximate values, etc.) are stored in the fuzzy meta-knowledge base (FMB).

(4) **CDEG(⟨attribute⟩) function**: This function shows a column with the fulfillment degree of the condition of the query for a specific attribute. We can use CDEG($*$) to obtain the fulfillment degree of each tuple in the condition (with all of its attributes, not just one of them).

5.2. Fuzzy Division in FSQL

We have extended the FSQL SELECT command to express the generalized fuzzy division with fuzzy quantifiers (Definition 6), the qualified fuzzy intersection (Definition 3), and other interesting queries with fuzzy quantifiers. This new syntax allows us to express any kind of fuzzy division easily and to retrieve the fulfillment degree of the selected items.

Let R and R' be relations with headers (A, B, C) and (B, D), respectively, where A, B, C, and D are simple attributes or sets of attributes denoted by, for example, $A = \{A_1, \ldots, A_n\}$, $B = \{B_1, \ldots, B_m\} \ldots$

According to Definition 2, the fuzzy division

$$R \div^{Q}_{A,B,X} R' \tag{26}$$

Table XV. Fuzzy constants that may be used in FSQL queries.

Fuzzy Constant	Significance
UNKNOWN	Unknown value but the attribute is applicable (type 8 in Table I).
UNDEFINED	The attribute is not applicable or it is meaningless (type 9 in Table I).
NULL	Total ignorance: We know nothing about it (type 10 in Table I).
$[a, b, c, d]	Fuzzy trapezoid ($a \leq b \leq c \leq d$): see Figure 1.
$label	Linguistic label: It may be a trapezoid or a scalar (defined in FMB).
[n, m]	Interval "Between n and m" ($a = b = n$ and $c = d = m$).
#nn	Fuzzy value "approximately n" ($b = c = n$ and $n - a = d - n =$ margin).

has the following general format in FSQL language:

$$
\begin{array}{ll}
\text{SELECT} & A\,[,\text{CDEG}(*)] \\
\text{FROM} & R[,\langle\text{table\_clause}\rangle] \\
\text{WHERE} & [\text{ONCEPERGROUP}] \\
 & [\$Q]\ [[\text{THOLD}]\gamma] \\
 & (\langle\text{subquery}\rangle)
\end{array}
$$

where the items in square brackets are optional items, the items in angled brackets are items to expand, and the meaning of each element is as follows:

- SELECT: This reserved word indicates that it is a query.
- A: It is a list with the R attributes that we are looking for.
- CDEG($*$): This function will show the compatibility or fulfillment degree of the A elements.
- FROM: This clause is used to indicate the relevant relation R and other possible relations in ⟨table_clause⟩. These relations (as in SQL) may be names of tables, views, snapshots, or subqueries. The optional clause ⟨table_clause⟩ is useful to indicate the list of attributes $X \subseteq \{B \cup D\}$ by using a subquery like (SELECT X FROM R') and whatever other relations. The following cases may occur:
  (1) $X = \varnothing$ or this clause does not appear: The command will show the generalized fuzzy division. If CDEG($*$) is presented, then it will show the fulfillment degree of A attributes in this fuzzy division.
  (2) $B \subseteq X$: The command will show the cartesian product according to Equation 1 but if CDEG($*$) is presented, then it will show the fulfillment degree of the qualified fuzzy intersection. The real qualified fuzzy intersection occurs when $X = B$.
  (3) Other X value: The command will show the cartesian product according to Equation 1 and if CDEG($*$) is presented then it will show the fulfillment degree of A attributes in the generalized fuzzy division. It would perhaps be useful to define that if one or more B attributes appear in X then CDEG($*$) will show the values of applying group functions $\mathscr{F}$ (according to Definition 6) group by these B attributes.
  (4) Other cases: If we use other relations, views, subqueries... then the command will show the cartesian product.
- Q: It is a fuzzy quantifier and it is preceded by the symbol $ to distinguish it easily (it is not a reserved word). The fuzzy quantifier Q must be defined in the FMB, except for the quantifier $ALL ($\forall$) and the quantifier $EXISTS ($\exists$). If $\$Q$ does not appear, then the $ALL quantifier is used. The optional value γ indicates the threshold (default is 1) applied to the fulfillment degrees in the resulting relation (column of CDEG function).
- ⟨subquery⟩: It is a subquery with the following format:

$$
\begin{array}{ll}
\text{SELECT} & * \\
\text{FROM} & R' \\
\text{WHERE} & R.B_1\ \langle\text{FCOMP}_1\rangle\ R'.B_1\ [[\text{THOLD}]\ \gamma_1] \\
\text{AND} & \vdots \\
\text{AND} & R.B_m\ \langle\text{FCOMP}_\text{m}\rangle\ \text{R}'.\text{B}_\text{m}\ [[\mathit{THOLD}]\ \gamma_\text{m}]
\end{array}
$$

where

— R': It is the second relevant relation, the divisor, in the fuzzy division. It may be a subquery or the DUAL table. DUAL is a table automatically created by Oracle along with the data dictionary. DUAL is in the schema of the user SYS, but is accessible by the name DUAL to all users. We will use DUAL when the relation R' does not exist, but we want to use a virtual relation using a constants set in the WHERE clause.

— B_i, with $i = 1, \ldots, m$: These values are the set B of R and R' attributes. They are qualified with the name of their relation (R or R') because they would have the same name. If we use DUAL instead of R' then we must use constants instead of R' attributes distinguishing tuples with the OR operator (see Example 6).

— $\langle \text{FCOMP}_i \rangle$, with $i = 1, \ldots, m$: They are the fuzzy comparators (Table XIV) used for each two attributes. Obviously, to retrieve the standard division results we must use the fuzzy comparators FEQ or NFEQ. Fuzzy division selects tuples of the first relation which are related, *in some way*, to Q of tuples in the second relation. That "*way*" is indicated by these fuzzy comparators.

— γ_i, with $i = 1, \ldots, m$: With these values, we can establish a minimum threshold for each B attribute. All of these thresholds must be zero in the standard division. If some threshold γ_i is greater than zero then in the qualified fuzzy intersection (Equation 6), tuples with a fulfillment degree less than γ_i will be removed, and it will not be used in the computation of the final fulfillment degree.

- ONCEPERGROUP option: With classic relations, one single tuple in R connects with zero or one tuple in R', but with fuzzy relations one single tuple in R may connect with zero, one, some, or all of the tuples in R'. This little problem was studied in Ref. 3 and one solution was the following: on performing the qualified fuzzy intersection, every tuple in R is only used *once in each group* (of one A element) according to where it obtains the greatest possibility degree. If there are some items with the same greatest value, then we must maximize all the degrees in that group of A values. If the reserved word ONCEPERGROUP is used, then this solution is applied. It should be noted that when solving this problem we may prevent some possibly useful information from being shown and it substantially increases the number of operations. This option may be especially useful when fuzzy comparators which are different from FEQ or NFEQ are used (see Example 8).

Example 6. According to Example 1, the FSQL query to retrieve the generalized fuzzy division in Table VI with the fuzzy quantifier "most" is:

```
SELECT    TEAM, CDEG(*)
FROM      R
WHERE     $Most THOLD 0
          (SELECT *
           FROM R'
           WHERE R.HEIGHT FEQ R'.HEIGHT THOLD 0
             AND R.QUALITY FEQ R'.QUALITY THOLD 0)
```

If relation R' does not exist, then we can use the following subquery (with R' as an alias) instead of R': (SELECT HEIGHT, QUALITY FROM R WHERE TEAM = 'Córdoba'). Moreover, the same result may be obtained if we want to set the values of R' (Table III) directly:

```
SELECT      TEAM, CDEG(*)
FROM        R
WHERE       $Most THOLD 0
            (SELECT *
             FROM DUAL
             WHERE R.HEIGHT FEQ $Short THOLD 0
                AND R.QUALITY FEQ $Very_ Good THOLD 0
                   OR R.HEIGHT FEQ $Very_ Tall THOLD 0
                AND R.QUALITY FEQ $Bad THOLD 0)
```

It should be noted that we can easily change the fuzzy quantifier.

Example 7. According to Example 1, the FSQL query to retrieve the qualified fuzzy intersection in Table V is:

```
SELECT      TEAM, CDEG(*)
FROM        R, (SELECT HEIGHT, QUALITY FROM R')
WHERE       $Most THOLD 0
            (SELECT *
             FROM R'
             WHERE R.HEIGHT FEQ R'.HEIGHT THOLD 0)
                AND T.QUALITY FEQ R'.QUALITY THOLD 0)
```

This is not a division because the first FROM clause includes a subquery with the same R' attributes which appear in the WHERE clause ($X = B$).

Example 8. According to Example 1, the FSQL query to retrieve those basketball teams in relation R whose players are possibly taller and better (HEIGHT and QUALITY attributes) than *most* of those of R' relation is:

```
SELECT      TEAM, CDEG (*)
FROM        R
WHERE       ONCEPERGROUP
            $Most THOLD 0
            (SELECT *
             FROM R'
             WHERE R.HEIGHT FGT R'.HEIGHT THOLD 0
                AND R.QUALITY FGT R'.QUALITY THOLD 0)
```

It should be noted that the ONCEPERGROUP option means that every tuple in R is only used once in each team according to where it obtains the greatest possibility degree. Thus, we avoid one R player matching more than one of the players in R'.

As well as all the kinds of fuzzy divisions which we have presented above, this syntax enables other useful queries to be expressed as shown in the following examples:

Example 9. "Retrieve teams in which *most* of the players are necessarily tall or very tall":

```
SELECT      TEAM, CDEG(*)
FROM        R R1
WHERE       $Most THOLD 0
            (SELECT *
             FROM R R2
             WHERE R1.TEAM = R2.TEAM
               AND R2.HEIGHT NFGEQ $Tall THOLD 0)
```

It should be noted that this query is not a division.

Example 10. Let us suppose that the relation MANAGERS(Player_Name, Manager_Name) stores all the managers of one player and all the players of one manager (a many–many relationship). Then, the query "Give me the players with *approximately two* managers" is solved with:

```
SELECT      Player_Name, CDEG(*)
FROM        MANAGERS M1
WHERE       $Aprox_2 THOLD 0
            (SELECT *
             FROM MANAGERS M2
             WHERE M1.Player_ Name = M2.Player_ Name)
```

It should be noted that this query is not a division and that we can only use fuzzy absolute quantifiers in these kind of queries.

6. CONCLUSIONS AND FUTURE LINES

The generalized fuzzy division presented is an extension of the division presented in Ref. 3, relaxing the universal quantifier which is inherent in classical relational division. As we have shown here,

- The fuzzy relation can store many types of fuzzy data (possibility distributions, scalars, etc.) and the only necessary thing is to have a comparison function $\Theta^=$ for these types of values. This function may be changed without the need to alter the process of the division.
- We can have any number of common attributes for both relations.
- We can use any fuzzy quantifier (absolute or relative) whatever the shape of its function: decreasing, nondecreasing, or both of them (see Example 1). The analyzed approaches only consider fuzzy quantifiers with a nondecreasing function and this is too restrictive, because some fuzzy quantifiers are inherently decreasing (globally or locally), such as "the minority exclusively", "a few exclusively", "approximately less than half", "approximately x" . . . .

- Results are very close to what would be expected intuitively.
- It is possible to use the generalized fuzzy division with fuzzy or crisp quantifiers in classical databases (without fuzzy attributes).

We have included a comparison with other approaches[13,15,18] which use fuzzy quantifiers and we have seen that the method we have presented is better in all five characteristics.

Moreover, its results can be added to those of other operations such as those presented in Refs. 19, 21, and 22 to obtain more information from a database.

For example, it is possible to know to what extent the following sentence is true and, moreover, to discover which students comply with this sentence and how far each student complies with it:

- *Most students satisfy almost all of the following conditions*: *they are Good in at least one subject and Bad in at least another subject*.

Here we have studied the relational division in fuzzy databases when the A attributes (attributes which are not common to both relations) are crisp, i.e., attributes in which problems do not arise when the projection is applied onto them. As stated in Ref. 3, if among the A attributes there are attributes with fuzzy domains, then it is necessary to establish a prior criterion to discover when two fuzzy values may be considered equal.

It is interesting to study another way of relaxing the division quantifier by using OWA operators[20] with the degrees which are obtained in the qualified fuzzy intersection. The problem is to obtain the OWA operator weights w_i (see Definition 7). Moreover, it may be interesting to consider an extension for generalized fuzzy division when one or both relations have compatibility attributes, i.e., fulfillment degrees associated to the values of some or all of the attributes.

With this and other works, we have achieved the two levels of query languages designed by Codd[28] for relational databases, but they have been extended to fuzzy relational databases: fuzzy relational calculus[29,30] and the fuzzy relational algebra, defined by the GEFRED model[7] and including the fuzzy division[3,25] with fuzzy quantifiers as we have shown in this paper.

Furthermore, in this paper we have extended the FSQL language[2,26,27] to express any kind of fuzzy division (Definition 6), the qualified fuzzy intersection (Definition 3), and other very useful queries using fuzzy quantifiers. It is easy to see that the FSQL syntax put forward is quite powerful and flexible. Moreover, we think that FSQL will be a powerful tool for data mining applications.[2,31,32]

References

1. Petry FE. Fuzzy databases: principles and applications (with chapter contribution by P Bosc). In: Zimmermann H-J, editor. International series in intelligent technologies. Norwall, MA: Kluwer Academic Publishers; 1996.

2. Galindo J. Tratamiento de la imprecisión en bases de datos relacionales: extensión del modelo y adaptación de los SGBD actuales. PhD Thesis, University of Granada, Spain, 1999, www.lcc.uma.es.
3. Galindo J, Medina JM, Aranda MC. Fuzzy division in fuzzy relational databases. An approach. Fuzzy Sets Syst, 2001.
4. Bosc P, Liétard L. On the division of relations with imprecise information. In: de Gooman T, Da Ruan, Kerre EE, editors. Foundations and applications of possibility theory. Proceedings of FART 95, Gent, Belgium. Singapore: World Scientific; 1995. p 287–294.
5. Buckles BP, Petry FE. A fuzzy representation of data for relational databases. Fuzzy Sets Syst 1982;7:213–226.
6. Buckles BP, Petry FE. Extending the fuzzy database with fuzzy numbers. Inform Sci 1984;34:145–155.
7. Medina JM, Pons O, Vila MA. GEFRED. A generalized model of fuzzy relational data bases. Inf Sci 1994;76(1–2):87–109.
8. Mouaddib N. Fuzzy identification in fuzzy databases: the nuanced relational division. Int J Intell Syst 1994;9:461–473.
9. Nakata M. Formulation of division operators in fuzzy relational databases. In: Pons O, Vila MA, Kacprzyk J, editors. Knowledge management in fuzzy databases. Heidelberg: Physica-Verlag; 2000. p 144–156.
10. Prade H, Testemale C. Fuzzy relational databases: representational issues and reduction using similarity measures. J Am Soc Inform Sci 1987;38(2):118–126.
11. Umano M. Freedom-O: a fuzzy database system. In: Gupta-Sanchez, editor. Fuzzy information and decision processes. Amsterdam: North-Holland; 1982. p 339–347.
12. Umano M, Fukami S. Fuzzy relational algebra for possibility-distribution-fuzzy-relational model of fuzzy data. J Intell Inform Syst 1994;3:7–27.
13. Vila MA, Cubero JC, Medina JM, Pons O. The generalized selection: an alternative way for the quotient operations in fuzzy relational databases. In: Bouchon-Meunier B, Yager RR, Zadeh LA, editors. Fuzzy logic and soft computing. Singapore: World Scientific; 1995. p 241–250.
14. Fukami S, Umano M, Muzimoto M, Tanaka H. Fuzzy database retrieval and manipulation language, IEICE Technical Reports, AL-78-85 (Automata and Language) 1979;78(233):65–72.
15. Dubois D, Nakata M, Prade H. Extended divisions for flexible queries in relational databases. In: Pons O, Vila MA, Kacprzyk J, editors. Knowledge management in fuzzy databases. Heidelberg: Physica-Verlag; 2000. p 105–121.
16. Bosc P, Dubois D, Pivert O, Prade H. Flexible queries in relational databases—the example of the division operator. Theor Comput Sci 1997;171:281–302.
17. Dubois D, Prade H. Semantics of quotient operators in fuzzy relational databases. Fuzzy Sets Syst 1996;78:89–93.
18. Yager RR. Fuzzy quotient operators for fuzzy relational databases, First International Fuzzy Engineering Symposium, IFES '91, 1991. p 289–296.
19. Zadeh LA. A computational approach to fuzzy quantifiers in natural languages. Comput Math Appl 1983;9:149–183.
20. Yager RR. On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans Syst Man Cybernet 1988;18:183–90.
21. Yager RR. Fuzzy quotient operators, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU '92, Palma Spain, 1992. p 317–322.
22. Vila MA, Cubero JC, Medina JM, Pons O. Using OWA operator in flexible query processing. In: Yager RR, Kacprzyk J, editors. The ordered weighted averaging operators: theory, methodology and applications. USA: Kluwer; 1997. p 258–274.
23. Delgado M, Sánchez D, Vila MA. A survey of methods for evaluating quantified sentences, EUSFLAT-ESTYLF Joint Conference, Palma de Mallorca, Spain, 1999. p 279–282.

24. Sánchez, D. Adquisición de relaciones entre atributos en bases de datos relacionales. PhD Thesis, University of Granada, Spain, 1999.
25. Galindo J, Medina JM, Aranda MC. Una solución al problema de la división relacional difusa, VII Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'97), Tarragona, Spain, 1997. p 51–56.
26. Galindo J, Medina JM, Pons O, Cubero JC. A server for fuzzy SQL queries. In: Andreasen T, Christiansen H, Larsen HL, editors. Flexible query answering systems, Lecture notes in artificial intelligence (LNAI) 1495. New York: Springer-Verlag; p 164–174.
27. Galindo J, Medina JM, Vila A, Cubero JC. Fuzzy comparators for flexible queries to databases, Iberoamerican Conference on Artificial Intelligence, IBERAMIA '98, Lisbon, Portugal, 1998. p 29–41.
28. Codd EF. Relational completeness of data base sublanguages. In: Rustin R, editor. Data base systems, Courant Computer Science Symposia Series, vol. 6. Englewood Cliffs, NJ: Prentice-Hall; 1972. p 33–64.
29. Galindo J, Medina JM, Aranda, MC. Querying fuzzy relational databases through fuzzy domain calculus. Int J Intell Syst 1999;14(4):375–411.
30. Galindo J, Medina JM, Cubero JC, García MT. Fuzzy quantifiers in fuzzy domain calculus, 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU '2000, Madrid, 2000. p 1697–1704.
31. Carrasco RA, Galindo J, Vila MA, Medina JM. Clustering and fuzzy classification in a financial data mining environment, 3rd International ICSC Symposium on Soft Computing, SOCO '99, Genova, Italy, 1999. p 713–720.
32. Carrasco RA, Vila MA, Galindo J, Cubero JC. FSQL: a tool for obtaining fuzzy dependencies, 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2000, Madrid, Spain, 2000. p 1916–1919.