FUZZY LOGIC, ESTIMATED NULL VALUES AND THEIR APPLICATION IN RELATIONAL DATABASES

by

SUSAN E. POWELL

B.S., Kansas State University, 1980

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1986

Approved by:

Major Professor

TABLE OF CONTENTS

TABLE OF CONTENTS
(Cont.)

LIST OF FIGURES

LIST OF FIGURES
(Cont.)

CHAPTER 1: INTRODUCTION

The information in a database represents or codifies the current knowledge of its users. Since knowledge may be incomplete the database must provide a means of representing values collectively not known by the user. These unknown values vary in type from data that is not currently available but will be in the near future, such as monthly sales figures, to data that is missing and can not be provided, such as a telephone number for a client with an unlisted number. In relational databases the null value, commonly represented by the symbol @, provides the means for users to store these kind of values. The null value is generally categorized as a value which is unknown, incomplete, or inconsistent.

The inclusion of null values in a relational database leads to the problem of processing those values correctly. Queries made by the user must provide accurate, hopefully not misleading, if incomplete, results. In a relational database this means defining relational operators so that semantic correctness is preserved.

Unfortunately, most research on the subject of null values has attempted to solve or lessen the problem of correctness by severely limiting the types

of null values allowed in the database. For example, Imielinski and Lipski (ImiLip84) define a null value as 'at present unknown (but the attribute applicable)'. Vassiliou (Vas80) limits the definition of nulls to those values which are 'missing' or 'nothing'. Buckles and Petry (BucPet82) note that researchers believe while data could be missing (incomplete), the existing data was exact. Boridga (Bor85) also mentions limitations placed on null value types. He states, "The different ways in which knowledge is allowed to be incomplete is usually very limited: 'value exists but is unknown', 'no value possible', or 'total lack of knowledge'." By restricting the types of null values allowed knowledge that can be derived from incomplete data is very limited. The null values essentially become placeholders in the database, contributing no additional useful information of their own.

One important type of incomplete data left out of null values is data which can be estimated. Borgida (Bor85) calls those values which can be estimated 'informative nulls' and notes their usefulness by stating, "These values (informative nulls) are known not to be accurate reflections of reality (they may be out of date or just projections) but are certainly not equivalent to a total lack of knowledge." For example

the projected employee turnover of a company estimated by an experienced personnel manager is far more informative than no value at all. Furthermore, if the manager is extremely accurate the estimate can become as valuable to the user as the unknown data it represents.

Unfortunately Borgida does not attempt to solve the problem of processing estimates in a relational database. He only stores the estimates in the database and notes their existence to the user. The problem addressed in this work is to increase the information provided to users by broadening the acceptable null values in the database to include estimates and to create ways of maintaining and querying databases containing these null values such that the results are logically consistent with reality. Chapter 2 addresses the difficulty of processing estimates and suggests a solution to a constrained problem of processing estimated data by using fuzzy logic. In Chapter 3 the method of including estimates in the database is presented and the database representation of estimates is defined. Chapter 4 outlines the querying of databases containing estimates using fuzzy set logic. Chapter 5 discusses the implementation of this solution to processing estimated data.

CHAPTER 2: EXISTING APPROACHES

Several different approaches have been taken to solve the problem of processing null values in relational databases. These approaches include predicate (first-order) logic, relational algebras, knowledge representation, and fuzzy logic.

PREDICATE LOGIC:

In predicate logic expressions correspond to mathematical statements. N-valued logics are used in predicate logic to allow the expression of nulls. Biskup (Bis83) introduces a three-valued logic where the third value is provided by extending the relational table with a two-valued tag field. The tag field is called the "Status" of the value and may be either definite or maybe, with the maybe used to represent null values. Vassiliou (Vas79) also used n-valued logic (modal logic) to represent nulls. His four-valued logic system allows nulls to have the values of either "missing" or "nothing".

RELATIONAL ALGEBRAS:

Relational algebras use range domains, sets, and logical quantifiers to represent nulls. Codd's (Cod75) approach was to extend predicate (first-order) logic to three-valued logic, therefore, extending the

operators of the relational algebra to include missing values. Grant (Gra77) and others have criticized Codd's approach on semantic grounds and Grant developed a method of representing partial values as ranges or sets of possible values. In Grant's approach, null values may be properly replaced with an actual range which is defined for the given domain by integrity constraints. He introduces three notions of operations which might be applied to his representation -- set theoretic, true, and maybe versions. For example, his true intersection operation omits all non-single entries while the maybe intersection selects all entries for which the ranges overlap even slightly. Figure 2.1 below gives a relation CLIENT with value ranges allowed in both the age and premium attributes.

CLIENT

#	AGE	PREMIUM
225	30	250
113	(40,45)	null
740	55	(105,125)
310	60	140

Figure 2.1: Sample Relation - CLIENT

If the Query is: Select Client where AGE) 35 and Premium) 100 the selections shown in Figure 2.2 on the following page would occur for true and maybe operations.

TRUE SELECTION		
310	60	140

MAYBE SELECTION		
113	(40,45)	null
740	55	(105,125)
310	60	140

Figure 2.2: True and Maybe Selection Operations

KNOWLEDGE REPRESENTATION:

Knowledge representation evolves from the area of artificial intelligence. The emphasis in this approach is to use human reasoning and knowledge bases with proper constraints to maintain semantics. Lipski (Lip79) defines the semantics of a query both internally and externally. The internal semantics is determined based on the information in the database. The external semantics is based on the real world modeled by the database and is not limited by incomplete information in the database. For example if the query is (Height < 5'5") the external interpretation is the set of persons who are in reality of height less than 5'5", whereas the internal interpretation consists of persons known (in the system) to be under 5'5". Alternatively, the internal interpretation may be altered to include unknown values by responding with all persons possibly under 5'5". The internal interpretation can ONLY approximate the external property (Height < 5'5").

FUZZY LOGIC:

Fuzzy logic, based on fuzzy set theory, is a more recently developed approach to processing null values. In fuzzy logic nulls are represented by their degree of membership in a fuzzy set. Buckles and Petry (BucPet82) present a structure in which the non-fuzzy database is a special case of the fuzzy database. Components of tuples can be multivalued and a similarity relation is required for each domain set of the database. A sample similarity relation for playing instruments would appear as shown in figure 2.3 below.

	piano	guitar	banjo	accordion
piano	1	0	0	.7
guitar	0	1	.6	0
banjo	0	.6	1	0
accordion	.7	0	0	1

Figure 2.3: Similarity Relation

The relation of the musicians and the instruments they might play is given in figure 2.4 below.

Name	Instrument
Jim	piano, accordion
Sally	banjo, guitar
Alan	piano, accordian

Figure 2.4: Result of Similarity Relation

Note: The attribute 'Instrument' can have multiple

values. Zadeh (Zad83) uses fuzzy logic to include both possibilistic and probabilistic theory in a single system. He applies the use of fuzzy logic specifically to solving the problem of inference in expert systems. The use of fuzzy logic reduces the problem of inference to that of solving a nonlinear program and leads to conclusions whose uncertainty is a cumulation of the uncertainties in the premises from which the conclusions are derived. The result is that the conclusions are fuzzy sets which are characterized by their possibility distributions. For example, the fuzzy set THIN is shown in figure 2.5 below.

WEIGHT	DEGREE OF MEMBERSHIP
110	1.00
125	.95
140	.8
155	.5
180	.45
195	.3
210	.2
225	0.00

Figure 2.5: Fuzzy Set - THIN

The degree of membership given for each weight in the fuzzy set THIN represents the possibility that a given weight-X is a member of the fuzzy set THIN. That is, if X = 125 then the possibility that 125 lbs. may have the value THIN is 95%.

LIMITATIONS OF PREDICATE LOGIC AND PROBABILITY THEORY:

When examining these solutions to query processing of estimates the first three approaches have two main limitations. They all are based on or strongly associated with predicate logic and probability theory. Predicate logic can be either two-valued, multivalued or n-valued. In two-valued predicate logic a proposition is either true or false. N-valued predicate logic allows a proposition to be true or have an intermediate truth-value which is an element of a finite or infinite truth-value set. Rescher (Res69) noted that the first step beyond two-valued logic into a third valued logic came from J. Lukasiewicz in 1920. Lukasiewicz introduced the idea of a third, "intermediate" or "neutral" or "indeterminate" value. A modern adaption of the example he provided for multiple valued logic or modality could be written as follows:

> I can assume without contradiction that - My whereabouts at this time next year are at present unknown. Based on this assumption the proposition "I shall be in New York at noon on June 9, 1987", can at the present be neither true or false. For if it were true now, my future presence in New York would have to be necessary, which is contradictory to the assumption. If it were false now, on the other hand, my future presence in New York would have to be impossible, which is also contradictory to the assumption. Therefore the proposition considered is at the moment neither true nor false and must possess a third value, different from '0' or falsity and

'1' or truth. The third value represents the possible and is represented by '1/2'.

Multi-valued logic continues along this line of reasoning to provide more than one possible value between the given truth or falsity of a proposition.

The theory of probability is viewed in the classical sense as reducing all events which can occur in a given circumstance to a certain number of equally possible cases, that is we are equally undecided about their existence. The number of those cases that are favorable to a given event is determined. The ratio of this number to all possible cases is the probability. (Nid60) A common example used to demonstrate probability is the throwing of a die. The events which can occur, are the equally possible cases of the numbers: 1, 2, 3, 4, 5, 6. If the given event is throwing a six, the number of cases favorable to that event is 1 because only one six appears on the die. The ratio of the number of cases (1) to the total number of possible cases (6) is 1/6. (i.e., the probability of throwing a six is 1/6)

Predicate logic and probability theory are limiting for estimates because they assume all information is known and provide no mechanism to handle uncertainty. Zadeh (Zad83) notes this limitation can result in poor or questionable results.

"In the existing expert systems, the fuzziness of knowledge is ignored because neither predicate logic nor probability-based methods provide a systematic basis for dealing with it. As a consequence, fuzzy facts and rules are generally manipulated as if they were nonfuzzy, leading to conclusions whose validity is open to question." For example, if the following statement is given as a fact: Stan will win the election (CF = 0.6). The CF (conditional factor) of 0.6 could be either Stan will win by 60% or that the probability of Stan winning is 60%. Probability theory allows the certainty factor to be ambiguous. Glas (Gla83) also states the limitations of probability theory when he states, "The probabilistic approach cannot represent uncertainties attached to systems where some deterministic dynamical characteristics are unknown or deliberately ignored as well as uncertainties attached to their mathematical model."

COMPARISON OF FUZZY LOGIC AND PREDICATE LOGIC:

The limitations of systems based on predicate logic and probability theory become more obvious when compared to the alternative of fuzzy logic. In fuzzy logic a proposition can have a truth-value over a range of fuzzy subsets. The relationship of a fuzzy

set to an ordinary set can best be shown by recalling the definition of the characteristic function of a set. With two-valued predicate logic a set has the form shown in figure 2.6 below.

$$U \rightarrow \{0,1\}$$

Figure 2.6: Predicate Logic Set Function

This set maps the universe U to a set of two elements. There is a binary choice between being in or out of the set. (With n-valued predicate logic the set may be extended to have a third, fourth or nth value.) However with fuzzy logic the set function would be defined with values in a unit interval as shown in figure 2.7 below.

$$U \rightarrow \{0,\ldots,1\}$$

Figure 2.7: Fuzzy Logic Set Function

This function allows an infinite number of possible choices. By using this function fuzzy quantifiers (old, frequently, almost 0.8, few) can become meaningful. For example, the term 'tall' can be defined in the range of human heights. If a person is of above average height, 6'2", the degree of membership of that height in the set would be 0.8. Thus more than whether that person is 6'2" or not can be determined. The term 'tall' can be defined mathematically. Fuzzy

logic also extends the framework of the system to deal with both probabilistic and possibilistic theory. In possibilistic theory the possibility value of a variable is determined by the 'degree of easiness' with which the elements of a given universe of discourse might be assigned to the variable. (BorKru83) An example of the difference between probability and possibility theory is best demonstrated by comparing a probability and possibility distribution. Consider the statement: Karen sold X units this month. The probability distribution would interpret X as a random variable and compute Px(u) (the probability of X over the values in U where U = {1,2,3,...} as the probability of Karen selling u units in an arbitrary month. The possibility distribution would compute Posx(u) as the 'degree of ease' with which Karen could sell u units in an arbitrary month. Figure 2.8 provides a possible set of distribution results.

u	1	2	3	4	5	6	7
Px(u)	0	0	0.1	0.6	0.3	0	0
Posx(u)	0.2	0.3	0.7	1	1	0.4	0.1

Figure 2.8: Probability and possibility Distributions of X.

The probability distribution must sum to a total probability of 1, whereas the possibility can

represent a much wider range of values in the domain. This difference provides possibility theory with a flexibility to handle uncertain data not present in probability theory. Borisov and Krumberg (BorKru83) demonstrate the importance of this flexibility when they state, "In many situations the decision maker has less information than required to use probability theory. There are cases in which one can speak in terms of possibilities but no concept of probability exists."

Based on these comparisons of predicate logic and probability theory to fuzzy logic the obvious solution to query processing of estimates is provided in fuzzy reasoning and possibility theory. Zadeh (Zad83) clearly states the usefulness of fuzzy logic when processing uncertain data. "Fuzzy logic provides a natural framework for the management of uncertainty in expert systems because - in contrast to traditional logical systems - its main purpose is to provide a systematic basis for representing and inferring from imprecise rather than precise knowledge." Zadeh (Zad83) goes on to state that fuzzy logic provides the basis for possibility theory to be included in the database. As he explains, "... a fuzzy-logic-based computational framework (can) be employed to deal with possibilistic and probabilistic uncertainty within a

single conceptual system. In this system, test-score semantics - which is the meaning-representational component of fuzzy logic - forms the basis for the representation of knowledge...".

With fuzzy logic as a means for manipulating data, estimates can now be introduced into the database and processed in database queries. The next chapter will outline the representation of estimates in the database system.

CHAPTER 3: THE MODEL: REPRESENTATION OF ESTIMATED VALUES

INTRODUCTION:

Before describing in detail the form of representation of an estimated value some consideration should be given to the type of values that will be candidates for estimates in the database. Not all values in a database can be estimated nor would an estimate be desirable in some cases. For example the name of an individual would not be a likely value to estimate because there would be no basis on which to make an educated guess. Another example of a value that is not a candidate for estimating and could even disrupt normal operation of the database is a social security number (SSN). The SSN could be estimated with some accuracy based on date of birth but if the estimating resulted in two people with the same SSN major problems could occur especially if the SSN were being used as a key. For instance transactions meant for only one of the individuals would affect both of them since the SSN was the same. The types of values much more suited for estimating are sales, costs of production, time to complete a project, etc. These values have a good basis for being estimated and are probably not key values that could cause problems by being duplicates of actual or other estimated data in

the database.

Once an attribute value is selected as a candidate to allow estimates the user must be informed that the value provided is not actual data but an estimate. This can be done by attaching a second value to each estimate. The second value would represent the accuracy of the estimate. This second value serves two functions: its presence notifies the user that the information provided is an estimate and its contents tells the user of the accuracy of the estimate. This second value can be termed a 'certainty factor' as it represents the certainty of the user about the accuracy of the estimate.

The certainty factor could be represented in many forms. In fuzzy logic the fuzzy quantifiers (as they are called) are often given in terms such as most, many, few, not very many, infrequently, etc. In this research, however, the choice was made to represent the certainty factors as percentages. The certainty factors range between 0% and 100% with 100% being reserved for known values. The higher the percentage the more accurate the estimate. For example, if the sales of company XYZ to other companies were unknown the estimated sales figure might be represented in the XYZ Sales relation as shown in figure 3.1 on the following page.

XYZ SALES

SOLD TO	LOCATION	SALES
Ace	Toledo	$2,000
Burns	Dallas	$30,000
CCC	Miami	($5,000/.8)
Dow	New York	($700,000/.95)

Figure 3.1: XYZ SALES - Estimated Values

The sales values for CCC and Dow are both estimates with certainty factors of 80% and 95% respectively.

These estimates with their corresponding certainty factors could be generated by using past history, forecasting, statistical regression analysis, etc. However, these types of estimates can be no more accurate than past information and can not account for the constantly changing environment of a company and the markets around it. Therefore, the model provides a means for knowledgable individuals to enter their "educated guesses" into the database as estimates. These estimates could easily be updated to match internal changes in the company and market changes that affect the company directly.

Each estimate entered by an individual would contain both the value of the estimate and the certainty factor mentioned above. To provide a common ground for users when estimating values a set of terms has been selected to represent the accuracy or

certainty factor. These terms are provided in figure 3.2 below:

"EXTREMELY LIKELY"	(A5)
"VERY LIKELY"	(A4)
"HIGHLY LIKELY"	(A3)
"LIKELY"	(A2)
"QUESTIONABLE"	(A1)

Figure 3.2: Certainty Factor Terms

The terms can be replaced by the acronyms at the right with "EXTREMELY LIKELY" being replaced by A5 and so on for simplicity. Therefore the individual could enter an estimate as: 5,000 - "HIGHLY LIKELY" or 5,000 - A3. These terms allow different estimators to enter values in a consistent manner but are not in the form of percentages that was presented earlier in this chapter. Transforming terms into percentages is done by using the mapping of a fuzzy set. The fuzzy set - ACCURACY - with the terms and their degree of memberships is shown in figure 3.3 below.

ACCURACY

TERMS	DEGREE OF MEMBERSHIP
"EXTREMELY LIKELY" A5	.95
"VERY LIKELY" A4	.9
"HIGHLY LIKELY" A3	.8
"LIKELY" A2	.65
"QUESTIONABLE" A1	.5

Figure 3.3: Fuzzy Set - ACCURACY

The degree of membership in the set ACCURACY becomes the percentage used in the database as the accuracy of the estimate. So that if the estimate is given as (5,000 - A3) the estimate would be placed in the database as (5,000/.8). The degree of membership values would initially be selected by the best means available to the company (i.e. a consensus of users) but could be adjusted later when estimates are compared to actual values.

THE MODEL:

Each estimate must be stored in the database. The estimate must be stored with the information about what value it is estimating, what the estimate is and who is doing the estimating. A formal model designed to include the needed information about an estimate is provided in figure 3.4 on the following page. The model is given as a 6-tuple entitled ESTIMATES with the values required for each component.

ESTIMATES (Name, Estimate, Time, Value, Certainty Factor, Creator)

Where: Name is the name of the relation in which the value being estimated appears.

Estimate is the specific attribute and database tuple that is being estimated in the relation Name.

Time is the date and time of day that the estimate was entered into the database.

Value and Certainty Factor are the actual estimate value and its accuracy.

Creator is the name of the individual entering the estimate.

Figure 3.4: Formal Model - ESTIMATES

The information provided by adding components to the basic estimate name and value serves several purposes. For example, by including the Time component a user can select recent estimates or estimates made during a given period of time. The Creator component allows a user to include or omit certain individual's estimates when processing data. The Creator component also identifies the estimate so that an individual can refer back to his or her own previously made estimates.

An example set of estimates in the form of a relational table is given in figure 3.5 on the following page.

ESTIMATES

Name	Estimate	Time	Value	Certainty Factor	Creator
Ace	Sales/5	5/25/86 08:05:30	$5,000	.9	Jim
XYZ	Sales/2	4/29/86 12:57:35	$3,000	.8	Ann S.
Ace	Time/4	6/1/86 16:03:43	200	.95	Art
CCC	Cost/7	3/25/86 09:00:21	$3.50	.65	Jean

Figure 3.5: Sample Relation of Estimated Data

The attribute Estimate provides the attribute name in the relation first and then an identifier of the tuple in which the value to be estimated is located (i.e. the relation CCC has an attribute named Cost and the estimate shown is for record 7.)

In Chapter 4 the querying of estimated values and their certainty factors will be further outlined using fuzzy logic as a basis of the database.

Chapter 4: QUERIES ON ESTIMATED VALUES

An estimate could simply be provided by one user but it is often more desirable that more than one individual estimate a value in the database. For example, if a manager has several salespeople in his division of the company and wants to project the potential of a new product he/she would probably want to have each salesperson provide their estimate of the future sales of that product. Also if several teams are each developing one section of a large computer program each team leader could provide an estimate of the time it would take his/her team to complete their section of the programming. By allowing for multiple estimates of a single value in the database a method for determining what value(s) will be provided when the database is queried must be determined. If the user also happens to be an estimator of values in the database he or she could select their own estimates to be provided by a query. However, most users will only be estimators of a few of the database values and some users may be outside the company and have no access rights at all. For these users the system must have the means for providing and handling all the estimates of a queried value or for combining the estimates into one or a few values that can be presented to the user.

The system must also maintain the integrity of data when operations are performed on values with multiple estimates.

UNION, INTERSECTION AND COMPLEMENT OPERATIONS

One method that has been proposed for handling queries on fuzzy data is to use the fuzzy set operations of Union, Intersection and Difference. Umano (Uma83), for example, uses the fuzzy set operation Union to do the projection of a fuzzy relation. The three operations of Union, Intersection and Complement are shown in figure 4.1 below.

Union: A U B = Ax V Bx

Intersection: A B = Ax ^ Bx

Complement: Ax = 1 - Ax

Figure 4.1: Fuzzy Set Operations: Union, Intersection, Complement

The operators V, ^ and - represent maximum, minimum and arithmetic difference, respectively. The result of the Union operation is the maximum of the certainty values. The result of the Intersection is the mimimum of the certainty factors. The result of the complement is 1 minus the certainty factor. To demonstrate the effect of these operations on a fuzzy set the following fuzzy set THIN provided in Chapter 2 is shown again in figure 4.2 on the following pages, with the result of

the Union, Intersection and Difference operations each given in order below the original set. A second set FAT, not previous mentioned, is used for the Union and Intersection operations.

THIN

WEIGHT	DEGREE OF MEMBERSHIP
110	1.00
125	.95
140	.8
155	.5
180	.45
195	.3
210	.2
225	0.00

FAT

WEIGHT	DEGREE OF MEMBERSHIP
110	0.00
125	.08
140	.25
155	.45
180	.55
195	.75
210	.89
225	1.00

UNION (THIN & FAT)

WEIGHT	DEGREE OF MEMBERSHIP
110	1.00
125	.95
140	.8
155	.5
180	.55
195	.75
210	.89
225	1.00

INTERSECTION (THIN & FAT)

WEIGHT	DEGREE OF MEMBERSHIP
110	0.00
125	.08
140	.25
155	.45
180	.45
195	.3
210	.2
225	0.00

COMPLEMENT
(THIN)

WEIGHT	DEGREE OF MEMBERSHIP
110	0.00
125	.05
140	.2
155	.5
180	.55
195	.7
210	.8
225	1.00

Figure 4.2: Example of Set Operations:
Union, Intersection, Complement

In the same manner as the Union operation shown above Umano's (Uma83) projection is the maximum of the certainty factors in the fuzzy relation. An example of a Union operation as a projection on a fuzzy set is given in figure 4.3 below.

STUDENTS

Name	Major	Grade
Jim	EE	3.5
Sally	Biol.	(2.5/.8)
Ann	CS	2.8
John	Biol.	(2.5/.65), (2.9/.8)
Ben	EE	(3.2/.65)

Projection[STUDENTS]: Grades of Biology Majors =

Major	Grade
Biol.	(2.5/.8), (2.9/.8)

Figure 4.3: Example Projection using Union Operation

The projection results in only the maximum certainty

factor of .8 remaining for the estimated value of 2.5 and the lower certainty factor of .65 is lost.

ALGEBRAIC SUM OPERATION:

Although the fuzzy operations of Union, Intersection and Difference (to a lesser degree) have gained acceptance their application has mainly been limited to expert systems. The use of these operations on null or estimated values seems limited, as in Umano's projection operation (Uma83), in that information is lost when only maximum or minimum values are preserved. A more reasonable solution is provided by the fuzzy set operation Algebraic Sum. Novak and Nekola (NovNek83) list the Algebraic Sum operation as an alternative to the Union operation. The Algebraic Sum operation preserves the fact that a second or multiple estimated values were present by combining the certainty factors. The Algebraic Sum operation is defined as follows: Given the fuzzy sets A and B with degree of membership values xA and xB the Algebraic Sum of these sets is given in figure 4.4.

$$A + B = 1 - (1 - xA)(1 - xB)$$

Figure 4.4: Algebraic Sum Operation

The result of the Algrebraic Sum operation is a single value that is higher than both of the degree of

memberships of the individual sets. The basis for this result is that given two sets of data the answer is more likely than with a single set or in the case of estimates, if two people estimate the same value that value should have a higher certainty than a value estimated by only one person. The effect of the fuzzy set operation Algebraic Sum on a projection operation is to combine the certainty factors of like estimates. For these reasons, the Algebraic Sum operation was incorporated into this model. Using the same example relation and projection shown for the Union operation earlier the results with the Algebraic Sum operation are given in figure 4.5 below.

STUDENTS

Name	Major	Grade
Jim	EE	3.5
Sally	Biol.	(2.5/.8)
Ann	CS	2.8
John	Biol.	(2.5/.65), (2.9/.8)
Ben	EE	(3.2/.65)

Projection[STUDENTS]: Grades of Biology Majors =

Major	Grade
Biol.	(2.5/.93), (2.9/.8)

$(A + B) = 1 - (1 - xA)(1 - xB)$

$(2.5 + 2.5) = 1 - (1 - .8)(1 - .65)$

$= 1 - (.2)(.35) = .93$

Figure 4.5: Example Projection using Union Operation

With the Algebraic Sum operation the single fuzzy value of 2.9 remains the same but the fuzzy value 2.5 has a certainty factor calculated using the Algebraic Sum operation. The calculation is shown directly below the result of the projection. Since two estimates were made of the single value 2.5 the certainty factor becomes much higher than with the one estimate of the value 2.9.

The Algebraic Sum operation can also be used to combine a large number of estimates for a single value in the database. One instance where a value would have a large number of estimates attached to it would be when the user desired to rate the possible estimated values from high to low. The user could provide estimators with a range of values and ask each person to rate the estimates from "EXTREMELY LIKELY" to "QUESTIONABLE" (as used in the fuzzy set ACCURACY). An example of using the Algebraic Sum operation for this purpose is provided in the following example: Given a set of possible estimates for the value - # of units sold and the accuracy of those estimates as provided by three users the result of the Algebraic Sum operation would be as shown in figure 4.6 on the following page:

# of Units Sold	Sam	Joe	Jim	Algebraic Sum
10,000			A2	.65
9,000	A1		A5	.955
8,000	A5		A4	.995
7,000	A4	A2	A3	.993
6,000	A3	A4	A1	.99
5,000	A2	A5		.965
4,000		A3		.8
3,000		A1		.5
2,000				
1,000				

Ex. (5,000): $1 - (1 - xA)(1 - xB) =$
$1 - (1 - .65)(1 - .9) =$
$1 - (.35)(.1) = 1 - .035 = .965$

Figure 4.6: Example Algebraic Sum Operation

The example given below the table shows how the result for the estimate of 5,000 units was obtained. The estimated values that were selected 2 or 3 times have significantly higher results than those only selected once. At this point 15 estimates have been reduced to 8 estimates and the system could select all 8 estimates to be provided when queried. If all the estimates are not deemed useful or there are still too many the single highest estimate or the top few estimates selected by statistical or other means could be provided.

Having provided a means for performing the projection operation in a database query the selection operation should also be addressed. The selection operation on fuzzy data in a relation can be viewed as Grant approached the maybe operation created by Codd.

A maybe selection as discussed in Chapter 2 selects all tuples that intersect with the value or range of values provided in the query. An example of the maybe selection as performed on a set of estimated values is provided in the figure 4.7 below.

XYZ SALES

SOLD TO	LOCATION	SALES
Ace	Toledo	($1,500/.65)
Burns	Dallas	$2,000
CCC	Miami	($500/.8,$375/.9)
Dow	New York	($450/.95)

If the Query is: Select XYZ SALES where Sales)= 500

the following selections would result.

MAYBE SELECTION

Ace	Toledo	($1,500/.65)
Burns	Dallas	$2,000
CCC	Miami	($500/.8,$375/.9)

Figure 4.7: Maybe Selection on Estimated Values

The tuples which fell in the range of sales equal or over $500 were selected. Both the values for the tuple CCC were provided in the selection even though only one estimate fell in the range given. This informs the user that the estimate of $500 was not the only estimate and in fact in this instance a higher certainty estimate of $375 exists.

CARDINALITY:

Before leaving the subject of fuzzy set operations a definition of cardinality or counting of fuzzy set elements should be provided. Zadeh (Zad83) uses the notion of a Sigma-Count which is the arithmetic sum of the degrees of membership. The arithmetic sum of the fuzzy set THIN is shown in figure 4.8 below.

THIN

WEIGHT	DEGREE OF MEMBERSHIP
110	1.00
125	.95
140	.8
155	.5
180	.45
195	.3
210	.2
225	0.00

Sigma-Count(THIN) = DEGREE OF MEMBERSHIP = 4.20

Figure 4.8: Cardinality of Fuzzy Set - THIN

The resulting value may be rounded off to the nearest integer if needed. Unfortunately the Sigma-Count can be a deceptive value when the fuzzy data has values with low degrees of membership. A large number of low degrees of membership can become count-equivalent to a small number of terms with high membership. Both Zadeh (Zad83) and Buckles and Petry (BucPet82) suggest the same solution for this problem. They both suggest some

type of minimum threshold value to determine if a degree of membership is included in the cardinality. Neither of the authors, however, suggest how the thresholds should be determined. Another more recent approach to cardinality is provided by Wygralak (Wyg86) and introduces the notion of fuzzy semicardinals. A fuzzy semicardinal is essentially a range of values using fuzzy numbers. Wygralak explains "...we construct fuzzy natural numbers defining degrees to which a finite fuzzy subset has, respectively, at most/least and less than/more than k elements."

CHAPTER 5: TESTING THE ALGEBRAIC SUM OPERATION

The implementation portion of this study centers on the Algebraic Sum operation used to combine estimated values. The intention of this implementation is to study the results from the Algebraic Sum operation using widely varying test data. By testing a wide range of estimates, the results provide a basis for making a determination of the soundness, reasonableness and consistency of the result of the Algebraic Sum operation.

The estimated data are entered onto a spreadsheet in the form of the model given in Chapter 3. The spreadsheet was selected for several reasons. First, the spreadsheet provides a clear mechanism for the manipulation of data with a formula. The Algebraic Sum operation is stored as a formula for operating on certainty factors. Secondly, multiple sets of test data are generated by simply dividing the certainty factors into ranges and raising or lowering one or more ranges while other ranges remain unchanged. Finally, the form of the spreadsheet allows a clear illustration of both the data and results on the same screen or printed page. With a database the results would be separated from the estimates making test data comparisons more difficult.

Each component of the data model is located in a separate column of the spreadsheet as shown in figure 5.1 below.

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/5	5/25/86	$5,000	.9	Jim
5			08:05:30			
6	XYZ	Sales/2	4/29/86	$3,000	.8	Ann S.
7			12:57:35			
8	Ace	Time/4	6/1/86	200	.95	Art
9			16:03:43			
10	CCC	Cost/7	3/25/86	$3.50	.65	Jean
11			09:00:21			

Figure 5.1: Sample Spreadsheet of Estimated Values

The Algebraic Sum operation is entered as a formula operating on the column corresponding to the certainty factors for the estimates. Certainty factors that are being combined are selected and addressed through the formula.

Test data are entered to simulate both single and multiple estimates of a unique database value. The test data varies both in degrees of membership and in the number of estimates. Sets of test data include estimates with high certainty factors, low certainty factors and a wide and narrow range of certainty factors. Each of these sets of data is tested for instances where numerous estimates or only a few estimates are available. The results of the Algebraic

Sum operation are compared and contrasted between each set of test data.

The spreadsheet provided in figure 5.2 illustrates the method used for varying test data. Sets of test data are grouped into ranges and seperated from other data with a solid line. Figure 5.2 is shown below.

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	$5,000	.9	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	$2,000	.8	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	$6,500	.95	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	$4,000	.9	Abe
11			15:22:00			
12	Ace	Sales/5	4/23/86	$5,000	.5	Joe
13			12:25:49			
14	Ace	Sales/5	5/26/86	$4,000	.65	Sam
15			14:25:39			
16	Ace	Sales/5	6/30/86	$5,000	.7	Jean
17			16:22:21			
18	Ace	Sales/5	7/21/86	$6,500	.5	Art
19			09:35:44			

Figure 5.2: Spreadsheet of Test Data Sets

The ranges of four elements each are shown in the Estimate, Value and Certainty Factor columns. By changing the values within each range the test data is varied. For example, if the values in the lower range of the Estimate column were all changed to 'Sales/2' the sample spreadsheet would represent eight estimates of one database value instead of two sets of four

estimates for two database values. The ranges in the Value column allow for changing from multiple to single estimates. By changing all the values in the top range to '$4,000' the test data has changed from four different estimated values to one estimated value to be combined using the Algebraic Sum operation. One use for the certainty factor ranges is to compare results from values that are grouped closely together or far apart. By lowering all the values in the top certainty factor range and raising the values in the bottom range the result is a wide and narrow range of test data.

The testing method described above is also used to simulate the projection operation in a relational database. Assuming the Name column does not change value, the two ranges of the Estimate column in figure 5.2 above represent two different tuples of a relation. By treating all the estimated values of each tuple (or range) as a group the tuples can be combined as in a projection operation. With more test data sets the projection operation is simulated for queries involving a few or a large number of tuples.

RESULTS:

The Algebraic Sum operation was tested on estimated values with two, four and eight similar estimates. The comparison of two similar estimates is

the smallest number of estimates that could be combined. The comparison of four and eight estimates provide a middle and high number of similar estimated values. The comparison of over eight values was not informative because all results from the Algebraic Sum operation were identical to six significant digits (the accuracy provided by the spreadsheet program). The results of the Algebraic Sum operation are provided in the Appendix in figures A.1 through A.7. A discussion and analysis of those results with reference to the figures follows.

The comparison of two similar estimates is provided in Figure A.1. The results of the Algebraic Sum operation on two high and two low certainty factors have a significant difference of .995 to .775. The comparison of a high and a low certainty factor with two medium certainty factors results in the high and low certainty factor (.975) being above the two medium certainty factors (.95). The high certainty factor has a more significant influence on the result than the low certainty factor. This results because the Algebraic Sum operation weights upward to demonstrate that two different individuals selected the value, not one. Algebraic Sum results on two estimated values are two high certainty factors (.995) first, a low and high certainty factor (.975) second, two medium certainty

factors (.95) third and two low certainty factors (.775) fourth.

In figures A.2 and A.3 the results are provided for the Algebraic Sum operation on four similar estimated values. On comparison the results are very similar to the results for two estimates. The high and low certainty factors have a significant difference in their resulting values (.999975-high & .949375-low). The Algebraic Sum of a wide range of certainty factors results in a higher value than the Algebraic Sum of a narrow range of medium certainty factors (.998875-wide & .99625-narrow). As with two estimates, the high certainty factors in the wide range have a more significant influence on the result than the low certainty factors in that range. The ranking of the Algebraic Sum results for four estimated values is four high certainty factors (.99975) first, four varied or wide range of certainty factors (.998875) second, four medium certainty factors (.99625) third, and four low certainty factors (.949375) fourth.

In comparing the Algebraic Sum operation on two and four certainty factors the four similar estimates are significantly higher that the two similar estimates in each area tested. For example the result for two high certainty factors was .995 and for four high certainty factors was .999975. This result follows the reasoning

behind the Algebraic Sum operation in that a four different individuals selecting the same estimated value is more significant than two individuals selecting the same estimated value.

Figures A.4 through A.7 provide the results of the Algebraic Sum operation on eight similar estimated values. The results follow in the same order of ranking as with the two and four certainty factor results (.999999-high, .999994-varied or wide range, .999988-medium or narrow range, .998405-low). The difference in the values resulting from the high, varied and medium certainty factors is not very significant with only the low certainty factor result showing a noticable difference in value. This lack of difference is due to the results extending beyond the number of significant digits provided by the spreadsheet. As more Algebraic Sum operations are performed the values approach .999999, which is the limit of significance for the spreadsheet. The comparison of the results for the eight certainty factors with the results for the four and two certainty factors is significant between the eight and two with little difference between the eight and four results. For the high certainty factors the results for eight values is .999994, for four values is .999975 and for two values is .995.

The overall soundness, reasonableness and consistency of the Algebraic Sum operation tests are favorable. The soundness of the operation is shown by all results falling in the range of 0..1. The resonableness of the results can be viewed in the ranking of the results and the comparison of two, four and eight estimated values. In the ranking the high certainty factors always had the highest resulting value and the low certainty factors had the lowest resulting value. The certainty factors with high and low estimates had higher results than those with medium estimates. This seems reasonable since the higher certainty factors are weighted more than the lower certainty factors thus not allowing an uncertain estimate to pull down the result. The comparison of two, four and eight certainty factors was reasonable in showing that as the number of individuals selecting the same estimated value increased the result was significantly higher. The difference between the results for four and eight certainty factors seemed to diminish because of a loss of significant digits. The consistency of the results was demonstrated by the rankings remaining the same for the two, four and eight certainty factors. The results from eight certainty factors was consistent but the loss of significant digits would make comparison or analysis of more than

eight similar estimated values useless.

CHAPTER 6: THE ESTIMATE AS A USABLE NULL VALUE

The estimated null value provides information to the user that has previously been ignored. By expanding null values beyond current definitions of 'at present unknown', 'nothing', 'no value possible', and others valuable data is provided to the user. The information included in the estimated null values represents the expertise of people who work constantly in the areas they are estimating. This research has resulted in the following:

- an expanded model of a datum
- a model for estimated null values
- a method based on the Algebraic Sum for processing these null values
- use of Grant's maybe operation to select the tuples.

EXPANDED DATUM:

The basic attributes of a data item, the name and value, have been expanded with additional informational components. The components of Name and Estimate uniquely identify the database value being estimated. The Value and Certainty Factor components provide an estimated value and the accuracy of the estimate. The components of Creator and Time allow the user to identify the individual making the estimate and the time the estimate was entered. The Creator and Time

components can also be utilized to selectively process certain estimates while ignoring others.

A MODEL FOR ESTIMATED NULL VALUES:

The model for an estimated value provides all the vital data of what, when, whom and even how accurate the estimate is believed to be. Each estimate has a unique set of data associated with it. The estimates and data can be easily maintained in the form of a relation to fit into the relational database format. Access rights limit the individuals allowed to enter estimates to those deemed as qualified. Multiple estimates of an individual value in the database allow comparison between estimates and provide a consensus of information for user queries.

PROCESSING ESTIMATED NULL VALUES - ALGEBRAIC SUM:

Fuzzy set theory provides a means of processing estimated null values without losing valuable information. The fuzzy set operation of Algebraic Sum is used to combine the certainty factors for similar estimates. The Algebraic Sum operation produces a new certainty factor of higher value to represent the combining of multiple estimated values. The Algebraic Sum operation can be performed on two or more estimated values. A large number of estimates on a single database value can be combined without losing the range

of estimates or giving estimates occuring only once the same weight as estimated values selected several times. The Algebraic Sum operation can also be used for projection operations to maintain integrity of the data by adjusting the certainty factor of similar estimates that are combined.

SELECTING ESTIMATED NULL VALUES - MAYBE OPERATION:

The maybe operation developed by Grant for ranges of null values fits in well with estimated fuzzy data. A maybe selection operation produces all tuples that fall in the range of the query. The user is provided with all possible and known values for a query and thus has more complete information to base a decision on. Even estimates not in the range of the query but in the tuple with a selected estimate are provided so the user can be aware of all estimates made on the database value.

FUTURE WORK:

The Algebraic Sum operation of fuzzy set theory is a beginning for manipulating fuzzy data in a relational database. Although the Algebraic Sum is well accepted as a part of fuzzy logic the application in a fuzzy database should be supported with proven theorems and tested data. The implementation discussed in Chapter 5 starts the process of testing results from operations

on estimated fuzzy data. At a later time the use of the maybe operation and fuzzy data should also be supported with additional theoretical work. The maybe operation should be tested on a varied set of estimated values. The estimated values should have a wide and narrow range to test the range overlap function of the maybe selection operation.

Cardinality or the counting of estimates is still an uncertain area. Although the means of counting members of a fuzzy set has been provided by several individuals the accuracy of the count for data with low degrees of memberships is in question. The solution of setting a threshold to degrees of memberships that are summed is not viable if no means for establishing the threshold is provided. The development of a method for determining the threshold value of cardinality on fuzzy data is an obvious area for future work.

CONCLUSION:

This representation of estimated values as fuzzy data in a relational database demonstrates the wealth of information not currently captured by null values. The Algebraic Sum and maybe operations both provide the user with complete and consistent information in response to a query. The fuzzy database system is a natural basis for handling estimated null values.

BIBLIOGRAPHY

(Bis83) Biskup, Joachim "A Foundation of Codd's Relation Maybe-Operations." ACM Transactions on Database Systems, 8(4):608-636, December 1983.

(Bor85) Borgida, Alexander "Handling Exceptions in Information Systems." ACM Transactions on Database Systems, 10(4):594-595, December 1985.

(BorKru83) Boriskov, Arkady & Krumber, Oyar "A Theory of Possibility for Decision Making." Fuzzy Sets and Systems, 9:13-23, 1983.

(BucPet82) Buckles, Billy P. & Petry, Frederick E. "A Fuzzy Representation of Data for Relational Databases." Fuzzy Sets and Systems, 7:213-226, 1982.

(Cod75) Codd, E.F. "Understanding Relations." FDT Bulletin. ACM SIGMOD 7, part 3-4, 1975. pp. 23-28.

(Cod79) Codd, E.F. "Extending the Database Relational Model to Capture More Meaning." ACM Transactions on Database Systems, 4(4):397-434, December 1979.

(Gla83) Glas, Michael de "Theory of Fuzzy Systems." Fuzzy Sets and Systems, 10:65-77, 1983.

(Gol85) Golshani, Forouzan "Growing Certainty with Null Values." Information Systems, 10(3):289-297, 1985.

(Gra77) Grant, J. "Null Values in a Relational Database." Information Processing Letters, 6(5):156-157, 1977.

(ImiLip84) Imielinski, Tomasz & Lipski, Witold Jr. "Incomplete Information in Relational Databases." Journal of the Association for Computing Machinery, 31(4):761-791, October 1984.

(Lip79) Lipski, Witold Jr. "On Semantic Issues Connected with Incomplete Information Databases." ACM Transactions on Database Systems, 4(3):263-296, September 1979.

(MizTan81) Mizumoto, Masaharu & Tanaka, Kokichi "Fuzzy Sets and Their Operations." Information and Control, 48:30-48, 1981.

(Nat83) Natvig, B. "Possibility Versus Probability." Fuzzy Sets and Systems, 10:31-36, 1983.

(Neg81) Negoita, C.V. Fuzzy Systems. (Cybernetics and Systems Series: 2). Kent, England: Abacus Press, 1981.

(Neg85) Negoita, Constantin Virgil Expert Systems and Fuzzy Systems. Menlo Park, California: The Benjamin/Cummings Publishing Co., Inc., 1985.

(Nid60) Nidditch, P.H. Elementary Logic of Science and Mathematics. Glencos, Illinois: The Free Press, 1960.

(Noj82) Nojiri, Hideyuki "A Model of the Executive's Decision Processes in New Product Development." Fuzzy Sets and Systems, 7:227-241, 1982.

(NovNek83) Novak, V. & Nekola, J. "Basic Operations with Fuzzy Sets from the Point of Fuzzy Logic." Fuzzy Information, Knowledge Representation and Decision Analysis: Proceedings of the IFAC Symposium, Marseille, France 19-21 July 1983. Edited by E. Sanchez. (IFAC Proceedings Series). Oxford: Pergamon PRess, 1984. pp. 249-253.

(Rad83) Radecki, Tadeusz "A Theoretical Background for Applying Fuzzy Set Theory in Information Retrieval." Fuzzy Sets and Systems, 10:169-183, 1983.

(Res69) Rescher, Nicholas Many-Valued Logic. New York: McGraw-Hill, 1969.

(Uma83) Umano, M. "Retreival From Fuzzy Database by Fuzzy Relational Algebra." Fuzzy Information, Knowledge Representation and Decision Analysis: Proceedings of the IFAC Symposium, Marseille, France 19-21 July 1983. Edited by E. Sanchez. (IFAC Proceedings Series). Oxford: Pergamon Press, 1984. pp. 1-6.

(Vas79) Vassiliou, Y. "Null Values in Data Base Management: A Denotational Semantics Approach." Proceedings of the ACM-SIGMOD International Symposium on Management of Data (Boston, Mass., May 30 - June 1). New York: ACM, 1979. pp.162-169.

(Vas80) Vassiliou, Y. "Functional Dependencies and Incomplete Information." Proceedings of the 6th Conference on Very Large Data Bases (Montreal, Ont., Canada, Oct. 1-3). New York: ACM, 1980. pp. 260-269.

(Wyg86) Wygralak, Maciej "Fuzzy Cardinals Based on the Generalized Equality of Fuzzy Subsets." Fuzzy Sets and Systems, 18:143-158, 1986.

(Zad83) Zadeh, L.A. "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems." Fuzzy Sets and Systems, 11:199-227, 1983.

APPENDIX

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	5000	0.9	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	5000	0.95	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	6500	0.55	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	6500	0.5	Abe
11			15:22:00			
12	Ace	Sales/5	4/23/86	4000	0.5	Joe
13			12:25:49			
14	Ace	Sales/5	5/26/86	4000	0.95	Sam
15			14:25:39			
16	Ace	Sales/5	6/30/86	3500	0.75	Jean
17			16:22:21			
18	Ace	Sales/5	7/21/86	3500	0.8	Art
19			09:35:44			

	Values	Results
20		
22	E4-E6 (2-high)	0.995
23	E8-E10 (2-low)	0.775
24	E12-E14 (low & high)	0.975
25	E16-E18 (2-medium)	0.95

Figure A.1: Algebraic Sum of Two Estimated Values

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	5000	0.9	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	5000	0.95	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	5000	0.95	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	5000	0.9	Abe
11			15:22:00			
12	Ace	Sales/5	4/23/86	3500	0.5	Joe
13			12:25:49			
14	Ace	Sales/5	5/26/86	3500	0.55	Sam
15			14:25:39			
16	Ace	Sales/5	6/30/86	3500	0.5	Jean
17			16:22:21			
18	Ace	Sales/5	7/21/86	3500	0.55	Art
19			09:35:44			
20						
21		Values			Results	
22		E4-E10 (4-high)			0.999975	
23		E12-E18 (4-low)			0.949375	

Figure A.2: Algebraic Sum of Four Estimated Values (high & low)

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	5000	0.8	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	5000	0.95	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	5000	0.75	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	5000	0.55	Abe
11			15:22:00			
12	Ace	Sales/5	4/23/86	3500	0.7	Joe
13			12:25:49			
14	Ace	Sales/5	5/26/86	3500	0.75	Sam
15			14:25:39			
16	Ace	Sales/5	6/30/86	3500	0.8	Jean
17			16:22:21			
18	Ace	Sales/5	7/21/86	3500	0.75	Art
19			09:35:44			
20						
21		Values			Results	
22		E4-E10	(4-wide range)		0.998875	
23		E12-E18	(4-narrow range)		0.99625	

Figure A.3: Algebraic Sum of Four Estimated Values (wide & narrow range)

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	5000	0.7	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	5000	0.95	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	5000	0.75	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	5000	0.55	Abe
11			15:22:00			
12	Ace	Sales/2	4/23/86	5000	0.65	Joe
13			12:25:49			
14	Ace	Sales/2	5/26/86	5000	0.9	Sam
15			14:25:39			
16	Ace	Sales/2	6/30/86	5000	0.8	Jean
17			16:22:21			
18	Ace	Sales/2	7/21/86	3500	0.5	Art
19			09:35:44			
20						
21		Values			Results	
22		E4-E18	(8-varied)		0.999994	

Figure A.4: Algebraic Sum of Eight Estimated Values (varied)

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	5000	0.9	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	5000	0.95	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	5000	0.85	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	5000	0.9	Abe
11			15:22:00			
12	Ace	Sales/2	4/23/86	5000	0.95	Joe
13			12:25:49			
14	Ace	Sales/2	5/26/86	5000	0.9	Sam
15			14:25:39			
16	Ace	Sales/2	6/30/86	5000	0.85	Jean
17			16:22:21			
18	Ace	Sales/2	7/21/86	5000	0.95	Art
19			09:35:44			
20						
21		Values			Results	
22		E4-E18	(8-high)		0.999999	

Figure A.5: Algebraic Sum of Eight Estimated Values (high)

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	5000	0.55	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	5000	0.5	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	5000	0.6	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	5000	0.55	Abe
11			15:22:00			
12	Ace	Sales/2	4/23/86	5000	0.5	Joe
13			12:25:49			
14	Ace	Sales/2	5/26/86	5000	0.55	Sam
15			14:25:39			
16	Ace	Sales/2	6/30/86	5000	0.65	Jean
17			16:22:21			
18	Ace	Sales/2	7/21/86	5000	0.5	Art
19			09:35:44			
20						
21		Values			Results	
22		E4-E18	(8-low)		0.998405	

Figure A.6: Algebraic Sum of Eight Estimated Values (low)

	A	B	C	D	E	F
1	Name	Estimate	Time	Value	Certainty Factor	Creator
2						
3						
4	Ace	Sales/2	5/25/86	5000	0.8	Jim
5			08:05:30			
6	Ace	Sales/2	5/26/86	5000	0.7	Sal
7			07:58:30			
8	Ace	Sales/2	6/01/86	5000	0.75	Glen
9			19:25:45			
10	Ace	Sales/2	7/25/86	5000	0.75	Abe
11			15:22:00			
12	Ace	Sales/2	4/23/86	5000	0.8	Joe
13			12:25:49			
14	Ace	Sales/2	5/26/86	5000	0.8	Sam
15			14:25:39			
16	Ace	Sales/2	6/30/86	5000	0.7	Jean
17			16:22:21			
18	Ace	Sales/2	7/21/86	5000	0.75	Art
19			09:35:44			
20						
21		Values			Results	
22		E4-E18	(8-medium)		0.999988	

Figure A.7: Algebraic Sum of Eight Estimated Values (medium)

FUZZY LOGIC, ESTIMATED NULL VALUES AND THEIR APPLICATION IN RELATIONAL DATABASES

by

SUSAN E. POWELL

B.S., Kansas State University, 1980

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1986

ABSTRACT

Since knowledge may be incomplete the relational database must provide a means of representing values collectively not known by the user.

In existing database systems uncertain, incomplete or inconsistent information is represented by values called nulls. Unfortunately the information provided by null values is severely limited by predicate logic and probability-based methods of processing data. An alternative approach to managing null values suggested in this work is based on the use of fuzzy logic. Fuzzy logic provides the natural basis for handling null values by extending the framework of the systems to deal with both probabilistic and possibilistic theory.

Null values, which can be estimated, are represented in the database by a 6-tuple model. Each estimated null value has an attached certainty factor corresponding to the accuracy of the estimate. Acceptable terms for use as certainty factors are defined by a fuzzy set - ACCURACY. Each term in the fuzzy set - ACCURACY - has a degree of membership representing inclusion in the set.

The Algebraic Sum operation of fuzzy logic utilized to combine estimates and implement projection operations. Grant's maybe operation compares ranges of values for processing selection operations.