ISSN No. 0976-5697

Volume 2, No. 5, Sept-Oct 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# Answering imprecise queries in Total Neutrosophic Databases

Meena Arora*
CSE Department
JSS Academy of Technical Education,
Noida, INDIA
Meena23dec@gmail.com

Dr. U.S.Pandey
Delhi University
Delhi, INDIA
Uspandey1@gmail.com

***Abstract*****—**Neutrosophic relation database model has been developed for representing and manipulating three kinds of uncertain information in databases: fuzzy, incomplete and inconsistent. The neutrosophic set is a powerful general formal framework that has been recently proposed. However, the neutrosophic set needs to be specified from a technical point of view. In order to handle inconsistent situation, we propose the relation-theoretic operations on them. We define algebraic operators that are generalizations of the usual operators such as intersection, union, selection, and join on fuzzy relations. We present an SQL–like SELECT statement construct for posing queries to total neutrosophic databases. The syntax and semantics of SELECT statement is defined for making it an effective tool for querying.

***Keywords-*** component; Total neutrosophic relation, Neutrosophic sets, Doubt factor, Belief factor.

## I. INTRODUCTION

Essentially all the information in the real world is imprecise, here imprecise means fuzzy, incomplete and even inconsistent. There are many theories existing to handle such imprecise information, such as fuzzy set theory, probability theory, probability theory, intuitionistic fuzzy set theory, vague theory, etc. These theories can only handle one aspect of imprecise problem but not the whole in one framework. For example, fuzzy set theory can only handle fuzzy, vague information not the incomplete and inconsistent information.

In this paper, we unify the above-mentioned theories under one framework. Under this framework, we can not only model and reason with fuzzy, incomplete information but also inconsistent information without danger of trivialization.

Relational data model, proposed by Ted Codd's pioneering paper [2] usually takes care of only well-defined and unambiguous data. However, imperfect information is ubiquitous, almost all the information that we have about the real world is not certain, complete and precise [10]. Imperfect information can be classified as: incompleteness, imprecision, uncertainty, inconsistency.

In order to represent and manipulate various forms of incomplete information in relational databases, several extensions of the classical relational model have been proposed [1, 3, 5, 8, 12, 13]. In some of these extensions, a variety of "null values" have been introduced to model unknown or not-applicable data values. Attempts have also been made to generalize operators of relational algebra to manipulate such extended data models [1, 3, 5, 6, 7].

Probability, possibility and Dempster-Shafer theory have been proposed to deal with uncertainty. Possibility theory [8] is built upon the idea of a fuzzy restriction. Wong [4] proposes a method that quantifies the uncertainty in a database using probabilities. Carvallo and Pittarelli [9] also use probability theory to model uncertainty in relational databases systems.

However, unlike incomplete, imprecise and uncertain information, inconsistent information has not enjoyed enough research attention. In fact, inconsistent information exists in a lot of applications.

*For example*, in data warehousing application, inconsistency will appear when trying to integrate the data from many different sources. Another *example* is that in the expert system, there exist facts which are inconsistent with each other.

We introduce neutrosophic relations and algebraic operators over neutrosophic relations that extend the standard operators such as selection, join, and union over vague relations. There are many potential applications of our new data model e.g. in Web mining, Bioinformatics, Decision Support System.

In this paper, we present an extension of the SQL SELECT statement for querying such databases. The syntax of this extended statement is similar to that of the ordinary SELECT statement; the semantics that we propose is quite different. With our new extended semantics, the statement becomes an effective tool for querying neutrosophic relational data model

The remainder of this paper is organized as follows. Section 2 presents a brief introduction of neutrosophy, neutrosophic sets. Section 3 gives a quick overview of total neutrosophic relations. Section 4 presents generalized algebra on neutropsophic relations with relational theoretic operators. Section 5 presents the syntax &new semantics of SQL-like SELECT statement for querying neutrosophic databases based on algebraic operators that are defined in section 4. Section 6 contains an example SELECT statement and a walk through the evaluation procedure for that query. Section 7 presents the area of application where this can be applied in real life. Finally, Section 8 concludes the paper with some mention of related and future work directions.

## II. NEUTROSOPHIC LOGIC AND NEUTROSOPHIC SETS

### *A. Neutrosophic Logic:*

Neutrosophic logic was created by Florentin Smarandache (1995) [11] and is an extension/combination of the fuzzy logic, intuitionistic logic, paraconsistent logic, and the three-valued logics that use an indeterminate value.

Definition 1 Neutrosophic Logic :A logic in which each proposition is estimated to have the percentage of truth in a subset T, the percentage of indeterminacy in a subset I, and the percentage of falsity in a subset F, is called *Neutrosophic Logic*. T, I, F are standard or non-standard subsets of the nonstandard interval $]^{-}0, 1^{+}[$, where $n_{inf} = \inf T + \inf I + \inf F \geq {}^{-}0$, and $n_{sup} = \sup T + \sup I + \sup F \leq 3^{+}$.

Definition 2 (Neutrosophic Set): Let X be a space of points (objects), with a generic element in X denoted by x. A neutrosophic set A in X is characterized by a truth-membership function $T_A$, an indeterminacy-membership function $I_A$ and a falsity-membership function $F_A$. $T_A(x)$, $I_A(x)$ and $F_A(x)$ are real standard or non-standard subsets of $]^{-}0, 1^{+}[$. That is

$$T_A: X \rightarrow ]^{-}0, 1^{+}[ \quad (1)$$
$$I_A: X \rightarrow ]^{-}0, 1^{+}[ \quad (2)$$
$$F_A: X \rightarrow ]^{-}0, 1^{+}[ \quad (3)$$

There is no restriction on the sum of $T_A(x)$, $I_A(x)$ and $F_A(x)$ so ${}^{-}0 \leq \sup T_A(x) + \sup I_A(x) + \sup F_A(x) \leq 3^{+}$.

### *B. Operations with sets:*

Let $S_1$ and $S_2$ be two (unidimensional) real standard or non-standard subsets, then one defines [11]

Addition of sets:

$S_1 + S_2 = \{x|x=s_1+s_2$ , where $s_1 \in S_1$ and $s_2 \in S_2\}$,

with $\inf S_1 + S_2 = \inf S_1 + \inf S_2$ , $\sup S_1 + S_2 = \sup S_1 + \sup S_2$;

Subtraction of sets:

$S1 - S_2 = \{x|x= s_1 - s_2$, where $s_1 \in S_1$ and $s_2 \in S_2\}$.

For real positive subsets (most of the cases will fall in this range) one gets

$\inf S_1 - S_2 = \inf S_1 - \sup S_2$, $\sup S_1 - S_2 = \sup S_1 - \inf S_2$;

Multiplication of sets:

$S_1 . S_2 = \{x|x= s_1 . s_2$, where $s_1 \in S_1$ and $s_2 \in S_2\}$.

For real positive subsets (most of the cases will fall in this range) one gets

$\inf S_1 . S_2 = \inf S_1 . \inf S_2$ , $\sup S_1 . S_2 = \sup S_1 . \sup S_2$;

Division of a set by a number:

Let $k \in R^*$, then $S_1 \emptyset k = \{ x|x= s_1/k$, where $s_1 \in S_1\}$.

For all neutrosophic set operations: if, after calculations, one obtains numbers < 0 or > 1, one replaces them by ${}^{-}0$ or $1^{+}$ respectively.

Definition 3. (Complement) The complement of a neutrosophic set A is denoted by c(A) and is defined by

$$T_{c(A)}(x) = \{1^{+}\} - T_A(x), \quad (4)$$
$$I_{c(A)}I(x) = \{1^{+}\} - I_A(x), \quad (5)$$
$$F_{c(A)}(x) = \{1^{+}\} - F_A(x), \quad (6)$$

for all x in X.

Definition 4. (Union) The union of two neutrosophic sets A and B is a neutrosophic set C, written as C = A∪ B, whose truth-membership, indeterminacy-membership and falsity-membership functions are related to those of A and B by

$$T_C(x) = T_A(x) + T_B(x) - T_A(x) \times T_B(x), \quad (7)$$
$$I_C(x) = I_A(x) + I_B(x) - I_A(x) \times I_B(x), \quad (8)$$
$$F_C(x) = F_A(x) + F_B(x) - F_A(x) \times F_B(x), \quad (9)$$

for all x in X.

Definition5. (Intersection) The intersection of two neutrosophic sets A and B is a neutrosophic set C, written as C = A ∩ B, whose truth-membership, indeterminacy-membership and falsity-membership functions are related to those of A and B by

$$T_C(x) = T_A(x) \times T_B(x), \quad (10)$$
$$I_C(x) = I_A(x) \times I_B(x), \quad (11)$$
$$F_C(x) = F_A(x) \times F_B(x), \quad (12)$$

for all x in X.

## III. TOTAL NEUTROSOPHIC RELATIONS

In this section, we introduce neutrosophic relations. A tuple in a neutrosophic relation is assigned a measure $\langle \alpha, \beta \rangle$ , $0 \leq \alpha, \beta \leq 1$.

Definition 6 Belief factor $\alpha$ :The interpretation of this measure is that we believe with confidence $\alpha$ that the tuple is in the relation.

In a neutrosophic relation R , $R(t)^{+}$ is the belief factor assigned to t by R.

Definition 7. Doubt factor**:** The interpretation of this measure is that we doubt with confidence $\beta$ that the tuple is in the relation.

In a neutrosophic relation R , $R(t)^{-}$ is the doubt factor assigned to t by R.

The belief and doubt confidence factors for a tuple need not add to exactly 1. This allows for incompleteness and inconsistency to be represented. If the belief and doubt factors add up to less than 1, we have incomplete information regarding the tuple's status in the relation and if the belief and doubt factors add up to more than 1, we have inconsistent information regarding the tuple's status in the relation.

In contrast to fuzzy relations where the grade of membership of a tuple is fixed, neutrosophic relations bound the grade of membership of a tuple to a subinterval $[\alpha, 1-\beta]$ for the case $\alpha + \beta \leq 1$.

We now formalize the notion of a neutrosophic relation. Let a relation scheme (or just scheme) Σ be a finite set of attribute names, where for any attribute name A ∈Σ, dom(A) is a non-empty set of distinct values for A. A tuple on Σ is any total map $t: \Sigma \rightarrow \cup_{A \in \Sigma} dom(A)$, such that t(A) ∈ dom(A), for each A ∈Σ.

Let

$\tau(\sum)$ denotes the set of all tuples on any scheme$\sum$.

$T(\sum)$ be the set of all total neutrosophic relations on $\sum$.

$R(t)^{+}$is the belief factor,

$R(t)^{-}$ is the doubt factor ,

$C(\sum)$ be the set of all consistent neutrosophic relations on Σ.

$\mathcal{V}(\Sigma)$ be the set of all neutrosophic relations on$\sum$.

Definition 8. A neutrosophic relation R on scheme $\sum$ is any subset of $\tau(\sum) \times [0,1] \times [0,1]$.

For any $t \in \tau(\sum)$, we shall denote an element of R as $\langle t, R(t)^{+}, R(t)^{-} \rangle$ where $R(t)^{+}$ is the belief factor assigned to t by R and $R(t)^{-}$ is the doubt factor assigned to t by R. Let $\mathcal{V}(\Sigma)$ be the set of all neutrosophic relations on $\sum$.

Definition 9. A neutrosophic relation R on scheme $\sum$ is consistent if $R(t)^{+} + R(t)^{-} \leq 1$, for all $t \in \tau(\Sigma)$.

Let $C(\sum)$ be the set of all consistent neutrosophic relations on $\Sigma$. R is said to be complete if $R(t)^{+} + R(t)^{-} \geq 1$, for all $t \in \tau(\Sigma)$. If R is both consistent and complete, i.e. $R(t)^{+} + R(t)^{-} = 1$, for all $t \in \tau(\Sigma)$, then it is a **total neutrosophic relation**, and let $T(\sum)$ be the set of all total neutrosophic relations on $\sum$.

For any $t \in \tau(\Sigma)$, we shall denote an element of R as <t, $R(t)^+$, $R(t)^-$>, where $R(t)^+$ is the belief factor assigned to t by R and $R(t)^-$ is the doubt factor assigned to t by R. Note that since contradictory beliefs are possible, so $R(t)^+ + R(t)^-$ could be greater than 1. Furthermore, $R(t)^+ + R(t)^-$ could be less than 1, giving rise to incompleteness.

As an *example*, suppose in the e-shopping environment, there are two items $Item_1$ and $Item_2$, which are evaluated by customers for some categories of quality - Capability, Trustworthiness and Price. Let the evaluation results are captured by the following total neutrosophic relation EVAL_RESULT on scheme {Item_Name, Quality_Category} as shown in Table I:

Table: 1 Eval Result

| ITEM_Name | Quality_Category | Evaluation |
|---|---|---|
| $Item_1$ | Capability | ⟨0.9,0.2⟩ |
| $Item_1$ | Trustworthiness | ⟨1.0,0.0⟩ |
| $Item_1$ | Price | ⟨0.1,0.8⟩ |
| $Item_2$ | Capability | ⟨1.0,1.0⟩ |
| $Item_2$ | Price | ⟨0.8,0.3⟩ |

The above relation contains the information that the confidence of $Item_1$ was evaluated “good” for category Capability is 0.9 and the doubt is 0.2. The confidence of $Item_1$ was evaluated “good” for category Trustworthiness is 1.0 and the doubt is 0.0. The confidence of $Item_1$ was evaluated “poor” for category Price is 0.8 and the doubt is 0.1. Also, the confidence of $Item_2$ was evaluated “good” for category Capability is 1.0 and the doubt is 1.0 (similarly, the confidence of $Item_2$ was evaluated “poor” for category Capability is 1.0 and the doubt is 1.0). The confidence of $Item_2$ was evaluated “good” for category Price is 0.8 and the doubt is 0.3. Note that the evaluation results of $Item_2$ for category Trustworthiness is unknown. The above information contains results of fuzziness, incompleteness and inconsistency. Such information may be due to various reasons, such as evaluation not conducted, or evaluation results not yet available, the evaluation is not reliable, and different evaluation results for the same category producing different results, etc.

## IV. ALGEBRA ON TOTAL NEUTROSOPHIC RELATIONS

In this section, we briefly introduce relational theoretic operators (natural join, projection, product, selection) for the semantics of SELECT statement in queries used in Neutrosopic Search. These generalized operators maintain the belief system intuition behind neutrosophic relations.

### A. *Relation-Theoretic Operators:*

We now define some generalized relation-theoretic algebraic operators (like join, product, selection, projection) on total neutrosophic relations to complete the semantics of SELECT statement. To reflect such generalizations a subscript ‘t’ is placed aside on an ordinary relation operator to obtain corresponding total neutrosophic relational operator.

**Definition 10** Let R and S be neutrosophic relations on scheme $\sum$. Then,

The union operator can be obtained as follows: Given a tuple t, since we believed that it is present in the relation R with confidence $R(t)^{+}$ and that it is present in the relation S with confidence $S(t)^{+}$, we can now believe that the tuple t is present in the “either-R-or- S" relation with confidence which is equal to the larger of $R(t)^{+}$ and $S(t)^{+}$. Using the same logic, we can now believe in the absence of the tuple t from the “either-R-or-S" relation with confidence which is equal to the smaller (because t must be absent from both R and S for it to be absent from the union) of $R(t)^{-}$ and $S(t)^{-}$.

$(R \cup^{t} S)(t) = \langle \max\{R(t)^{+}, S(t)^{+}\}, \min\{R(t)^{-}, S(t)^{-}\} \rangle$, for any $t \in \boldsymbol{\tau}(\boldsymbol{\Sigma})$;

**Definition 11** Let R and S be neutrosophic relations on schemes $\Sigma$ and $\Delta$, respectively. Then, the **natural join** (further for short called join) of R and S, denoted $R \bowtie^{t} S$, is a total neutrosophic relation on scheme $\Sigma \cup \Delta$, given by

$(R \bowtie^{t} S)(t) = \langle \min\{R(\boldsymbol{\pi}_{\boldsymbol{\Sigma}}(t))^{+}, S(\boldsymbol{\pi}_{\Delta}(t))^{+}\}, \max\{R(\boldsymbol{\pi}_{\boldsymbol{\Sigma}}(t))^{-}, S(\boldsymbol{\pi}_{\Delta}(t))^{-}\} \rangle$,

Where $\pi$ is the usual projection of a tuple.

Similar to the intersection operator, the minimum of the belief factors and the maximum of the doubt factors are used in the definition of the join operation.

We now define the projection operator on total Neutrosophic relation.

**Definition 12** Let R be a neutrosophic relation on scheme $\Sigma$ and $\Delta \subseteq \Sigma$. Then, the **projection** of R onto $\Delta$, denoted by $\pi_{\Delta}^{t}(R)$, is a total neutrosophic relation on scheme $\Delta$, given by

$(\boldsymbol{\pi}_{\boldsymbol{\Delta}}^{\boldsymbol{t}}(R))(t) = \langle \max\{R(u)^{+} \mid u \in \boldsymbol{t}^{\boldsymbol{\Sigma}}\}, \min\{R(u)^{-} \mid u \in \boldsymbol{t}^{\boldsymbol{\Sigma}}\} \rangle$

The belief factor of a tuple in the projection is the maximum of the belief factors of all of the tuple's extensions onto the scheme of the input neutrosophic relation. Moreover, the doubt factor of a tuple in the projection is the minimum of the doubt factors of all of the tuple's extensions on to the scheme of the input neutrosophic relation.

**Definition 13** For any total Neutrosophic relation R and S, $R \times^{t} S = t(R) \bowtie^{t} t(S)$.

The **product** of total Neutrosophic relation R and S is essentially a join after renaming their attributes to make their schemes disjoint. Let $t$(R) be the total Neutrosophic relation in totality with the same tuples in R, but with attribute names of the form "R.A" for each attribute name A of R.

We will now define the **selection operator** on total neutrosophic relations.

**Definition 14** Let R be a total neutrosophic relation on scheme $\sum$, and C be a condition on tuples of $\sum$ denoted $\langle t_c(t),\ f_c(t)\rangle$. Then, the selection of R by C, denoted by $\sigma_C^t$(R ) is a total neutrosophic relation on scheme $\sum$, given by

( $\sigma_C^t$(R ) ) (t) = $\langle \min R(t)^+, t_c(t), \max R(t)^-, f_C(t) \rangle$.

In the generalized SELECT statement, we let the condition occurring in the where clause be infinite valued. The infinite values, except $\langle 1, 0\rangle$ or $\langle 0, 1\rangle$, arise essentially due to any nested subqueries.

For any arithmetic expressions $E_1$ and $E_2$, comparisons such as $E_1 \leq E_2$ are simply 2-valued conditions

( $\langle 1,0\rangle$ or $\langle 0,1\rangle$ ) . Let $\xi$ be a subquery of the form (select …… from ……. where …….) occurring in the where clause of a SELECT statement. And let R be the neutrosophic relation on scheme $\sum$ that the subquery $\xi$ evaluates to. Then, conditions involving the subquery $\xi$ evaluate as follows.

1. The condition exists $\xi$

evaluates to $\langle \alpha, \beta\rangle$

$\alpha$ = max {a} , a = $R(t)^+$ , for all t $\in \tau(\sum)$,

$\beta$ = min {b}, b = $R(t)^-$ , if $R(t)^+ + R(t)^- \leq 1$, b = 1 – $R(t)^+$ , if $R(t)^+ + R(t)^- > 1$ , for all t $\in \tau(\sum)$ .

2. For any tuple t $\in \tau(\sum)$, the condition

T in $\xi$

Evaluates to $\Phi_R$ (t).

3. If $\sum$ contains exactly one attribute, then for any (scalar value) t $\in \tau(\sum)$, the condition

t > any $\xi$

evaluates to $\langle \alpha, \beta\rangle$,

$\alpha$ = max {a} , a = $R(k)^+$ , if t > k, for some k $\in$ R, ($\beta$ = min {b}, b = $R(k)^-$ , if $R(k)^+ + R(k)^- \leq 1$,

B = 1 – $R(k)^+$ , if $R(k)^+ + R(k)^- > 1$ ) , if t >k, for some k $\in$ R ;

$\alpha$ = 0, $\beta$ = 1, otherwise

Note that conditions involving such operators never evaluate to the value $\alpha, \beta$ such that

$\alpha + \beta > 1$.

We complete our semantics for conditions by defining the **not**, **and** and **or** operators on them. Let C and D be any conditions, and value of C = <$t_c$, $f_c$> and value of D = <$t_d$, $f_d$>. Then, the value of the condition not C is given by

**not C** = <$f_c$, $t_c$ >

while the value of the condition C and D is given by

**C and D** = <min { $t_c$, $t_d$}, max{ $f_c$, $f_d$}>

and that of the condition C or D is given by

**C or D** = <max { $t_c$, $t_d$}, min{ $f_c$, $f_d$}>

The duality of *and* and *or* is evident from their formulas. The following algebraic laws exhibited by the above operators:

1. Double Complementation Law: not (not C) = C
2. Identity and Idempotence Laws:

C and <1, 0> = C and C = C

C or <0, 1> = C or C = C

3. Commutativity Laws:

C and D = D and C

C or D = D or C

4. Associativity Laws:

C and (D and E) = (C and D) and E

C or (D or E) = (C or D) or E

5. Distributivity Laws:

C and (D or E) = (C and D) or (C and E)

C or (D and E) = (C or D) and (C or E)

6. De Morgan Laws:

not (C and D) = (not C) or (not D)

not (C or D) = (not C) and (not D)

## V. SYNTAX AND SEMANTICS OF SELECT QUERIES FOR TOTAL NEUTROSOPHIC RELATIONS.

The most popular construct for information retrieval from most commercial systems is the SQL's SELECT statement. While the statement has many options and extensions to its basic form, here we just present generalization for total neutrosophic relations. The basic form of the statement contains three clauses select, from and where, and has the following format:

Select $A_1, A_2$ , …. $A_m$ From $R_1$ , $R_2$ , …., $R_n$ Where C

where

1. $A_1, A_2$ , ………. $A_m$ is a list of attribute names whose values are to be retrieved by the query,

2. $R_1$ , $R_2$ , ………., $R_n$ is a list of relation names required to process the query, and

3. C is a boolean expression that identifies the tuples to be retrieved by the query.

Without loss of generality, we assume that each attribute name occurs in exactly one relation, because if some attribute $A_i$ occurs in more than one relation, we require, instead of simply the attribute $A_i$, a pair of the form $R_j.A_i$ qualifying that attribute.

The result of the SELECT statement is a relation with attributes $A_1, A_2$ , ………. $A_m$ chosen from the attributes of $R_1 \times R_2 \times \ldots\ldots\ldots \times R_n$ for tuples that satisfy the Boolean condition C, i.e.

$\pi_{A1,A2,\ldots\ldots\ldots Am}(\sigma_C (R_1 \times R_2 \times \ldots\ldots\ldots \times R_n))$

where $\pi, \sigma, \times$ and _ are the projection, selection and product operations, respectively, on ordinaryrelations. We retain the above syntax in the generalized SELECT statement for the total neutrosophic relations. However, the relation names $R_1$ , $R_2$ , ………., $R_n$ now represent some neutrosophic relations and C is some infinite-valued condition. The result of the generalized SELECT statement is then the value of the algebraic expression:

$\pi^t_{A1,A2,\ldots\ldots\ldots Am}(\sigma^t_C( R_1 \times^t R_2 \times^t \ldots\ldots\ldots \times^t R_n ))$

Where $\pi^t$ , $\sigma^t$ , $\times^t$ are, respectively, the projection, selection and product operations on total neutrosophic relations. Furthermore, the result of the generalized SELECT statement is also a total neutrosophic relation.

## VI. AWALKTHROUGH OF THE EVALUATION FOR AN EXAMPLE

Let us now consider a **query**:

What items showed contradictory evaluation of some category of quality in the total neutrosophic relation EVAL as shown in TABLE II on scheme {ITEM_Name, Quality_Category} of the item-quality .

Table: 2 Eval Result

| ITEM_Name | Quality_Category | Evaluation |
|---|---|---|
| $Item_1$ | Capability | $\langle 0.9,0.2\rangle$ |
| $Item_1$ | Trustworthiness | $\langle 1.0,0.0\rangle$ |
| $Item_1$ | Price | $\langle 0.1,0.8\rangle$ |
| $Item_2$ | Capability | $\langle 1.0,1.0\rangle$ |
| $Item_2$ | Price | $\langle 0.8,0.3\rangle$ |

***Solution:***

A SELECT statement for this query is:

Select Item_Name from EVAL_RESULT where not ((Item_name, Quality_Category) in EVAL_RESULT)

One possible evaluation method for the above query in ordinary 2-valued SQL is to produce the Item attribute of those rows of EVAL_RESULT that satisfy the where condition. Since the where condition in the above case is exactly that row not be in EVAL_RESULT, in 2-valued logic the above query will produce an empty answer.

The stepwise output is shown below :

In neutrosophic logic, however, the where condition needsto be evaluated, to one of infinite possible values, for every possible row with attributes Σ = (Item_name, Quality_Category).

The result is then combined with EVAL_RESULT according to the semantics of $\sigma^t$, on which $\pi^t$is performed to produce the resulting total neutrosophic relation.

Therefore, for each of the 6 rows in τ(Σ), we first evaluate the where condition C is as shown in Table III:

Table: 3 Relation Schema with Condition

| ( Item_Name, Quality_Category) | C= not ((( Item_Name, Quality_Category) in EVAl_RESULT) |
|---|---|
| ( $Item_1$, Capability ) | $\langle 0.2, 0.9\rangle$ |
| ( $Item_1$, Trustworthiness) | $\langle 0.0,1.0\rangle$ |
| ( $Item_1$, Price ) | $\langle 0.8,0.1\rangle$ |
| ( $Item_2$, Capability ) | $\langle 1.0,1.0\rangle$ |
| ( $Item_2$, Trustworthiness ) | $\langle 0.0,0.0\rangle$ |
| ( $Item_2$, Price ) | $\langle 0.3,0.8\rangle$ |

Now, $\sigma_C^t$(EVAL) according to the definition of $\sigma^t$evaluates to the total neutrosophic relation is as shown in Table IV.

$\sigma_C^t$(EVAL)

Table: 4 Relation Schema with Select Clause

| ITEM_Name | Quality_Category | Evaluation |
|---|---|---|
| $Item_1$ | Capability | $\langle 0.2, 0.9\rangle$ |
| $Item_1$ | Trustworthiness | $\langle 0.0,1.0\rangle$ |
| $Item_1$ | Price | $\langle 0.1,0.8\rangle$ |
| $Item_2$ | Capability | $\langle 1.0,1.0\rangle$ |
| $Item_2$ | Price | $\langle 0.3,0.8\rangle$ |

Finally, $\pi^t$of the above is the total neutrosophic relation as shown in Table V:

$\pi_I^t(\sigma_C^t$(EVAL))

Table : 5 Relation Schema with Project Clause

| ITEM_Name | Evaluation |
|---|---|
| $Item_1$ | $\langle 0.1,0.8\rangle$ |
| $Item_2$ | $\langle 1.0,0.0\rangle$ |

which is the result of the SELECT statement.

Explanation :

The result states that $Item_1$ showed contradictory evaluation result for some category with confidence is 0.1 and doubt is 0.8, so it is safe to conclude that $Item_1$ did not show contradictory evaluation result, but $Item_2$ showed contradictory evaluation result for some category with confidence 1.0 and doubt is 0.0, the explanation is that $Item_2$ did show contradictory result for some category and did not show contradictory for other category at the same times.

## VII. APPLICATION

Web services are playing an important role in e-business application integration and other application fields such as bioinformatics. So it is crucial for the success of both service providers as well as service consumers to provide and invoke the high quality of service (QoS) Web services. Since different application domains have different requirements for QoS it is impractical to use classical mathematical modeling methods to evaluate the QoS of semantic Web services. Our model is scalable to handle fuzzy, uncertain and inconsistent QoS metrics effectively. For example, capability of a Web service is fuzzy. It is unreasonable to use crisp values to describe it. So we can use several linguistic variables such as a "little bit low" and "a little bit high" to express the capability of services.

## VIII. CONCLUSIONS

How to model and reason with fuzzy, incomplete and even inconsistent information is an important research topic. In this paper we have presented syntax & semantics for the SQL

SELECT statement for querying neutrosophic databases. We have presented a generalization of fuzzy relations, intuitionistic fuzzy relations (interval valued fuzzy relations) and vague relations, called total neutrosophic relations, in which we allow the representation of confidence (belief and doubt) factors with each tuple.

We introduced generalized operators on total neutrosophic relations. These generalized operators maintain the belief system intuition behind neutrosophic relations. We have given the syntax & extended semantics for the SELECT statement based on the extended algebraic operators (projection, selection and product) on neutrosophic relations.

As the future work, we plan to extend this work to develop tuple-relational and domain-relational calculus for the total neutrosophic relations.

## IX. REFERENCES


[1]. D. Maier. "The Theory of Relational Databases" . Computer Science Press , Rockville, Maryland, 1983.

[2]. E.F. Codd. "A relational model for large shared data banks". Communications of the ACM, Volume -13, Number 6, Pages 377-387, June 1970.

[3]. E.F. Codd. " Extending the database relational model to capture more meaning". ACM Transactions of Database Systems, Volume 4, Number 4, Pages: 397-434, December 1979.

[4]. E. Wong." A statistical approach to incomplete information in database systems ". ACM Trans. on Database Systems, Volume 7, Pages 470-488, 1982.

[5]. J. Biskup. " A foundation of codd's relational operations". ACM Trans. Database Syst. Volume 8, Number 4, Pages: 608-636, Dec. 1983.

[6]. K.C. Liu and R. Sunderraman. "Indefinite and may be information in relational databases". ACM Transaction on Database Systems, Volume 15, Number 1: Pages 1-39, 1990.

[7]. K.C. Liu and R. Sunderraman. "A generalized relational model for indefinite and may be information". IEEE Transaction on Knowledge and Data Engineering, Volume 3, Number 1 Pages: :65-77, 1991.

[8]. L. A. Zadeh." Fuzzy sets as the basis for a theory of possibility". Fuzzy Sets and Systems, Volume 1, Pages: 1-27, 1978.

[9]. R. Cavallo and M. Pottarelli. "The theory of probabilistic databases" in Proceedings of the 13th Very Large Database Conference, pages 71 -81, 1987.

[10]. S. Parsons. "Current approaches to handling imperfect information in data and knowledge bases". IEEE Trans. Knowledge and Data Engineering, Volume 3, Pages: 353 -372, 1996.

[11]. Smarandache, F. (1999). **"**A Unifying Field in Logics. Neutrosophy: Neutrosophic Probability, Set and Logic**".** Rehoboth: American Research Press.

[12]. W. Lipski. "On semantic issues connected with incomplete information databases". ACM Trans. Database Syst. Volume 4, Number 3, Pages: 262-296, Sept. 1979.

[13]. W. Lipski. " On databases with incomplete information". Journal of the Association for Computing Machinery, Volume 28, Issue 1, Pages 41 -70, 1981.