

FUZZY RELATIONAL DATABASE MODEL AND MANAGEMENT SYSTEM FOR IMPRECISE INFORMATION

NGUYEN HOA^{1,3,*}, NGUYEN THI UYEN NHI², LE NHAT DUY³

¹*Information Technology Faculty, Saigon University, Viet Nam*

²*Faculty of Statistics and Informatics, University of Economics,
The University of Danang, Viet Nam*

³*Faculty of Information Technology, Industrial University of Ho Chi Minh City, Viet Nam*



Abstract. This paper introduces a fuzzy relational database model (FRDB) and the management system for it. FRDB is built by extending the classical relational database model with the fuzzy membership degree of tuples in relations that can represent and query imprecise information in the real world applications. In FRDB, the membership degree of tuples for a fuzzy relation is represented by fuzzy numbers on $[0, 1]$, the fuzzy relational algebraic operations are defined by using the extension principle for computing the minimum and maximum values of such fuzzy numbers. Some properties of the fuzzy relational algebraic operations in FRDB are also formulated and proven. The management system for FRDB with the query language like SQL is built by using a classical open-source management system, implementing the model to allow expressing and executing the soft queries of imprecise information in practice.

Keywords. Fuzzy number; Fuzzy relation; Fuzzy relational database; Fuzzy relational algebraic operation; Fuzzy relational database management system.

1. INTRODUCTION

As we know, the classical relational database model (CRDB) is very useful for modeling, designing and implementing large-scale systems. However, it is restricted for representing and handling uncertain and imprecise information of objects in practice [1]. For example, applications of the classical relational database model cannot deal with the query “find all patients who are *young* and have lung cancer and a *high* treatment cost”, where *young* and *high* are the vague notion and imprecise value [2, 3]. So far, there have been many *fuzzy relational database models* studied and built (e.g. [4-15], [18-26]) based on the fuzzy set theory [2, 3] to overcome the limitations of the classical relational database model in

*Corresponding author.

E-mail addresses: nguyenhoah@sgu.edu.vn (N. Hoa), nhintu@due.edu.vn (N. T. U. Nhi).

representing and handling uncertain and imprecise information of objects in practice.

There are two main approaches to represent fuzzy relations in fuzzy relational database models: (1) representing each fuzzy relation as a set of tuples whose each attribute may take a fuzzy set or a possibility distribution is inferred from a fuzzy set (e.g. [4-5] or [6-11], respectively), whereby the membership degree of tuples for the relation is hidden in that of their attribute values; (2) representing each fuzzy relation as a fuzzy set of tuples whose each attribute only takes a single and precise value (e.g. [12-27]), whereby the membership degree of tuples for the relation also is the membership degree of elements for the fuzzy set expressing that fuzzy relation.

Fuzzy relational database models that are built based on one of above approaches are extensions of classical relational database model with fuzzy sets. They have different capabilities for expressing and dealing with uncertain and imprecise information. Moreover, no model would be so universal that could include all measures and tackle all aspects of uncertainty and imprecision.

For the first approach, as the models in [4] and [5], the attribute value of relations was represented by a fuzzy set or a set of fuzzy sets, the fuzzy relational algebraic operations were defined by using similarity relations on the value domain of the attributes and proximity binary relations on fuzzy sets. Meanwhile, regarding the proposes in [6-9], the attribute value of relations was represented by a possibility distribution, the fuzzy relational algebraic operations were defined by using the possibility theory and proximity binary relations on possibility distributions. Also, but the works in [10] and [11] only focused on the proof for the set of the operations $\{\cup, \times, -, \sigma, \Pi\}$ to be a complete set of all fuzzy relational algebraic operations and the equivalence and transformation on them, respectively. The models based on the first approach are capable of expressing well imprecise information due to the attribute values of relations are explicitly represented by fuzzy sets or possibility distributions but are limited the performance in handling such information because of the high complexity in manipulating and computing similarity relations, proximity binary relations on fuzzy sets and possibility distributions.

For the second approach, as the models in [12-27], the attribute of relations only takes a single and precise value, each relation was represented as a fuzzy set of tuples (i.e. a fuzzy extension of CRDB relations), the fuzzy relational algebraic operations were defined by extending directly classical relational algebraic operations based on fuzzy set operations. The models based on the second approach are capable of querying and manipulating data more effective than the models based on the first approach due to the low complexity in computing and handling classical relations well as well fuzzy set operations but the capability of expressing imprecise information is not as good as those models, because the imprecision of the information is not explicitly represented by attribute values of relations but only inferred from the membership degree of tuples for the relation including those attributes. There was a large number of models as in [12-21] and [23-24] that the membership degree of tuples is a number in $[0, 1]$. These models were restricted in representing the associated imprecise degree of attribute values. In the real world relational databases, since attribute values of tuples may be imprecise, there are many situations in which we do not know exactly the membership degree of tuples as a number in $[0, 1]$ but only can estimate it as an approximate number (or a fuzzy number) on $[0, 1]$. Some fuzzy relational database models as in [22] and [25-27] have been introduced to overcome the shortcoming of above mentioned fuzzy

relational database models. However, in [26], only the notions of the relational schema and instance were defined meanwhile the relational algebraic operations were not introduced. In [27], the data representative notions were not defined really formally and some fuzzy relational algebraic operations were missing. In [25], the set of basic fuzzy relational algebraic operations was not complete. In [22], a full set of basic fuzzy relational algebraic operations was proposed but no management system was built for the model. Also, there has not been any management system for the models in [25-27]. Thus, the abilities of expressing and dealing with imprecise information of those models were limited in the real world applications.

In this paper, we propose a fuzzy relational database model (FRDB) as an extension of CRDB based on the second approach to overcome the shortcomings of the models in ([22], [25-27]) and develop a management system for it with the query language like SQL to represent and query softly imprecise information in practice. In our new FRDB, the notions of the data representative model are completely defined formally, the membership degree of each tuple in a relation is expressed by a fuzzy number on the interval $[0, 1]$, the full set of basic fuzzy relational algebraic operations is built based on the extension principle for computing the minimum and maximum values of fuzzy numbers. Some properties of these algebraic operations are also formulated and proven. The new management system for FRDB built by using SQLite [28] as an underlying database management system (DBMS) inherits completely components of a general DBMS and has new features such as creating FRDB schemas and fuzzy relations, editing soft queries of imprecise information and executing them.

Fuzzy sets as the basis of mathematics to develop FRDB is presented in Section 2. Schemas and relations of FRDB are introduced in Section 3. Section 4 presents fuzzy relational algebraic operations and their properties. The management system for FRDB is presented in Section 5. Finally, Section 6 concludes the paper and outlines further research directions in the future.

2. FUZZY SETS

For a classical set, an element is to be or not to be in the set or, in other words, the membership degree of an element in the set is binary. For a fuzzy set, the membership degree of an element in the set is expressed by a real number in the interval $[0, 1]$. The fuzzy set is extended from the classical set as in [2] and is defined as follows.

Definition 1. A *fuzzy set* A on a domain X is defined by a membership function μ_A from X to the closed interval $[0, 1]$. For each $x \in X$, $\mu_A(x)$ is the membership degree of x for A .

We note that a classical set A on X also is a fuzzy set [3] with the membership function $\mu_A(x) = 1, \forall x \in A$ and $\mu_A(x) = 0, \forall x \notin A$ or even a element e in X is also considered as a special fuzzy set on X with the membership function $\mu_e(e) = 1$ and $\mu_e(x) = 0, \forall x \in X$ and $x \neq e$. The fuzzy set A on X can be denoted by $A = \{x : \mu_A(x) \mid x \in X\}$. In addition, the notation $A(x)$ can be used to replace $\mu_A(x)$.

The *support* of a fuzzy set A on X is the classical set that contains all the elements of X that have nonzero membership degrees in A . The *height* $h(A)$ of a fuzzy set A on X is the largest membership degree obtained by any element in that set. It means $h(A) = \sup_{x \in X} \mu_A(x)$. A fuzzy set A is called *normal* if $h(A) = 1$ and it is called *subnormal* if $h(A) < 1$. A fuzzy set A on the real number set \mathbb{R} is called *convex* if for any elements x, y, z

in the support of A , the relation $x < y < z$ implies that $\mu_A(y) \geq \min(\mu_A(x), \mu_A(z))$.

The fuzzy numbers are special fuzzy sets that are used to represent the fuzzy relations in FRDB. The fuzzy numbers are defined in [3] as follows.

Definition 2. A *fuzzy number* A is a fuzzy set on the real number set \mathbb{R} such that:

1. A is a normal and convex fuzzy set.
2. The support of A is bounded.

Example 1. The fuzzy set *approx_0.6* that is given by the membership function and its graph as Figure 1 below being a fuzzy number.

$$\text{approx_0.6}(x) = \begin{cases} (x - 0.4)/0.2 & \forall x \in [0.4, 0.6], \\ (0.8 - x)/0.2 & \forall x \in (0.6, 0.8], \\ 0 & \forall x \notin [0.4, 0.8]. \end{cases}$$

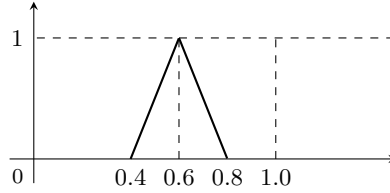


Figure 1. Fuzzy number *approx_0.6*

For computing and combining the membership degrees of tuples in the fuzzy relational algebraic operations, we use the operations MIN and MAX to determine the minimum and maximum values of fuzzy numbers. The operations MIN and MAX are defined by the extension principle in [3] as below.

Definition 3. Let A and B be two fuzzy numbers. The *minimum* and *maximum values* of A and B are fuzzy numbers that are defined by

1. $\text{MIN}(A, B)(z) = \sup_{z=\min(x,y)} \min[A(x), B(y)]$,
 2. $\text{MAX}(A, B)(z) = \sup_{z=\max(x,y)} \min[A(x), B(y)]$,
- $\forall x, y, z \in \mathbb{R}$.

As noted in Definition 1, here $A(x)$ and $B(y)$ are identified with the membership functions of x and y for the fuzzy sets A and B , respectively (i.e. $A(x) = \mu_A(x)$, $B(y) = \mu_B(y)$).

Example 2. Let

$A = \{x : A(x) \mid x \in \mathbb{R}\} = \{1 : 1, 0.9 : 0.8, 0.8 : 0.3\}$
and $B = \{y : B(y) \mid y \in \mathbb{R}\} = \{0.6 : 0.3, 0.5 : 1, 0.4 : 0.4\}$ be two fuzzy numbers on $[0, 1] \subseteq \mathbb{R}$, then $\text{MIN}(A, B) = \{0.6 : 0.3, 0.5 : 1, 0.4 : 0.4\}$.

3. PROPOSED FRDB MODEL

As for CRDB, the schema, relation are the fundamental concepts in the FRDB model.

3.1. FRDB schemas

The FRDB schema is extended from those of CRDB with the fuzzy membership function on the domains of the relational attributes and is defined as follows.

Definition 4. A *fuzzy relational schema* is a pair $R = (U, \mu)$, where

1. $U = \{A_1, A_2, \dots, A_k\}$ is a set of pairwise different attributes.
2. μ is a mapping from $D_1 \times D_2 \times \dots \times D_k$ to $\mathfrak{I}([0, 1])$, where D_i is the domain of the attribute A_i ($i = 1, \dots, k$) and $\mathfrak{I}([0, 1])$ is the set of all fuzzy numbers on $[0, 1]$.

As in CRDB, the notations $R(U, \mu)$ and R can be used to replace $R = (U, \mu)$. In addition, each $t = (v_1, v_2, \dots, v_k) \in D_1 \times D_2 \times \dots \times D_k$ is called a tuple on the set of attributes $\{A_1, A_2, \dots, A_k\}$, where $v_i \in D_i$ is a single value that the attribute A_i may take, $\mu(t)$ is a fuzzy number on $[0, 1]$ that represents the imprecise degree that the attributes A_1, A_2, \dots, A_k can take the values v_1, v_2, \dots, v_k , respectively.

Note that in the fuzzy relational database models [3, 14, 21], $\mu(v_1, v_2, \dots, v_k) \in [0, 1]$, in CRDB we can consider as $\mu(v_1, v_2, \dots, v_k)$ taking 0 or 1. Thus, the notion of the *fuzzy relational schema* in Definition 4 is an extension of relational schemas in CRDB and the fuzzy relational database models [3, 14, 21], respectively.

Example 3. A fuzzy relational schema **PATIENT** in FRDB describing patients can be as follows

PATIENT(P_NAME, P_AGE, P_DISEASE, D_COST, μ),

where μ is the mapping from **string** \times **integer** \times **string** \times **real** to $\mathfrak{I}([0, 1])$, **string**, **integer** and **real** are the domains of the attributes P_NAME, P_DISEASE, P_AGE and D_COST, respectively.

3.2. FRDB relations

A FRDB relation is an instance of a FRDB schema, where each relational tuple takes a fuzzy number on $[0, 1]$ representing the aggregated imprecise degree of the relational attribute values of the tuple, as the definition below.

Definition 5. Let $U = \{A_1, A_2, \dots, A_k\}$ be a set of k pairwise different attributes. A *fuzzy relation* r over the fuzzy relational schema $R(U, \mu)$ is a finite set of tuples $\{t_1, t_2, \dots, t_n\}$ on the set $\{A_1, A_2, \dots, A_k\}$ such that each tuple t_i is associated with the fuzzy number $\mu(t_i)$ representing the membership degree of t_i in r , for every $i = 1, 2, \dots, k$. The notation $t.A$ or $t[A]$ is used to denote the value of the attribute A of the tuple t in r . The membership degree of t_i in r is denoted by $\mu_r(t_i)$.

Note that in the fuzzy relational database models [3, 14, 21], $\mu_r(t_i) \in [0, 1]$ and in CRDB, we can think that $\mu_r(t_i)$ takes 0 or 1. Thus, the relation in CRDB and the models [3, 14, 21] is a special case of the fuzzy relation in Definition 5. As in CRDB, if we only care about an unique relation over a schema then we can unify its symbol name with its schema's name.

For each set of attributes $X \subseteq \{A_1, A_2, \dots, A_k\}$, the notation $t[X]$ is used to denote the rest of t after eliminating the value of attributes not belonging to X .

Example 4. A fuzzy relation, named **PATIENT**, over the schema **PATIENT**(P_NAME, P_AGE, P_DISEASE, D_COST, μ) can be given as Table 1.

In the relation **PATIENT**, the attributes P_NAME, P_AGE, P_DISEASE, and D_COST describe the information about the name, age, disease and daily treatment cost of each patient, respectively. In reality, while diagnosing, the disease of each patient is not always determined certainly by the physicians. Similarly, the daily treatment cost for patients is also

Table 1. Relation PATIENT

P_NAME	P_AGE	P_DISEASE	D_COST	μ
John	53	lung cancer	180	0.9
Paul	65	cirrhosis	9	<i>approx_0.6</i>
Mary	21	hepatitis	10	<i>high</i>
Anna	50	bronchitis	6	1.0

not known definitely even as the patients know about their diseases. Here, the conventional unit for the daily treatment cost is 1 (USD).

We note that $\mu(t)$ represents the membership degree of each tuple t in the relation (by Definition 5), it means the precise degree of the information of the attribute values that is expressed by t . For example, consider the tuple t_1 (the first tuple) in the relation PATIENT, assuming that the information of the patient's name (John) expressed by t_1 is precise, then $\mu(t_1) = 0.9$ of t_1 represents the aggregated precise degree of the information about the age (53), diagnosed disease (Lung cancer) and daily treatment cost (180 USD) of the patient. With t_3 , we do not know precisely both the information of the attribute values that it represents and its membership degree in the relation PATIENT. We are able to just estimate $\mu(t_3)$ is high where $high = \{0.5 : 0, 0.6 : 0.5, 0.7 : 0.8, 0.8 : 0.9, 0.9 : 1.0, 1 : 1.0\}$ is a fuzzy number on $[0, 1]$. Meanwhile, the information of the patient Anna in t_4 is precisely assumed, so $\mu(t_4) = 1.0$.

In the real world applications, fuzzy sets that represent the membership degrees of tuples in a fuzzy relation, such as *high* and *approx_0.6* above, will be defined compatibly and consistently based on the meaning and precise degree of the information that these tuples express. The fuzzy sets *high* and *approx_0.6* that are given as in this example only simply illustrates for Definition 5.

Now, the notion of a fuzzy relational database is defined as follows.

Definition 6. A *fuzzy relational database* over a set of attributes is a set of fuzzy relations corresponding with the set of their fuzzy relational schemas.

4. FRDB ALGEBRAIC OPERATIONS

The basic fuzzy relational algebraic operations on FRDB are extended those operations of CRDB taking into account the fuzzy membership degree of tuples in relations.

4.1. Selection

Before defining the selection operation for FRDB, we present the formal syntax and semantics of selection conditions by extending those definitions of CRDB with fuzzy sets and fuzzy binary relations as the following definitions.

Definition 7. Let R be a FRDB schema and X be a set of relational tuple variables and θ be a binary relation from $\{=, \neq, \leq, <, >, \geq\}$. Then *selection conditions* are inductively defined and have one of the following forms:

1. $x.A\theta v$, where $x \in X$, A is an attribute in R and v is a precise value.
2. $x.A \rightarrow v$, where $x \in X$, A is an attribute in R , \rightarrow is a binary fuzzy relation and v is a fuzzy set value.
3. $x.A_1\theta x.A_2$, where $x \in X$, A_1 and A_2 are two different attributes in R .
4. $\neg E$ if E is a selection condition.

5. $E_1 \wedge E_2$ if E_1 and E_2 are selection conditions on the same relational tuple variable.
6. $E_1 \vee E_2$ if E_1 and E_2 are selection conditions on the same relational tuple variable.

Definition 8. Let $R(\mathbf{U}, \mu)$ be a FRDB schema, r be a relation over R , x be a tuple variable and t be a tuple in r . The interpretation of selection conditions with respect to R, r and t , denoted by $Int_{R,r,t}$, is the partial mapping from the set of all selection conditions to the set of all fuzzy numbers on $[0, 1]$ that is inductively defined as follows:

1. $Int_{R,r,t}(x.A\theta v) = \mu_r(t)$ if $t.A\theta v$ and $Int_{R,r,t}(x.A\theta v) = 0$ otherwise.
2. $Int_{R,r,t}(x.A \rightarrow v) = \text{MIN}(\mu_r(t), \mu_\varphi(t))$, with $\varphi = x.A \rightarrow v$.
3. $Int_{R,r,t}(x.A_1\theta x.A_2) = \mu_r(t)$ if $t.A_1\theta t.A_2$ and $Int_{R,r,t}(x.A_1\theta x.A_2) = 0$ otherwise.
4. $Int_{R,r,t}(\neg E) = 1 - Int_{R,r,t}(E)$.
5. $Int_{R,r,t}(E_1 \wedge E_2) = \text{MIN}(Int_{R,r,t}(E_1), Int_{R,r,t}(E_2))$.
6. $Int_{R,r,t}(E_1 \vee E_2) = \text{MAX}(Int_{R,r,t}(E_1), Int_{R,r,t}(E_2))$.

Note that $\varphi = x.A \rightarrow v$ is the fuzzy set whose elements are tuples in r . For each $t \in r$, $\mu_\varphi(t) = v(t.A)$. Intuitively, $Int_{R,r,t}(x.A\theta v)$ and $Int_{R,r,t}(x.A \rightarrow v)$ are respectively the satisfied degrees of the conditions $t.A\theta v$ and $t.A \rightarrow v$ for the tuple t in r while $Int_{R,r,t}(x.A_1\theta x.A_2)$ is the satisfied degree of the condition $t.A_1\theta t.A_2$ for the tuple t in r .

Example 5. Let the fuzzy set *young* represent the young age of a patient whose membership function and graph as Figure 2 below

$$young(x) = \begin{cases} 1, & \forall x \in [0, 20] \\ (35 - x)/15, & \forall x \in (20, 35) \\ 0, & \forall x \geq 35 \end{cases}$$

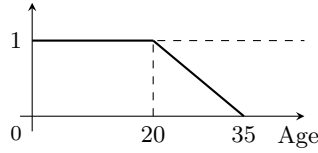


Figure 2. Graph of the fuzzy set young

and $E = x.P_AGE \rightarrow young \wedge x.P_DISEASE = hepatitis$ be a selection condition on the schema $R = \mathbf{PATIENT}$ in Example 4, then

$$\begin{aligned} Int_{R,r,t_3}(E) &= \text{MIN}(Int_{R,r,t_3}(x.P_AGE \rightarrow young), Int_{R,r,t_3}(x.P_DISEASE = hepatitis)) \\ &= \text{MIN}(\text{MIN}(\mu_r(t_3), young(21)), \mu_r(t_3)) \\ &= \text{MIN}(\text{MIN}(high, 0.93), high) \\ &= \{0.5 : 0, 0.6 : 0.5, 0.7 : 0.8, 0.8 : 0.9, 0.9 : 1.0, 0.93 : 1.0\}. \end{aligned}$$

Let $approx_0.92 = \{0.5 : 0, 0.6 : 0.5, 0.7 : 0.8, 0.8 : 0.9, 0.9 : 1.0, 0.93 : 1.0\}$, we have $InfLt_{R,r,t_3}(E) = approx_0.92$.

Definition 9. Let $R(\mathbf{U}, \mu)$ be a FRDB schema, r be a fuzzy relation over R and ϕ be a selection condition. The *selection* on r with respect to ϕ , denoted by $\sigma_\phi(r)$, is the fuzzy relation r^* over R , including all tuples defined by $r^* = \{t \in r \mid \mu_{r^*}(t) = Int_{R,r,t}(\phi) \neq 0\}$.

Example 6. Consider the relation $r = \mathbf{PATIENT}$ in Example 4, the query “Find all patients who are young and have hepatitis” can be done by the selection operation

$$r^* = \sigma_\phi(\text{PATIENT}),$$

where,

$$\phi = x.\text{P_AGE} \rightarrow \text{young} \wedge x.\text{P_DISEASE} = \text{hepatitis}.$$

The selection is implemented by checking the satisfaction of all tuples in PATIENT for the selection condition ϕ . From the result computed in Example 5 above, we can see that only the tuple $t_3 = (\text{Mary}, 21, \text{hepatitis}, 10)$ satisfies ϕ with the value of membership function $\mu(t_3)$ being $\text{approx_}0.92 = \{0.5 : 0, 0.6 : 0.5, 0.7 : 0.8, 0.8 : 0.9, 0.9 : 1.0, 0.93 : 1.0\}$. So, $\sigma_\phi(\text{PATIENT}) = \{t = (\text{Mary}, 21, \text{hepatitis}, 10) \mid \mu(t) = \text{approx_}0.92\}$.

4.2. Projection

A projection of a FRDB relation on a set of attributes is a new FRDB relation where only the attributes in that set are considered for each tuple of the new relation. The projection operation of a FRDB relation is extended from the projection operation of a CRDB relation with the fuzzy membership degree of tuples and is defined as follows.

Definition 10. Let $R = (\mathbf{U}, \mu)$ be a FRDB schema, r be a relation over R and $\mathbf{L} = \{A_1, A_2, \dots, A_k\}$ be a subset of \mathbf{U} . The projection of r on \mathbf{L} , denoted by $\Pi_L(r)$, is the fuzzy relation r^* over the schema R^* , determined by:

1. $R^* = (\mathbf{L}, \mu^*)$, where μ^* is the mapping from $D_1 \times D_2 \times \dots \times D_k$ to $\mathfrak{S}([0, 1])$, D_i is the value domain of $A_i (i = 1, \dots, k)$.
2. $r^* = \{t^* = t[\mathbf{L}] \mid t \in r \text{ and } \mu_{r^*}^*(t^*) = \text{MAX}_{t \in r} \{\mu_r(t) \mid t^* = t[\mathbf{L}]\}\}$.

Example 7. The projection of the relation PATIENT in Table 1 on $\mathbf{L} = \{\text{P_NAME}, \text{P_DISEASE}\}$ is the relation $\Pi_L(\text{PATIENT})$ as in Table 2.

Table 2. Relation $\Pi_L(\text{PATIENT})$

P_NAME	P_DISEASE	μ^*
John	lung cancer	0.9
Paul	cirrhosis	<i>approx_0.6</i>
Mary	hepatitis	<i>high</i>
Anna	bronchitis	1.0

4.3. Cartesian product

For the Cartesian product of two FRDB relations, as in CRDB, we assume the set of attributes of their schemas are disjoint and every k -tuple $t = (v_1, v_2, \dots, v_k)$ is an un-ordered list. The Cartesian product of two FRDB relations is extended with the fuzzy membership degree of tuples from the Cartesian product of two CRDB relations as follows.

Definition 11. Let $\mathbf{U}_1, \mathbf{U}_2$ be two sets of attributes that have not any common element, $R_1 = (\mathbf{U}_1, \mu_1), R_2 = (\mathbf{U}_2, \mu_2)$ be two FRDB schemas, r_1, r_2 be two relations over R_1 and R_2 , respectively. The *Cartesian product* of r_1 and r_2 , denoted by $r_1 \times r_2$, is the fuzzy relation r over R , determined by:

1. $R = (\mathbf{U}, \mu)$, where $\mathbf{U} = \mathbf{U}_1 \cup \mathbf{U}_2, \mu$ is the mapping from $D_1 \times D_2 \times \dots \times D_{k+m}$ to $\mathfrak{S}([0, 1])$, $k = |\mathbf{U}_1|, m = |\mathbf{U}_2|, D_i$ is the value domain of $A_i \in \mathbf{U}_1 \cup \mathbf{U}_2$.
2. $r = \{t = (v_1, v_2, \dots, v_k, v_{k+1}, v_{k+2}, \dots, v_{k+m}) \mid t_1 = (v_1, v_2, \dots, v_k), t_2 = (v_{k+1}, v_{k+2}, \dots, v_{k+m}), t_1 \in r_1, t_2 \in r_2, \text{ and } \mu_r(t) = \text{MIN}(\mu_{1r_1}(t_1), \mu_{2r_2}(t_2))\}$.

4.4. Join

The join of two FRDB relations is extended from the natural join of two CRDB relations with the fuzzy membership degree of tuples as following definition.

Definition 12. Let \mathbf{U}_1 and \mathbf{U}_2 be two sets of attributes such that if they have the same name attributes, respectively in those two sets then such attributes have the same value domain. Let $R_1 = (\mathbf{U}_1, \mu_1)$ and $R_2 = (\mathbf{U}_2, \mu_2)$ be two FRDB schemas, r_1, r_2 be two relations over R_1 and R_2 , respectively and $\{A_k, \dots, A_l\} = \mathbf{U}_1 \cap \mathbf{U}_2$. The *natural join* of r_1 and r_2 , denoted by $r_1 \bowtie r_2$, is the fuzzy relation r over the schema R , determined by:

1. $R = (\mathbf{U}, \mu)$, where $\mathbf{U} = \mathbf{U}_1 \cup \mathbf{U}_2$, μ is the mapping from $D_1 \times D_2 \times \dots \times D_n$ to $\mathfrak{J}([0, 1])$, $n = |\mathbf{U}|$, D_i is the value domain of $A_i \in \mathbf{U}_1 \cup \mathbf{U}_2$.
2. $r = \{t = (v_1, \dots, v_i, v_k, \dots, v_l, v_m, \dots, v_n) \mid t_1 = (v_1, \dots, v_j, v_k, \dots, v_l), t_2 = (v_k, \dots, v_l, v_m, \dots, v_n), t_1 \in r_1, t_2 \in r_2 \text{ such that } v_k = t_1[A_k] = t_2[A_k], \dots, v_l = t_1[A_l] = t_2[A_l] \text{ and } \mu_r(t) = \text{MIN}(\mu_{1r_1}(t_1), \mu_{2r_2}(t_2))\}$.

Example 8. Let $\mathbf{U}_1 = \{\text{P_ID}, \text{P_DISEASE}\}$, $\mathbf{U}_2 = \{\text{P_NAME}, \text{P_DISEASE}\}$ be two sets of attributes and $\text{PATIENT}_1, \text{PATIENT}_2$ be two fuzzy relations over two schemas $R_1 = (\mathbf{U}_1, \mu_1)$ and $R_2 = (\mathbf{U}_2, \mu_2)$ respectively as Tables 3 and 4. It is easy to see that

$$\text{MIN}(\{1 : 1, 0.9 : 0.8, 0.8 : 0.3\}, \{0.6 : 0.3, 0.5 : 1, 0.4 : 0.4\}) = \{0.6 : 0.3, 0.5 : 1, 0.4 : 0.4\}.$$

So, the result of the join of PATIENT_1 and PATIENT_2 is the relation $\text{PATIENT}_1 \bowtie \text{PATIENT}_2$ over the schema $R = (\mathbf{U}_1 \cup \mathbf{U}_2, \mu)$ computed as in Table 5.

Table 3. Relation PATIENT_1

P_ID	P_DISEASE	μ_1
PT005	bronchitis	1.0
PT006	gall-stone	$\{1 : 1, 0.9 : 0.8, 0.8 : 0.3\}$

Table 4. Relation PATIENT_2

P_NAME	P_DISEASE	μ_2
Edgar	bronchitis	0.9
Diana	gall-stone	$\{0.6 : 0.3, 0.5 : 1, 0.4 : 0.4\}$

Table 5. Relation $\text{PATIENT}_1 \bowtie \text{PATIENT}_2$

P_ID	P_NAME	P_DISEASE	μ
PT005	Edgar	bronchitis	0.9
PT006	Diana	gall-stone	$\{0.6 : 0.3, 0.5 : 1, 0.4 : 0.4\}$

4.5. Intersection, union, and difference

By extending the operations of fuzzy sets in [2, 3], the set operations on the fuzzy relations in FRDB are defined in turn as below.

Definition 13. Let $R = (\mathbf{U}, \mu)$ be a FRDB schema, r_1 and r_2 be two relations over R . The *intersection* of r_1 and r_2 , denoted by $r_1 \cap r_2$, is the fuzzy relation r over R including tuples t 's on \mathbf{U} defined by

$$r_1 \cap r_2 = \{t \mid \mu_{r_1 \cap r_2}(t) = \text{MIN}(\mu_{r_1}(t), \mu_{r_2}(t))\}$$

Example 9. Consider two fuzzy relations DIAGNOSE₁ and DIAGNOSE₂ over the schema DIAGNOSE (\mathbf{U}, μ) as in Tables 6 and 7, where $\mathbf{U} = \{P_ID, D_ID, P_AGE, P_DISEASE, DATE\}$. Then the intersection of DIAGNOSE₁ and DIAGNOSE₂ is the relation DIAGNOSE₁ ∩ DIAGNOSE₂ computed as in Table 8.

Table 6. Relation DIAGNOSE₁

P_ID	D_ID	P_AGE	P_DISEASE	DATE	μ
PT215	DT093	60	tuberculosis	15/2/2020	1.0
PT234	DT102	41	hepatitis	18/2/2020	high

Table 7. Relation DIAGNOSE₂

P_ID	D_ID	P_AGE	P_DISEASE	DATE	μ
PT383	DT102	68	lung cancer	18/2/2020	0.9
PT234	DT102	41	hepatitis	18/2/2020	1.0
PT242	DT025	17	cholecystitis	15/2/2020	approx_0.6

Table 8. Relation DIAGNOSE₁ ∩ DIAGNOSE₂

P_ID	D_ID	P_AGE	P_DISEASE	DATE	μ
PT234	DT102	41	hepatitis	18/2/2020	high

Definition 14. Let $R = (\mathbf{U}, \mu)$ be a FRDB schema, r_1 and r_2 be two relations over R . The union of r_1 and r_2 , denoted by $r_1 \cup r_2$, is the fuzzy relation r over R including tuples t 's on \mathbf{U} defined by

$$r_1 \cup r_2 = \{t \mid \mu_{r_1 \cup r_2}(t) = \text{MAX}(\mu_{r_1}(t), \mu_{r_2}(t))\}.$$

Definition 15. Let $R = (\mathbf{U}, \mu)$ be a FRDB schema, r_1 and r_2 be two relations over R . The difference of r_1 and r_2 , denoted by $r_1 - r_2$, is the fuzzy relation r over R including tuples t 's on \mathbf{U} defined by

$$r_1 - r_2 = \{t \mid \mu_{r_1 - r_2}(t) = \text{MIN}(\mu_{r_1}(t), 1 - \mu_{r_2}(t))\}.$$

4.6. Property of algebraic operations

The properties of the fuzzy relational algebraic operations in FRDB are extended from those in CRDB as equations below.

Proposition 1. Let r, r_1, r_2 and r_3 be FRDB relations, ϕ_1 and ϕ_2 be two selection conditions on r , \mathbf{A} and \mathbf{B} be two subsets of attributes of r and $\mathbf{A} \subseteq \mathbf{B}$. Then

$$\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_2}(\sigma_{\phi_1}(r)) \quad (1)$$

$$\Pi_{\mathbf{A}}(\Pi_{\mathbf{B}}(r)) = \Pi_{\mathbf{A}}(r) \quad (2)$$

$$r_1 \times r_2 = r_2 \times r_1 \quad (3)$$

$$(r_1 \times r_2) \times r_3 = r_1 \times (r_2 \times r_3) \quad (4)$$

$$r_1 \bowtie r_2 = r_2 \bowtie r_1 \quad (5)$$

$$(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3) \quad (6)$$

$$r_1 \cap r_2 = r_2 \cap r_1 \quad (7)$$

$$(r_1 \cap r_2) \cap r_3 = r_1 \cap (r_2 \cap r_3) \quad (8)$$

$$r_1 \cup r_2 = r_2 \cup r_1 \quad (9)$$

$$(r_1 \cup r_2) \cup r_3 = r_1 \cup (r_2 \cup r_3) \quad (10)$$

Note that, in (3) and (4), r_1 , r_2 and r_3 do not have any common attribute (Definition 11), in (5) and (6), if r_1 , r_2 and r_3 have the common attributes then such attributes must have the same domain (Definition 12).

Proof. For the equation (1), let $s = \sigma_{\phi_2}(r)$, we have

$$\begin{aligned}
\sigma_{\phi_1}(\sigma_{\phi_2}(r)) &= \{t \in s \mid \text{Int}_{R,s,t}(\phi_1) \neq 0\} && \text{(Definition 9)} \\
&= \{t \in r \mid \text{Int}_{R,r,t}(\phi_2) \neq 0 \wedge \text{Int}_{R,s,t}(\phi_1) \neq 0\} \\
&= \{t \in r \mid \text{Int}_{R,r,t}(\phi_2) \neq 0 \wedge \text{Int}_{R,r,t}(\phi_1) \neq 0\} && \text{(Because } s \subseteq r) \\
&= \{t \in r \mid \text{MIN}(\text{Int}_{R,r,t}(\phi_2), \text{Int}_{R,r,t}(\phi_1)) \neq 0\} && \text{(Definition 8)} \\
&= \{t \in r \mid \text{Int}_{R,r,t}(\phi_2 \wedge \phi_1) \neq 0\} = \sigma_{\phi_1 \wedge \phi_2}(r).
\end{aligned}$$

So, $\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_1 \wedge \phi_2}(r)$, the equation $\sigma_{\phi_2}(\sigma_{\phi_1}(r)) = \sigma_{\phi_2 \wedge \phi_1}(r)$ is proven similarly. Since $\phi_1 \wedge \phi_2 \Leftrightarrow \phi_2 \wedge \phi_1$ (the logical conjunction of selection conditions is commutative), hence $\sigma_{\phi_1 \wedge \phi_2}(r) = \sigma_{\phi_2 \wedge \phi_1}(r)$. Therefore, it results in $\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_2}(\sigma_{\phi_1}(r))$. Thus, the equation (1) is proven.

For the equation (2), because $\mathbf{A} \subseteq \mathbf{B}$, so $\mathbf{A} \cap \mathbf{B} = \mathbf{A}$. From Definition 10, it is easy to see that the sides of (2) are the relations over the same schema with the set of attributes $\mathbf{A} \cap \mathbf{B} = \mathbf{A}$. By the property of the projection of the classical relations, Definition 5 and Definition 10, it follows that two classical sets of tuples which are collected respectively from two relations $\Pi_A(\Pi_B(r))$ and $\Pi_A(r)$ are the same. Moreover, by Definition 10, the operation MAX in two sides of (2) is executed on the same value set of the membership degrees of tuples of r . From that $\Pi_A(\Pi_B(r)) = \Pi_{A \cap B}(r) = \Pi_A(r)$. Thus, the equation (2) is proven.

For the equation (5), clearly, $r_1 \bowtie r_2$ and $r_2 \bowtie r_1$ are two relations over the same schema. By the property of the join of the classical relations, Definition 5 and Definition 12, it follows that two classical sets of tuples which are collected respectively from two relations $r_1 \bowtie r_2$ and $r_2 \bowtie r_1$ are the same. In addition, the operation MIN of two fuzzy numbers (two membership degrees of two tuples in r_1 and r_2 , respectively) has commutativity. From that leading the join of tuples has commutativity. So, by Definition 12 we have $r_1 \bowtie r_2 = r_2 \bowtie r_1$.

For the equation (6), By Definition 12, clearly $(r_1 \bowtie r_2) \bowtie r_3$ and $r_1 \bowtie (r_2 \bowtie r_3)$ are two relations over the same schema. By the property of the join of the classical relations, Definition 5 and Definition 12, it follows that two classical sets of tuples which are collected respectively from two relations $(r_1 \bowtie r_2) \bowtie r_3$ and $r_1 \bowtie (r_2 \bowtie r_3)$ are the same. Let A be a common attribute in $\mathbf{U}_1, \mathbf{U}_2$ and \mathbf{U}_3 of R_1, R_2 và R_3 , because the operation MIN of two fuzzy numbers and the identical operation of attribute values have associativity, from that the join of tuples has associativity. Thus, by Definition 12, it results in $(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$.

Because the Cartesian product (Definition 11) is a particular case of the join, we have the equations (3) and (4) as the straight corollaries of the equations (5) and (6), respectively.

For the equations (7), (8), (9) and (10), because the intersection and union operations of sets and the MIN and MAX operations of fuzzy numbers have commutativity and associativity. So, by Definitions 13 and 14, it follows that the equations (7), (8), (9) and (10).

For ending this section, we note that the computing complexity of FRDB algebraic operations is a polynomial under the size of relations. For instance, regarding the selection operation, since the computation time of the membership degree of fuzzy sets, MAX and MIN of the membership degrees of tuples is bounded above by some constant (Definition 1

and 3), then the cost for the interpretation of a selection condition for a tuple (Definition 8) also is some constant (i.e. $O(1)$). It results in the computing time complexity of the selection operation on a fuzzy relation (Definition 9) having n tuples is $O(n)$. Similarly, the computing time complexity of Cartesian product and join operations on two fuzzy relations having n and m tuples is $O(nm)$. Thus, we can say that the performance of FRDB model in computing and manipulating imprecise information is good and can apply it in practice. ■

5. FRDB MANAGEMENT SYSTEM

5.1. Overview of FRDB-SQLite

We have developed a management system with the query language like SQL for FRDB, called FRDB-SQLite. It is built over SQLite [28], an open-source database library written in C#, as a classical relational database management system. One can develop software using C# and employing SQLite for storing and retrieving data in classical relational databases. Small size, high performance, and simplicity in managing data are advantages of SQLite. This is the reason that we use SQLite as an underlying database management system to develop FRDB-SQLite. On top of it, we have defined classes to represent the fuzzy features of the FRDB model as well as to express and execute FRDB queries. Figure 3 shows the architecture of FRDB-SQLite comprising of three layers:

Presentation layer. This is a set of classes which controls the interactive processes of the user and FRDB-SQLite's interfaces to create schemes (schemas), relations, attributes, and implements soft queries.

Business (core) layer. This is a set of classes which performs the operation process of FRDB-SQLite. Particularly, this set of classes plays roles of handling and executing queries in FRDB-SQLite. A new query after created on the interface will be passed to the parser. The parser will define tokens and variable values in the query. The defined tokens then will be analyzed to specify relations, attributes, compared operators, logical operators, selection conditions in the query. In the analysis process, the analyzer also checks for attribute types, value types to make sure that they are proper and consistent. At last, the query executor will execute by specifying the related tuples which satisfy the selection condition of the query and unify these tuples to a new relation and output to the interface.

Data access layer. This is a set of classes for retrieving and updating data on the database management system SQLite. These classes operate when the user stores or opens a new database. Data in FRDB-SQLite when saving will be converted to strings and stored by SQLite. When processing, these data will be mapped into values of attributes's domains to perform comparison operations and calculation operations.

As presented in the previous sections, the FRDB model is effective in manipulating data due to the computing complexity of the fuzzy relational algebraic operations to be polynomials under the size of relations, SQLite has high performance in managing and handling data, thus FRDB-SQLite also is a management system to have high performance in managing and manipulating data.

With the inheritance of SQLite as an underlying database management system and complemented the classes implementing the particular fuzzy features of the FRDB, the management system FRDB-SQLite has completely components of a general relational database

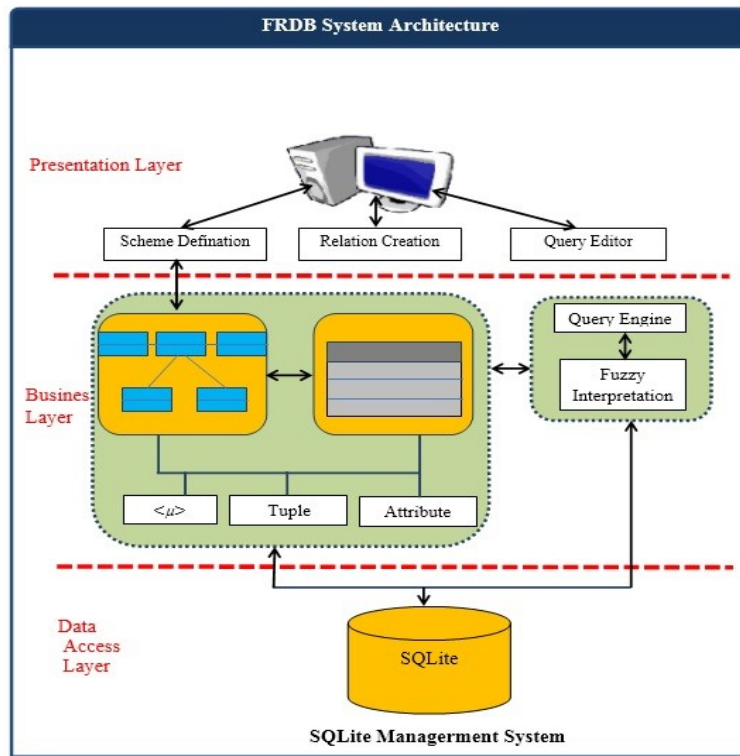


Figure 3. FRDB-SQLite Architecture

management system and users can create a new FRDB, including its schemes and relations (tables), browse and query on an existing FRDB. A created FRDB scheme can also be reused for other FRDBs. The creation of FRDB schemes, relations and the execution of the queries on relations with language like SQL in FRDB-SQLite are presented as below.

5.2. FRDB-SQLite schemes and relations

To define a new scheme in FRDB-SQLite, the user needs to input a scheme name and specify a finite set of scheme's attributes. For each attribute in a scheme, user must specify properties of it including Primary Key, Attribute Name, Data Type and Domain.

A relation in FRDB-SQLite can only be defined on an existing scheme. FRDB-SQLite relation is designed very simply and similar to that of other relational database management systems, but it has an additional tool to define and input fuzzy numbers as the membership degree of each tuple in a relation. The user can input a fuzzy number directly or insert it from a created library in the system.

In particular, the system has two methods for creating discrete fuzzy numbers and continuous fuzzy numbers, namely, Discrete Fuzzy Number Editor and Continuous Fuzzy Number Editor. A discrete fuzzy number is defined by a list of pairs (x, m) 's, where x is a real number and m is the membership degree of x for the fuzzy number. A continuous fuzzy number is assumed to be of the trapezoid shape, defined by four corner points on the coordinate system Oxy , namely, Bottom-Left($a, 0$), Top-Left($b, 1$), Top-Right($c, 1$) and Bottom-Right($d, 0$), with noting that a triangle shape also is a special trapezoid shape when the coordinates of

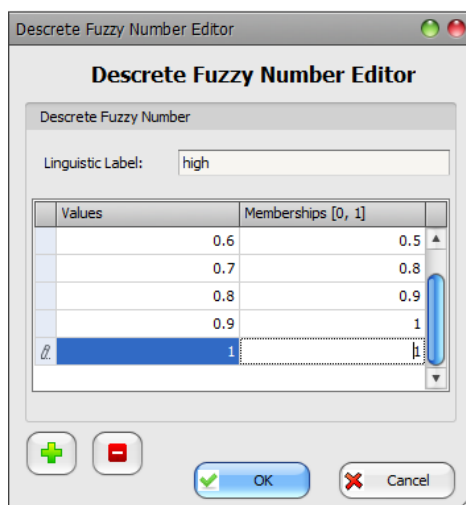


Figure 4. Discrete Fuzzy Number Editor

Top-Left and Top-Right is the same. To define a continuous fuzzy number in the system, one only needs to input the abscissas a, b, c and d for the points Bottom-Left, Top-Left, Top-Right and Bottom-Right, respectively. A created fuzzy number by Discrete Fuzzy Number Editor or Continuous Fuzzy Number Editor is associated with a linguistic label and saved in the libraries, namely, Discrete Fuzzy Numbers or Continuous Fuzzy Numbers, respectively. Figure 4 demonstrates Discrete Fuzzy Number Editor for creating the discrete fuzzy number with the linguistic label being *high*. In Figure 4, the columns of “Values” and “Membership[0, 1]” respectively display the elements and their membership degrees for the fuzzy number $high = \{0.6 : 0.5, 0.7 : 0.8, 0.8 : 0.9, 0.9 : 1.0, 1 : 1.0\}$.

When a fuzzy number with a linguistic label created in the libraries, one can browse and assign it to the membership function of the tuples in a fuzzy relation. In addition, the libraries of fuzzy numbers in the system can be updated by creating, deleting, modifying a fuzzy number as well as renaming its linguistic label.

5.3. FRDB-SQLite queries

The FRDB-SQLite supports the queries of FRDB model by giving a query editor for composing and executing queries. The FRDB-SQLite queries are much similar to those of structured query language (SQL), but they still have a little bit difference in the syntax and the simplest type as below:

```
select ⟨[ attributes ] or [*]⟩
from ⟨fuzzy relation⟩
where [fuzzy selection condition (in Definition 7)].
```

Here, “fuzzy selection condition” includes fuzzy sets created similarly as for creating fuzzy numbers presented above. A created fuzzy set by Discrete Fuzzy Set Editor or Continuous Fuzzy Set Editor is also associated with a linguistic label and saved in the libraries, namely, Discrete Fuzzy Set or Continuous Fuzzy Set, respectively. As for the libraries of fuzzy numbers, the libraries of fuzzy sets can also be updated by creating, deleting, modifying a fuzzy set and renaming the linguistic label of a fuzzy set. Figure 5 demonstrates Continuous Fuzzy

Set Editor for creating the continuous fuzzy set with the linguistic label being *young*, where “Bottom-Left: 0”, “Top-Left: 0”, “Top-Right: 20” and “Bottom-Right: 35” express four corner points Bottom-Left(0, 0), Top-Left(0, 1), Top-Right(20, 1) and Bottom-Right(35, 0) on its graph in the coordinate system Oxy . This fuzzy set is saved in the library Continuous Fuzzy Set and the its graph is generated by the system as Figure 6.

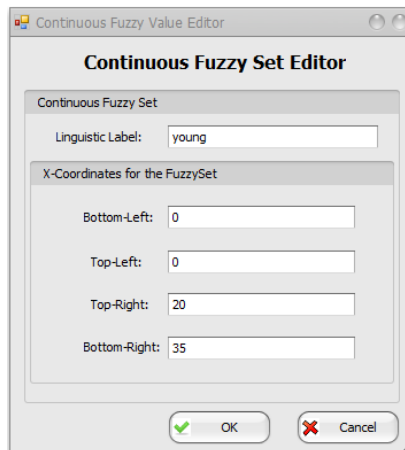


Figure 5. Continuous Fuzzy Set Editor

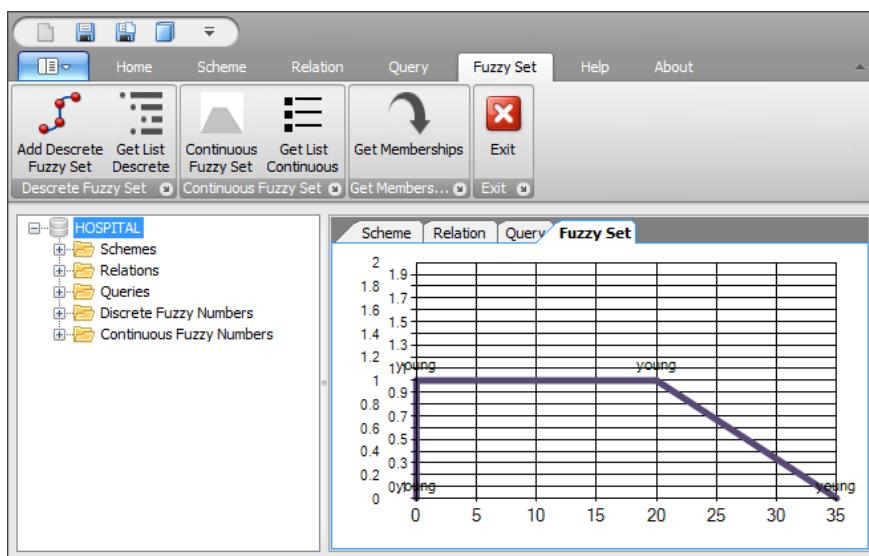


Figure 6. Graph of the fuzzy set *young*

Example 10. The query “Find the name, age, disease and daily treatment cost of all patients who are young and have hepatitis or pay at least 10 USD for the daily treatment cost” can be defined as follows:

```
select Name, Age, Disease, Cost
from Patient
```

```
where Age  $\rightarrow$  young and Disease = “hepatitis” or Cost  $\geq$  10
```

where “young” is the fuzzy set that represents the imprecise age of the patients defined as in Example 5 and created as above, the relation “Patient” has the attributes of Name, Age,

Disease and Cost (daily treatment cost). Figure 7 demonstrates the execution of the query. It is entered in the upper Textbox of the Query Editor window. There are two tuples is selected when the query execution is completed and the result of the query is shown on the lower Gridview of the Query Editor window, where “*approx.0.92*” is the fuzzy number $\{0.5 : 0, 0.6 : 0.5, 0.7 : 0.8, 0.8 : 0.9, 0.9 : 1.0, 0.93 : 1.0\}$ that represents the membership degree of the second selected tuple and computed by the query as in Example 6. It is easy to see that the result of the query also is consistent with the result of the selection operation (Definition 9) on the relation Patient with respect to the selection condition $\phi = x.Age \rightarrow \text{young}$ and $x.Disease = \text{hepatitis} \vee x.Cost \geq 10$ in the FRDB model.

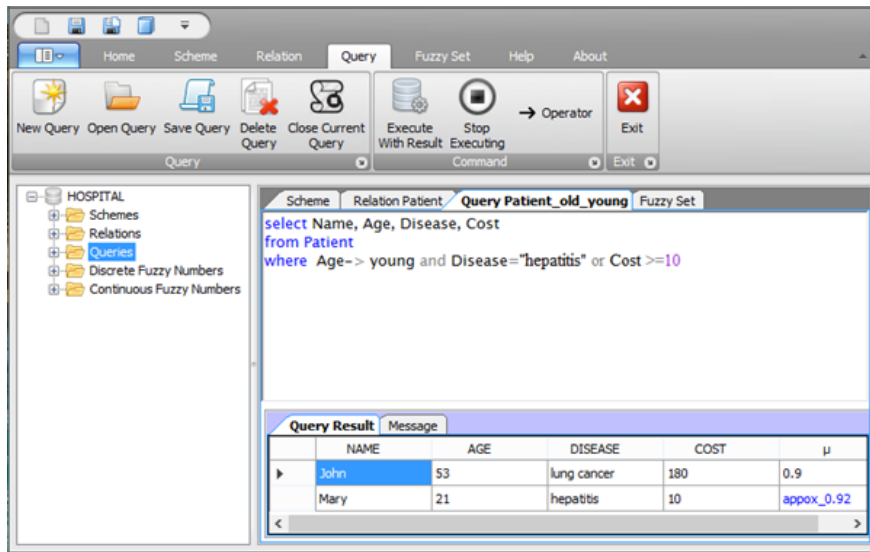


Figure 7. FRDB-SQLite query

6. CONCLUSION

In this paper, we have introduced a fuzzy relational database model (FRDB) and a management system for it. In FRDB, the membership degrees of tuples in a relation are represented by the fuzzy numbers on the interval $[0, 1]$. The data model and fuzzy relational algebraic operations in FRDB have been defined formally and consistently. Computing and associating the membership degrees of tuples in manipulating of the algebraic operations are implemented by the operations MAX and MIN using the extension principle. FRDB allows expressing and executing the soft queries that are associated with fuzzy sets for dealing with imprecise information in real databases. An effective management system for FRDB with the familiar querying language like SQL has been built by using the open-source management system SQLite. The new management system inherited SQLite as an underlying database management system and complemented the classes implementing the particular fuzzy features of the FRDB, has completely components of a general relational database management system and can be applied in practice.

In the next steps, we will extend the FRDB model and management system with the aggregate and grouping operations for aggregating and grouping of attribute values in fuzzy relations.

REFERENCES

- [1] E. F. Codd, "A relational model of data for large shared data banks", *Communications of the Association for Computing Machinery*, vol. 13, no. 6, 1970, pp. 377–387. https://doi.org/10.1007/978-3-642-59412-0_16
- [2] L.A. Zadeh, "Fuzzy sets", *Information and Control*, vol. 8, 1965, pp. 338–353.
- [3] G.J. Klir, and B. Yuan, *Fuzzy Sets and Fuzzy Logic—Theory and Applications*, Prentice Hall PTR, 1994.
- [4] B. P. Buckles, and F. Petry, "A fuzzy representation of data for relational databases", *Fuzzy Sets and Systems*, vol. 7, no. 3, May 1982, pp. 213–226. [https://doi.org/10.1016/0165-0114\(82\)90052-5](https://doi.org/10.1016/0165-0114(82)90052-5)
- [5] S. Shenoi, and A. Melton, "Proximity relations in the fuzzy relational database model", *Fuzzy Sets and Systems*, vol. 100, no. 3, 1999, pp. 285–296. [https://doi.org/10.1016/0165-0114\(89\)90201-7](https://doi.org/10.1016/0165-0114(89)90201-7)
- [6] H. Prade, and C. Testemale, "Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries", *Information Sciences*, vol. 34, no. 2, 1984, pp. 115–143. [https://doi.org/10.1016/0020-0255\(84\)90020-3](https://doi.org/10.1016/0020-0255(84)90020-3)
- [7] M. Umamo, and S. Fukami, "Fuzzy relational algebra for possibility-distribution-relational model of fuzzy data", *J Intell Inf Syst*, vol. 3, 1994, 7–27. <https://doi.org/10.1007/BF01014018>
- [8] J.M. Medina, O. Pons, M.A.Vila, "GEFRED: A generalized model for fuzzy relational databases", *Information Sciences*, vol. 76, no. 1–2, 1994, pp. 87–109. [https://doi.org/10.1016/0020-0255\(94\)90069-8](https://doi.org/10.1016/0020-0255(94)90069-8)
- [9] J.C. Cubero, J.M. Medina, O. Pons, and M.A. Vila, "Extensions of a resemblance relation", *Fuzzy Sets and Systems*, vol. 86, no. 2, 1 March 1997, pp 197–212. [https://doi.org/10.1016/0165-0114\(95\)00398-3](https://doi.org/10.1016/0165-0114(95)00398-3)
- [10] X. Tang, and G. Chen, "A complete set of fuzzy relational algebraic operators in fuzzy relational databases", *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, 2004, pp. 565–569, vol.1. Doi: 10.1109/FUZZY.2004.1375797.
- [11] X. Tang and G. Chen, "Equivalence and transformation of extended algebraic operators in fuzzy relational databases", *Fuzzy Sets and Systems*, vol. 157, no. 12, 2006, pp.1581–1596. <https://doi.org/10.1016/j.fss.2005.12.009>
- [12] N.A. Chaudhry, J.R. Moyne, and E.A. Rundensteiner, "A design methodology for databases with uncertain data", *Seventh International Working Conference on Scientific and Statistical Database Management*, 1994, pp. 32–41. Doi: 10.1109/SSDM.1994.336964.
- [13] S. Parsons, "Current approaches to handling imperfect information in data and knowledge bases", *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 3, pp. 353–372, June 1996. Doi: 10.1109/69.506705.
- [14] F. E. Petry, *Fuzzy databases: Principles and applications*, Kluwer Academic Publishers, 1996.
- [15] P. Bosc, and O. Pivert, *Fuzzy databaes. Handbook of Fuzzy Computation*, IOP Publishing Ltd, 1999.

- [16] X. Meng, Z.M. Ma, and X. Zhu, “A Knowledge-based fuzzy query and results ranking approach for relational databases”, *Journal of Computational Information Systems*, vol. 6, pp. 2037–2044, 2010.
- [17] S. Chakraborty, “Codd’s relational data model and fuzzy logic: a practical approach to find the computer solution”, *International Journal of Advanced Technology & Engineering Research*, vol. 2, no. 4, pp. 21–27, 2012.
- [18] J. Mishra, and S. Ghosh, “Uncertain query processing using vague set or fuzzy set: which one is better?”, *International Journal of Computers Communications & Control*, vol. 9, pp. 730–740, 2014.
- [19] O. Pivert, and H. Prade, “Dealing with aggregate queries in an uncertain database model based on possibilistic certainty”, In: Laurent A., Strauss O., Bouchon-Meunier B., Yager R.R. (eds) *Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2014. Communications in Computer and Information Science*, vol 444. Springer, Cham. https://doi.org/10.1007/978-3-319-08852-5_16
- [20] A.A. Sabour, A.M. Gadallah, and H.A. Hefny, “Flexible querying of relational databases: fuzzy set based approach”, Taher Azar A. (eds), *Advanced Machine Learning Technologies and Applications. AMLTA 2014. Communications in Computer and Information Science*, vol 488. Springer, Cham. https://doi.org/10.1007/978-3-319-13461-1_42
- [21] H. Nguyen, “A fuzzy relational database model”, *Journal of Information and Communications Technology (Vietnam Ministry of Information and Communication Technology)*, vol. 5, no. 1, pp. 37–45, 2015 (in Vietnamese).
- [22] H. Nguyen, “A type-2 fuzzy relational database model”, *Journal of Information & Communication Technology: Research and Development on Information & Communication Technology*, vol. E-3, no. 14, pp. 19–26, 2017.
- [23] H. Nguyen, “Extending relational database model for uncertain information”, *Journal of Computer Science and Cybernetics*, vol.35, no. 4, pp.355–372, 2019.
- [24] L. Yan, and Z.M. Ma, “A probabilistic nested relational database model with fuzzy probability measures”, in *Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, New Orleans, Louisiana, USA, 2019, pp.253–257.
- [25] A. Niewiadomski, “A type-2 fuzzy approach to linguistic summarization of data”, in *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 198–212, Feb. 2008. Doi: 10.1109/TFUZZ.2007.902025.
- [26] P. C. Saxena, and D.K. Tayal, “Normalization in type-2 fuzzy relational data model based on fuzzy functional dependency using fuzzy functions”, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 20, no. 1, pp. 99–138, 2012. <https://doi.org/10.1142/S0218488512500067>
- [27] S. M. Darwish, T.F. Mabrouk, and Y.F. Mokhtar, “Enriching vague queries by type-2 fuzzy orderings”, *Lecture Notes on Information Theory*, vol. 2, pp.177–185, 2014. Doi:10.12720/lnit.2.2.177-185
- [28] J. A. Kreibich, *Using SQLite: small. fast. reliable. choose any three*, O’Reilly Media, 2010.

Received on August 08, 2020

Accepted on May 13, 2021