Design Model of Application Measurement Imperfect Information to Procesing Data Surveys Level of Website Learning With Fuzzy Query Basis Data Method

Sri Ipnuwati
Computer and Informatics Management College
STMIK Pringsewu, Lampung Indonesia
nengachie@gmail.com

M. Islam Mahdi
Computer and Informatics Management College
STMIK Pringsewu, Lampung Indonesia
omiisskom@gmail.com

***Abstract*— Mastery of information technology applied in the design of information systems in the form of the web at this time becomes an absolute necessity in implementing business processes of an institution and organization. The level of students 'ability in information systems in web design is a goal to increase students' competitive value in global trading climate. In an effort to increase the mastery of students in designing a web needs to measure the level of mastery, so that the material evaluation of lecturers in the process of teaching and learning activities, especially web courses. Method Fuzzy Query Database is one method to measure the level of imperfect data and information precision (Imperfect Information). In the process of survey level mastery of programming materials and web design data collected not only the exact data but it can be data that contains doubt, imperfection and uncertainty so that in the process of decision-making occurs imperfect information so ineffective and accurate. This research is expected to assist computer lecturers in evaluating the achievement of learning, lecture materials and teaching techniques in the lecture hall.**

Keywords—component; formatting; style; styling; insert (key words)

I. INTRODUCTION

At this time private universities and government colleges have implemented and utilize information management system of universities based on information technology to be able to improve competitive advantage to service quality and efforts to foster and develop harmonious relationship between student and management of college (Student Relationship Management). Thus, it is expected that strong value chains will exist between them; the application of relational database model has proven to be the right model and has been applied in various application of the information system of universities.

Measuring the level of ability and skills of students in designing the web for E-Commerce, E-Government and E-learning and web-based college information system needs to be made study model analysis system that can assist lecturers in making learning strategies. Merging the concept of relational database technology and fuzzy set concept produces a concept called Fuzzy Query Database. In a conventional database, all data generated is data that satisfies the query correctly. Fuzzy Query Database allows the data generated is data with the level of accuracy (satisfaction Degree) in accordance with the desired value. This data base model can accommodate imperfect information (Imperfect Information).

By using Fuzzy Query Database Query method, this research will create database model that can be used to accommodate imperfect information and data (Imperfect Information) to improve accuracy in evaluating the level of mastery of web course material by students majoring in information system

II. LITERATURE REVIEW

A. *Fuzzification*

Fuzzification is the conversion of all input / output variables to a fuzzy set of forms. The range of input variable values is grouped into several fuzzy sets and each set has a certain degree of membership. The fuzzification form used in this system is the shape of triangle and shoulder shape. The fuzzification shape determines the degree of membership of an input / output range value. The degree of fuzzy set membership is calculated by using the membership function formula of the fuzzified triangle maintaining the Integrity of the Specifications.

In college service business applications, information is often uncertain (uncertain) and unclear (vague). Several ways have been taken to classify the various possibilities of imperfect information. Inconsistency, imprecision, uncertainty, and ambiguity are the basic causes of imperfect information in database and information systems.[1]

1) Inconsistency

   Inconsistency is a semantic conflict when an aspect in the real world is represented more than once with different meanings in one or more data bases. For example, a researcher has "student" and "alumni" grades for university status in a database. Information inconsistency usually occurs from the integration process.

2) Imprecision

The imprecision is related to the content of the value of an attribute. This means that an option must be made from an interval or set but it is not known exactly which to choose. For example, a worker has a value of "between 30 to 40 years" for an age attribute.

3) Vagueness

The intuitive vagueness is the same as the inaccuracy because it is both related to the content of the value of an attribute. An example of uncertainty is that a worker has a "middle-aged" value for the age attribute. This obscure information is represented by a language term.

4) Uncertainty

Uncertainty (uncertainty) is related to the degree of truth of the value of an attribute. For example the phrase "Harris 90 percent believes that Ikhlas has graduated from college" represents uncertain information

5) Ambiguity

Ambiguity means that information can have more than one interpretation. For example the phrase "Afif income of three hundred thousand dollars" is an ambiguous sentence because the income can be every week or every month

B. Membership Function

Membership function is a curve showing the mapping of data input points into their membership values (also called membership degrees) that have intervals between 0 and 1. One way that can be used to obtain membership value is through a function approach. The degree of membership in the set (degree of membership) is denoted by μ. In the fuzzy system many are known to various membership functions (membership function).[2]

III. METHODE

The concept of fuzziness as described by Zadeh includes imprecision, uncertainty, and degrees of truthfulness of values. Zadeh introduced a theory whose objects fuzzy sets are sets with boundaries that are not precise and the membership in this fuzzy set is not a matter of true or false, but rather a matter of degree. This concept was called Fuzziness and the theory was called Fuzzy Set Theory[3]. It is particularly frequent in all areas in which human judgment, evaluation and decisions are important.

When A is a fuzzy set and x is a relevant object then the proposition “x is a member of A” is not necessarily either true or false [2], as required by the two-valued Boolean logic, but it may be true to some degrees, and the degree to which x is actually a member of A, is a real number in the interval [0,1].Theoretically

$$F= \{x, \mu F(x))| x \epsilon X\}$$

F is the Fuzzy Set where X is a collection of objects denoted by x μF(x) is called the membership function(or grade of membership) of x in F. the range of the membership function is a subset of the non-negative real numbers in the interval [0,1].

Classification of Fuzzy Data can be further divided into two types:

a) **Approximate Value**: The information data is not totally vague and there is some approximate value , which is known and the data lies near that value

b) **Linguistic Variable**: This variable is used to represent a fuzzy number. A linguistic term is a name given to the fuzzy set.[4]

A. An Approach to Fuzzy Database Querying, Analysis and Realisation

Users search databases in order to obtain data needed for analysis, decision making or to satisfy their curiosity. The SQL is a standard query language for relational databases. The simply SQL query is as follows:

Select attribute_1,…,attribute_n

from T

where attribute_p > P and attribute_r < R

The result of the query is shown in graphical mode in figure 1. Values P and R delimit the space of interesting data. Small squares in the graph show database records. In the graph it is obviously shown that three records are very close to meet the query criterion. These records could be potential student and lecturer could attract them or municipalities which almost meet the criterion for some web mastery capability support for example.

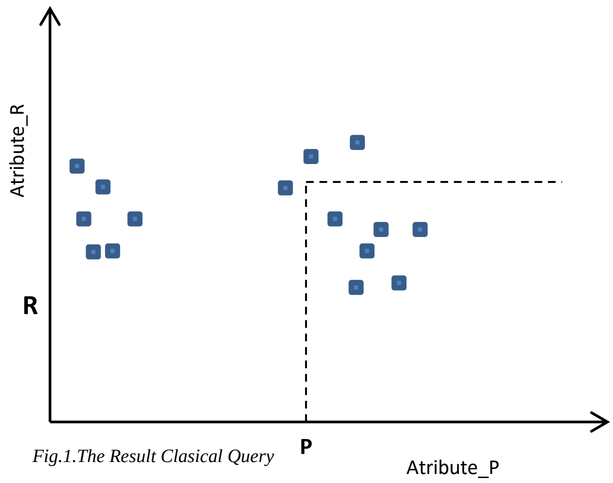

Fig.1.The Result Clasical Query

The SQL uses the crisp logic in querying process that causes crisp selection. It means that the record would have not been selected even if it is extremely close to the intent of the query criterion. As the criterion becomes more and more complex, the set of records selected by the WHERE statement becomes more and more crisp. If the classical SQL is used for solving this problem, the SQL relaxation would have to be used in the following way:

select attribute_1,…,attribute_n
from T
where ateribute_p > P-p and attribute_r < R+r

Where p and r are used to expand the initial query criteria to select records that almost meet the query criteria. This approach has two disadvantages [3].First, the meaning of the initial query is diluted in order to capture adjacent records. The meaning of a query: “where attribute p is more than P” is changed and adjacent records satisfy a query in the same way as initial ones. More precisely, the difference between original and adjacent data (caught records along the “edge” of interesting space) does not exist. Secondly problem arises from the question: what about records that are very close to satisfy the new expanded query and it is useful to make another expanding of a query. In this way more data from the database is selected, but the user has lost the accuracy of his query.

Many applications have created uncountable accesses to wide variety of data. The data and the classical access to data are simply not enough in many cases. In cases when the user cannot unambiguously separate interesting data from not interesting by sharp boundaries or when the user wants to obtain data that is very close to meet the query criterion and to know the index of distance to full query satisfaction, it is necessary to adapt the SQL to these requirements. The SQL was initially developed in [7]. Since then the SQL has been used in many relational databases and information systems for data selection. The use of SQL may be regarded as one of the mayor reasons for the success of relational databases in the commercial world [13]. In this research the core of SQL remains intact and the extension is done to improve the selection process. Adding some flexibility to the SQL meets above mentioned requirements and increases effectiveness and comprehensibility of the whole querying process.

B. Queries based on fuzzy logic

The generalized logical condition (GLC) for the WHERE part of the SQL based on linguistic expressions has been created in [5]. For the further reading it is important to define the Query Compatibility Index (QCI). The QCI is used to indicate how the selected record satisfies a query criterion. The QCI has values from the [0, 1] interval with the following meaning: 0 - record does not satisfy a query, 1 - record fully satisfies the query, interval (0, 1) - record partially satisfies a query with the distance to the full query satisfaction. The GLC has the following structure :

$$where\ \theta_{i=1}^{n}(a_{i\emptyset\ L_{xi})}$$

Where n denotes the number of attributes with fuzzy constraints in a WHERE
Clause of a query,

$$\theta = \begin{cases} and \\ or \end{cases}$$

where *and* and *or* are fuzzy logical operators,
and

$$a_i \begin{cases} a_{i>Ld_i} & a_{i\ is\ beginer} \\ a_{i<}Lg_i \quad a_i & is\ intermediate \\ a_i > L_{di} \quad and\ a_i > L_{gi,} & a_{i\ \ is\ advance} \end{cases}$$

where ai is a database attribute, Ldi is the lower bound and Lgi is upper bound of a linguistic expression described by fuzzy sets shown in figure 2. Conditions in queries contain these comparison operators: >, <, =, ≠ and between when numerical attributes are used. These crisp logical comparison operators are adapted for fuzzy queries in the following way: operator > (greater than) was improved with fuzzy set „Big value“ (figure 2a), operator < (less than) was improved with fuzzy set „Small value“ (figure 2b) and operator = (equal) was improved with fuzzy set „About value“ (figure 2c). Operator ≠ is the negation of the operator = so this operator is not further analysed. Analogous statement is valid for the operator *between* because it is similar to the operator = from the fuzzy point of view. Other types of fuzzy sets could be added in the future to catch other linguistic expressions.

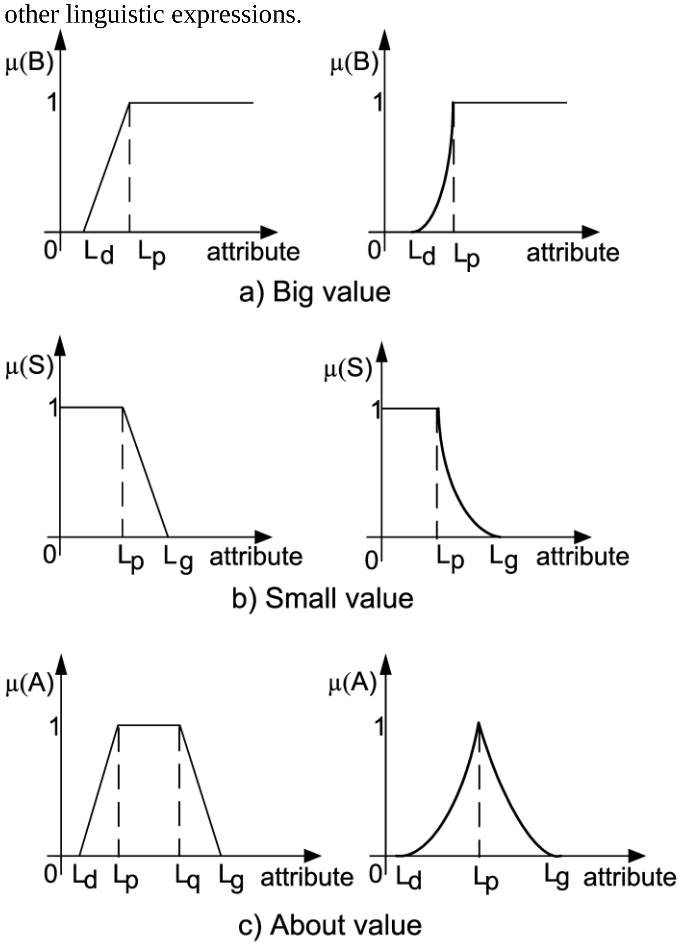

Fig. 2. Fuzzy sets

In the figure 2 two often used fuzzy sets are shown and implemented in the fuzzy query approach. It is possible to add more types of fuzzy sets for each of linguistic expressions. A useful theoretical overview about fuzzy sets and fuzzy operators can be found in [9] and a useful practical overview can befound in [3].

If a query in the WHERE clause contains fuzzy as well as classical constraints, these classical constraints could be easy added to the WHERE clause (3) in the following way:

where〖 θ〗_(i=1)^n (a_(i∅ L_xi)) [and/or][attribute_m LIKE “String”] [and/or] where LIKE is the SQL comparison operator and string is an arbitrary string variable.

IV. System Development

The development of fuzzy query system in the design stage for the application of the student's level of ability measurement in the learning process requires a dataset containing data imperfections. Fuzzy Database System consists of three components, namely extended database, fuzzy interpreter, and interface. The design of Fuzzy Database System can be seen in Figure 3.1.

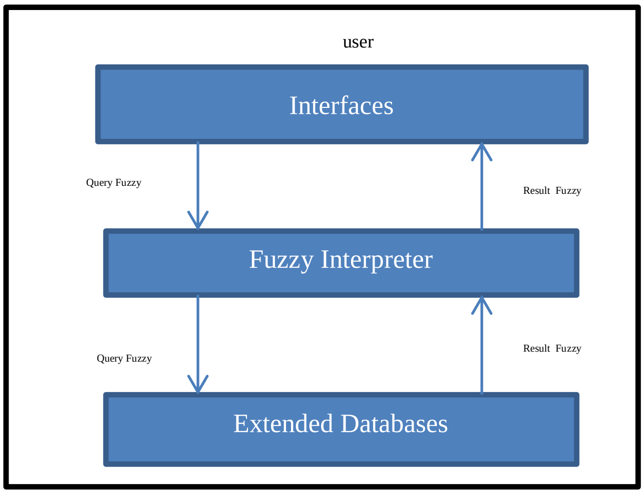

Figure 3.1 Fuzzy Database System Design

A. *Extended Database Design*

The main objective of this research is to implement fuzzy union and fuzzy interaction operations that implemented in the application model of student's mastery of the lecture material that has followed the web programming lecture. These operations are binary operations that run on two sets. Therefore, it takes at least two tables that will result in two different tables that are the result of running the query. The result table can be viewed as a set of tuples that have a satisfaction degree value on each line.

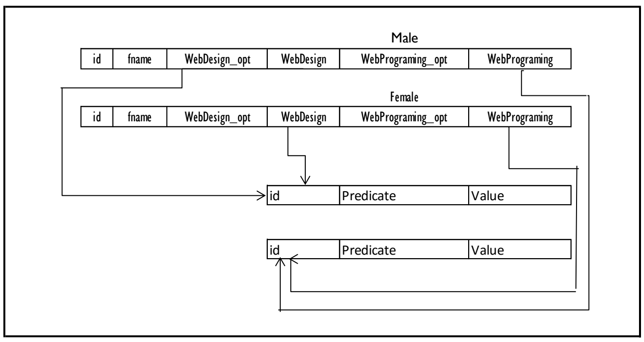

Figure 3.2 Extended Database Design

There are four tables created, namely Male, Female, WebDesign, and WebPrograming. Male and Female tables are the main tables in this extended database. Both tables have the same attributes, namely id, fname, WebDesign_opt, WebDesign, WebPrograming_opt, and WebPrograming. The id attribute is the primary key. The fname attribute is used to store the name. Attributes of age and income are solely used to store age and income. Both of these attributes can be crisp or fuzzy. Therefore, two new tables are created, namely the Age and Income tables, to accommodate the fuzzy values of both attributes. Value type, crisp or fuzzy, in the age and income attributes are sequentially stored in the WebDesign_opt and WebPrograming_opt attributes.

The WebDesign and WebPrograming tables have the same attributes, id, predicate, and value. The id attribute is the primary key of each table. The predicate attribute is used to store linguistic terms in the membership function. For example, the terms Beginner WebPrograming, Intermediate WebPrograming, and Advanced WebPrograming in the WebPrograming table. The value attribute is used to store the values that are the parameters of the membership function. In the extended database design, the WebDesign table with WebPrograming has the same attributes.

B. *Fuzzy Interpreter Module Design*

The conventional database used can not accommodate fuzzy queries. Therefore, a module must be created that transforms the user's fuzzy query to the crisp query that can be processed by the database used. This module is called a fuzzy interpreter. This module will also transform the crisp information from the database to the fuzzy information that will be provided to the user. The core of the fuzzy interpreter module is nine main procedures, namely a satisfaction degree fitting procedure, four fuzzy union operation procedures, and four fuzzy intersection operation procedures. The definitions of the satisfaction degree and fuzzy union and fuzzy intersection operations have been given in the previous chapter. Implementation of these nine modules will be explained in the system implementation section

C. *Trial Fuzzy*

In the trial, the author perform the following steps:

1. *Running Operation Join*
   In this step the author performs two equi-join operations on the Male and Female table. In the first join operation, the attribute used as the join parameter is WebDesign (join on WebDesign) while in the second operation attribute used is WebPrograming (join on WebPrograming). The result of this step are two new tables. The first table is a combination of rows in the Male and Female tables that have the same WebDesign with the satisfaction degree greater than zero while the second table is that have the same income with satisfaction degree more than zero.
2. *Running Fuzzy Union Operations*
   In this step the authors run fuzzy union operations with two table parameters generated in the first stage, the join-WebDesign table and the join-on-WebPrograming table. The fuzzy union operations performed are standard union, algebraic sum, bounded sum, and drastic sum. The results of this stage are four new tables. Each row in each table has a satisfaction degree which is the result calculation of fuzzy

union operators with satisfaction degree parameters of two input tables. Only rows that have a satisfaction degree more than zero will be part of the results table.

3. *Running Fuzzy Intersection Operations*
   In this step the authors run fuzzy intersection operations with two table parameters generated in the first stage, ie join-WebDesign table and join-on-WebPrograming table. Intersection fuzzy operations that run are standard intersection, algebraic product, bounded product, and drastic product.
   Just as in stage two, the results of this stage are four new tables. Each row in each table has a satisfaction degree which is the result of fuzzy intersection operators with satisfaction degree parameters of two input tables. Only rows that have a satisfaction degree more than zero will be part of the results table.

D. Database

The data used in the trial are two tables, namely Male table and Female table. Each table consists of ten lines. The WebDesign and WebPrograming attribute values can be crisp or fuzzy values. Table 4.1 show the data used in this trial

	Male			Female		
id	fname	WebDesign	WebPrograming	fname	WebDesign	WebPrograming
1	Amir	Almost proficient	Intermediate	Mintarsih	Almost proficient	Beginner
2	Adi	Beginer	General average	Vio	Beginer	Intermediate
3	Agus	Intermediate	Intermediate	Romlah	Intermediate	advance
4	Hendro	General average	General average	Neni	General average	Beginner
5	Bambang	Beginner	Intermediate	Sri	Beginner	Intermediate
6	Dadi	Intermediate	General average	Erna	Intermediate	Beginner
7	Zaky	advance	Beginner	Irma	General average	Beginer
8	Asep	Beginner	Intermediate	Suci	Intermediate	Intermediate
9	Dadang	advance	advance	Siti	advance	General average
10	Nanang	Beginner	Beginer	Mimin	Beginner	Beginner

E. Implementation of Procedures in Fuzzy Interpreter Module

In this section will be presented procedures that are implemented on fuzzy interpreter module Fuzzy Database System.

1). Procedure Determination of Satisfaction Degree

The procedure of determining the satisfaction degree is used to determine how much accuracy a row satisfies a given query. The parameters of the given query are WebDesign or WebPrograming that can be crisp or fuzzy. The age or income value on the line to be compared with the query can also be crisp or fuzzy.

```
getSatisfactionDegree(jenisBaris,jenisQuery,nilaiBaris,nilaiQuery)
{
        if (jenisBaris dan jenisQuery sama-sama crisp)
{
        satisfactionDegree = 1 jika nilaiBaris = nilaiQuery;
        satisfactionDegree = 0 jika tidak;
}
    else if (jenisBaris dan jenisQuery sama-sama fuzzy)
{
satisfactionDegree = 1 jika nilaiBaris = nilaiQuery;
satisfactionDegree = titik potong dengan nilai terbesar antara
kedua fungsi fuzzy;
}
else // jika memiliki jenis nilai yang berbeda
{
satisfactionDegree = pemetaan nilai crisp ke fungsi fuzzy
}
return satisfactionDegree;
}
```

2). Procedure of Standard Union

The standard union procedure is used to determine the satisfaction degree value of a row in the merged table of two tables based on the definition of standard union operation. The following will explain the standard union procedure.

```
getSatDegreeStandardUnion(satDegree1, satDegree2)
 {
        satDegreeStdUnion = maksimum(satDegree1, satDegree2);
    return satDegreeStdUnion;
}
```

3). Procedure of Algebraic Sum

The algebraic sum procedure is used to determine the satisfaction degree value of a row in the merged table of two tables based on the definition of algebraic sum operation. The following will explain the algebraic sum procedure.

```
getSatDegreeAlgSum(satDegree1, satDegree2)
{
      satDegreeAlgSum = satDegree1+satDegree2 - satDegree1*satDegree2;
    return satDegreeAlgSum;
}
```

4. Procedure of Bounded Sum

The bounded sum procedure is used to determine the satisfaction degree value of a row in the merged table of two tables based on the definition of a bounded sum operation. The following will explain the bounded sum procedure.

```
getSatDegreeBnddSum(satDegree1, satDegree2)
{
        satDegreeBnddSum = minimum(1, satDegree1+satDegree2);
    return BnddSum;
}
```

5. Procedure of Drastic Sum

```
getSatDegreeDrstcSum(satDegree1, satDegree2)
{
satDegreeDrstcSum = maksimum(satDegree1, satDegree2);
if(satDegree1 = 0)
{
satDegreeDrstcSum = satDegree2;
}
else if(satDegree2 = 0)
{
satDegreeDrstcSum = satDegree1;
}
else // tidak ada yang bernilai 0
{
satDegreeDrstcSum = 1;
}
return satDegreeDrstcSum;
}
```

6. Procedure of Intersection

```
getSatDegreeStdIntersection(satDegree1, satDegree2)
{
satDegreeStdIntersection = minimum(satDegree1, satDegree2);
return satDegreeStdIntersection;
}
```

7.Procedure of Algebraic Product

```
getSatDegreeAlgProd(satDegree1, satDegree2)
{
    satDegreeAlgProd = satDegree1*satDegree2;
   return satDegreeAlgProd;
}
```

8.Procedure of Bounded Product

```
getSatDegreeBnddProd(satDegree1, satDegree2)
{
  satDegreeBnddProd = maksimum(0, satDegree1+satDegree2 - 1);
return BnddProd;
}
```

9.Procedure of Drastic Product

```
getSatDegreeDrstcProd(satDegree1, satDegree2)
{
if(satDegree1 = 1)
{
satDegreeDrstcProd = satDegree2;
}
else if(satDegree2 = 1)
{
satDegreeDrstcProd = satDegree1;
}
else // tidak ada yang bernilai 1
{
satDegreeDrstcProd = 0;
}
return satDegreeDrstcProd;
}
```

V. Conclusion

Fuzzy Database System can accommodate imperfect information. This research data uses vague data, such as fuzzy values on WebDesign and WebPrograming attributes, which are common in real-world cases. Conventional database systems can not handle this because it can only accommodate precise values. Implementation of fuzzy union and intersection operations can be done on the Fuzzy Database System. These are important operations because the use of these operations allows for the combination of one or more tables that are often encountered in real-world applications using a conventional database.

Acknowledgment

This research is supported by Computer and Informatics Management College (Sekolah Tinggi Manajemen Informatika & Komputer) Pringsewu, Lampung and DIKTI Research Grants for Beginner Lecturers (PDP) Financing in 2016.

References

[1] Z. Ali and V. Singh, “POTENTIALS OF FUZZY LOGIC : AN APPROACH TO HANDLE IMPRECISE DATA,” vol. 2, no. 4, pp. 358–361, 2010.

[2] J. B. Simha and S. S. Iyengar, “Fuzzy data mining for customer loyalty analysis,” *9th Int. Conf. Inf. Technol.*, pp. 3–4, 2006.

[3] M. Lesani and N. Montazeri, “Fuzzy trust aggregation and personalized trust inference in virtual social networks,” *Comput. Intell.*, vol. 25, no. 2, pp. 51–83, 2009.

[4] A. Meier and A. Tiefenaustr, “Using a Fuzzy Classification Query Language for Customer Relationship Management,” *Context*, pp. 1089–1096, 2005.

[5] Bosc, P, and H Prade. "An Introduction of Fuzzy Set and Possibility Theory Based Approach to The Treatment of Uncertainty and Imprecision in Database Management System." *Proceedings of the Second Workshop on Uncertainty Management in Information Systems: From Needs to Solutions.* 1993.

[6] Calva, Diana, and Mario Lehman. "An Analysis of the Possible Applications of Artificial Intelligence Techniques to a Clinical Laboratory Information Management Systems." *IJCSNS International Journal of Computer Science and Network Security*, 2008: 82-86.

[7] H.Lee, Kwang. *First Course of Fuzzy Theory.* Berlin: Springer, 2005.

[8] Hsieh, Nan-Chen, Ding-An Chiang, and Tsong-Sheng Wang. "Answer to Database Queries Concerning Imprecise Information in Logical Fuzzy Relational Database." *Tamkang Journal of Science and Engineering*, 2004: 149-160.

[9] Meier, Andreas, Nicolas Werro, Martin Albrecht, and Miltiadis Sarakinos. "Using a Fuzzy Classification Query Language for Customer Relationship Management." *31st VLDB Conference.* Trondheim, 2005.

[10] Yang, Qi, et al. "Efficient Processing of Nested Fuzzy SQL Queries in a Fuzzy Database." *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 2001: 884-901.