# IMPERFECT DATA IN DATABASE CONTEXT. HOW ARE THEY STORED IN EXTENDED RELATIONAL DATABASES ?

**Cornelia TUDORIE**

*Department of Computer Science, University "Dunarea de Jos", Galati*
*Domneasca 111, 6200 Galati, Tel, Fax: 460182*
*email: Cornelia.Tudorie@ugal.ro*

Abstract: Building a more accurate reality model requires taking into account imperfect information present in our knowledge and language. This paper presents several aspects of data imperfection in the database context and the appropriate frameworks for their treatment. It's concluding that null value, possibility distribution and probability theory are the best solutions to represent incomplete, imprecise and uncertain data. For each of these problems there are some relational model extension proposals, including data representation and relational algebra.

Keywords: Artificial Intelligence, Database, Fuzzy Logic, Possibility Theory

## 1. INTRODUCTION

The relational model was for a very long time considered the best solution to organize lots of data and for their efficient exploitation. The practice has proved that the classical relational model doesn't reflect the reality with enough fidelity. A great part of information come from various sources, like: *process*, where to accurately measure data on a numerical scale is very difficult (so it is better to work with symbols), or the *human usual communication*, where the natural language expression includes vague terms. This is the reason why research groups began to work on the relational model extension, in order to deal with imperfect data.

In a database context, the imperfection of data may appear at two levels:

1) in user queries sent to DBMS (DataBase Management System) – vague criteria used when selecting objects.
2) in data stored and managed by DBMS – vague attribute values or uncertain relationships between objects

The first issue is discussed in (Tudorie et al., 2001); it concerns vague queries addressed to crisp (classical) databases, and how they may be interpreted and processed.

How the imprecision and the uncertainty may be involved in database area?

The paper presents the main aspects of imperfection of data (mainly imprecision) stored in databases and the methods to manage it.

The second section discusses the nature of data imperfection and its classification.

The next sections deal with three kinds of imperfect data and some approaches to manage each of them.

Finally, some personal results and future prospects are presented.

## 2. IMPERFECT DATA STORED IN DATABASE

The discussion is related to relational databases, where all information may be expressed as Attribute-Object-Value (in conformity with the AOV theory). Examples: "Age - of Peter - is 26", "Number of students – at the Computer Science Faculty – is 200". The relational database consists in a set of expressions following this format. How can these expressions format be modified as to support data imperfection?

What kinds of techniques can manage them?

**Incompleteness** can have existential meaning, in the case of a single missing value ("The address of Peter is ..?.."), or it can be universal, in the absence of all values of an object attribute ("The marks of Peter are ..?.."). The absence of the value can have two possible significations: *unknown*, if the value exists

This paper was recommended for publication by Severin BUMBARU

but it is not known, or *inapplicable*, if the attribute refers a nonexistent feature for the specified object.

**Imprecision** typically characterises situations when the exact value of an attribute is not well-known or is vague. For example, if one is evaluating the age of a person without knowing it's exact numerical value, he may say: "between 25 and 30", "about 27", or "young".

**Uncertainty** is related to the truth of the proposition with regard to the reality. A degree of uncertainty expresses the doubts about the truth. Uncertainty may be expressed also trough linguistic phrases like "it is possible that", "probable", "almost certain".

Certainly, there are situations where information is affected by both uncertainty and imprecision. Example: "It is unlikely possible that Peter is young". So, uncertainty and imprecision may interfere with the knowledge and generally, they are correlated: when one of them is increasing, the other one is decreasing.

For example if somebody says that "Peter is between 20 and 30" he may have full reason without risk (null uncertainty). Instead if he wants to be more precise, and he says "Peter is 26 or 27", of course the uncertainty degree should increase.

## 3. INCOMPLETE DATA IN RELATIONAL DATABASE

The point of view shared by most researchers is to consider the *null* value (introduced in (Codd, 1979)) as the best solution to represent and store the absence of an attribute value; it replaces the attribute effective value. The relational tables containing such values are named **Codd-tables**. The only difficult problem is interpreting the meaning of the null values in databases that has to be modelled and taken into account in the query evaluation process.

An extension of relational algebra is necessary; so, the relational operators (projection, selection, junction) are redefined in order to interpret in a proper way the null values. Moreover, they must maintain the null-value semantic during the entire query treatment process.

W. Lipski (Lipski, 1979) gives an interesting proposal for considering the null values. For example, if one asks for all blue products, two sets will be provided, corresponding to the two limits of the response set: the products certainly blue, and the products certainly not blue.

Tables with variables (**V-tables**) are more flexible for unknown data processing, when some additional information is available. Variables replace the null values, so that it's possible to mark either the tuples having the same attribute value, even unknown, or the tuples having different attribute values, even unknown.

A more efficient type of table is the **conditional table** where an additional column contains constraints for the possible but unknown values.

Finally, the incompleteness can affect the database structure too, when the object is incompletely described by the table structure; the situation when a certain value can't be stored in the database may occur due to the insufficient schema of the database (Zicari, 1993).

## 4. IMPRECISE DATA IN RELATIONAL DATABASE

The possibility theory (Dubois and Prade, 1988) offers a unified adequate framework to manage imprecise information in a unique formal context.

Generally, the possibility distribution restricts the possible domain values for a certain variable. In the database context, it defines the possible values of an attribute A for a tuple *x*.

Let D be the domain of the attribute A and *e* an extra-element corresponding to the case when the attribute does not apply to the object. In the case when the exact value of the attribute A for the tuple *x* is not known, a possibility distribution $\pi_{A(x)}$ can replace it. The possibility distribution is defined:

$$\pi_{A(x)} : D \cup \{e\} \rightarrow [0,1]$$

The degree $\pi_{A(x)}(d)$ indicates the possibility that d is the exact value of the attribute A for the object represented by the tuple *x*.

$\pi_{A(x)}(d) = 1$ shows that it's completely possible that d is the A attribute value, but not certainly (or not necessary). This information is in the same time precise (possible and necessary), only if $\pi_{A(x)}(d) = 1$ and $\pi_{A(x)}(d') = 0$, $\forall d' \neq d$.

In a consistent and coherent state, the normalisation constraint is satisfied, that is

$$\max_d \pi_{A(x)}(d) = 1 \ \ , d \in D \cup \{e\}$$

It is equivalent to say that at least one domain value or *e* is completely possible.

Bosc and Prade (1997) proved that this approach based on possibility distribution allows the representation of all types of well-known, ill-known or unknown values for attributes in a database context.

For instance, let D be the domain of an attribute Comm (Commission). Here are several variants of attribute value representation for a tuple x, all of them using possibility distribution on D ∪ {e}:

1) the attribute is not applicable on x
$$\pi_{Comm(x)}(e) = 1 \ ; \ \pi_{Comm(x)}(d) = 0 \ \ , \forall d \in D$$

2) unknown but existing value
$$\pi_{Comm(x)}(e) = 0 \ ; \ \pi_{Comm(x)}(d) = 1 \ \ , \forall d \in D$$

3) total ignorance (everything is possible)

$\pi_{Comm(x)}(e) = 1$ ; $\pi_{Comm(x)}(d) = 1$ , $\forall d \in D$

4) precise value (Comm=200):

$\pi_{Comm(x)}(200) = 1$

$\pi_{Comm(x)}(e) = 0$ ; $\pi_{Comm(x)}(d) = 0$ , $\forall d \in D \setminus \{200\}$

5) disjunctive information (Comm is 200 or 400)

$\pi_{Comm(x)}(200) = 1$ ; $\pi_{Comm(x)}(400) = 1$

$\pi_{Comm(x)}(e) = 0$ ; $\pi_{Comm(x)}(d) = 0$ , $\forall d \in D\setminus\{200,400\}$

6) interval type value (Comm is between 200 and 400)

$\pi_{Comm(x)}(d) = 1$ , $\forall d \in [\ 200,400\ ]$

$\pi_{Comm(x)}(e) = 0$ ; $\pi_{Comm(x)}(d) = 0$ , $\forall\ d \in D\setminus[200,400]$

7) discrete possibility distribution type value (Comm is completely possible 200 or 300, or it's 70% possible to be 400)

$\pi_{Comm(x)}(200)=1$; $\pi_{Comm(x)}(300)=1$; $\pi_{Comm(x)}(400)=0.7$

$\pi_{Comm(x)}(e)=0$; $\pi_{Comm(x)}(d)=0$ ,$\forall d \in D\setminus\{200,300,400\}$

8) label type value (Comm is *low*)

$\pi_{Comm(x)}(e) = 0$ ; $\pi_{Comm(x)}(d) = \mu_{low}(d)$ , $\forall d \in D$

where $\mu_{low}(d)$ is the membership function of the fuzzy set *low* defined on D (example in fig. 1)

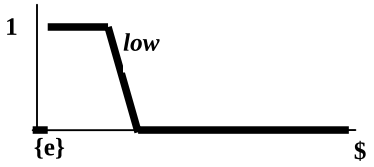

Fig. 1. Representation of the fuzzy set "*low*".

How the relation concept from the relational model is extended in the possibility theory framework as to include imprecise data?

The possibilistic relation is a subset of the cartesian product of possibility distributions sets. The possibilistic database is defined by a set of attributes $A_i$, a set of value domains $D_i$, one for each attribute, and a set of possibilistic relations $r_i$. The relational algebra operators are redefined in order to be able to work with possibilistic relations.

**STUDENT**

| Name | Age | Mark | Address |
|---|---|---|---|
| John | { 22,23 } | big | Galați |
| George | around 22 | 8 | {Galați,Iaşi } |
| Marie | {0.8/21, 1.0/22} | [ 6, 9 ] | 'unknown' |
| Paul | young | little | Iaşi |

Fig. 2. A possibilistic relation

Figure 2 shows an example for the content of a possibilistic relation, STUDENT.

Remark

Many approaches of data imprecision representation in databases are supported by the fuzzy set theory. Actually, Bosc and Prade (1997) show that in a database context, both fuzzy sets and possibility distribution benefit of the same representation (at least for the 7-th precedent point), but they suffer different interpretation, depending on the considered situation. In other words, "ill-known attribute values can be represented by means of fuzzy sets viewed as possibility distributions." (Dubois and Prade, 1996, pag. 90)

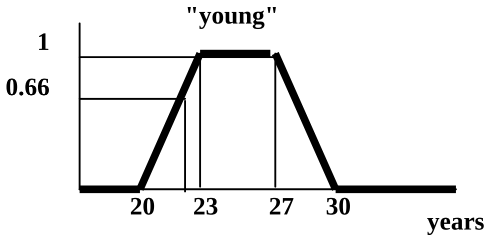

Fig. 3. Representation of the fuzzy set "young".

Figure 3 means that the value 22 is compatible with a 0.66 degree with the concept "young", represented by the membership function of a fuzzy set, i.e.

$$\mu_{young}(22) = 0.66$$

Let's suppose that Peter is recorded in database with age=22; a vague query that asks the "young" persons from the database will include Peter on the response list with 0.66 satisfaction degree of the criterion.

On the other hand, the same figure 3 means that it is possible at a 0.66 degree that 22 is the value of an ill-known attribute described by the possibility distribution.

Let's suppose that Peter is recorded in database with the label "young"; a precise query that asks the persons having age=22 from the database will include Peter in the answer, with a 0.66 degree of possibility, because

$$\pi_{Age(Peter)}(22) = 0.66$$

In conclusion

1) fuzzy sets represent gradual properties whose satisfaction may be a matter of degree and they are appropriate to be used in vague queries processing;
2) possibility distributions restrict the possible values of attributes in databases, so they are appropriate to be used in imprecise data representation in databases (imprecise databases).

Query evaluation in possibilistic relational database

When imprecise data are selected even by crisp (precise) condition, the result can not be certainly *true* or *false*. Intuitively, only a certainty degree can measure the possibility to satisfy the criterion by a certain relation tuple.

In the possibility theory, there are two indicators for the satisfaction degree measurement: *possibility* and *necessity degrees*.

Let D be the value domain for an attribute A and S a subset of D, considered as selection criterion. The query results set must be $\{x \mid x.\mathrm{A} \in \mathrm{S}\}$ .

In a possibilistic environment, for every tuple x there will be computed (Dubois and Prade 1994]):

1) the *possibility degree* that *x*.A ∈ S (or in which x is possible to satisfy the criterion S )

$$\Pi_x(S) = \sup_{d \in S} \pi_x(d)$$

2) the *necessity degree* that *x*.A ∈ S (or in which x certainly satisfies the criterion S )

$$N_x(S) = \inf_{d \notin S}(1 - \pi_x(d)) = 1 - \Pi_x(\overline{S}) \quad ,$$

where $\overline{S}$ is the complement of S in D.

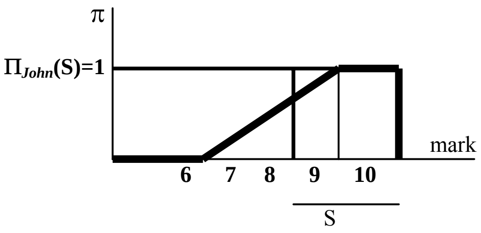

Fig. 4. The possibility measure attached to a tuple of a query result.

In figure 4 and 5 it's shown how these two measures are graphically computed, for a criterion (mark between 8 and 10, i.e. S=[8,10]) applied to a tuple corresponding to student John.

Starting from here, the literature offers some proposals to generalise relational algebra and aggregation operators for the possibilistic relational model (Prade and Testemale, 1984; Rundensteiner and Bic, 1991; etc).

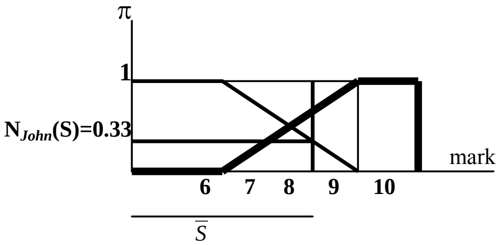

Fig. 5. The necessity measure attached to a tuple of a query result.

## 3. UNCERTAIN DATA IN RELATIONAL DATABASE

In order to take into account the uncertainty aspect of the information, the usual method is to attach it a subjective measure of its certitude. This has to reflect an estimation of the possibility that the information is true (or false). Many variants are possible, where this measure is expressed by:

1) *a number*; the probability theory, the possibility theory, the evidence theory or other techniques based on uncertainty degree, are usually used in order to properly interpret and manipulate such information;
2) *a symbol*; various deduction methods are used in order to obtain conclusions from uncertain information treatment.

In the following, two of the most important approaches, are presented; both consider a numerical measure for the information confidence.

- ❖ *Probability theory based approach*

The simplest idea when using probability to quantify uncertainty in relational database is to attach a probability to each tuple of the relation and to use it in order to estimate a probability that a certain query result is the real one.

Some studies have proved how the relational model must be extended in order to take into account probabilistic information and to be able to process queries in that context.

For example (fig. 6), the table STUDENT is modified by adding a now column *Pr* to record the probability degree for each tuple. So:

*Pr* [ STUDENT ('John','Sciences', 6,'Galați')] = 0.4

The probabilistic table includes all tuples that can be real, even with a minimal probability. Only those tuples, having null probability, are missing.

**STUDENT**

| Name | Department | Mark | Address | *Pr* |
|---|---|---|---|---|
| John | Sciences | 6 | Galați | 0.4 |
| John | Computers | 6 | Galați | 0.6 |
| George | Computers | 8 | Brăila | 1 |
| Mary | Sciences | 9 | Iaşi | 0.8 |
| Michael | Languages | 8 | Iaşi | 0.2 |
| Michael | Languages | 9 | Brăila | 0.5 |

Fig. 6. A probabilistic relation

Looking the table, we can conclude that John exists with certainty (the total probability is 1), but the department where he is studying is uncertainly known. The student George exists with certainty and all his attributes are well known.

For the tables where all attributes are certainly known (departments table, for example), the probability column (Pr) may be absent; in other words, the probability for each tuple will be 1, even if it isn't explicitly recorded.

D. Dey and S. Sarkar (1998) propose an uncertain extension of the relational model, using probabilities; it consists in: defining new kinds of structures, redefining relational algebra operators, defining a new nonprocedural language to access data (PSQL), reassessing database design techniques in the new probabilistic context.

<u>Query evaluation in probabilistic relational database</u>

The specific operation that takes into account the tuples probabilities is *coalesce operation*. It cumulates the probabilities of tuples having identical values for the same attribute.
For example:
Pr['John']=1;
Pr['Michael']=0.7;
Pr['Michael', 'Languages']=0.7.
This operation becomes very frequent, especially after projections. (Note: the projection always includes the primary key and additionally other non-key attributes)

Two examples for possible queries and their results are:
*What students may have the address in Braila?*
SELECT Name FROM Student WHERE
Address='Braila';
(*Answer*: 'George', 'Michael')

*What is the probability that Michael is studying in Languages?*
SELECT *Pr*[Name, Department] FROM Student
WHERE Name='Michael' AND
Department='Languages';
(*Answer*: 0.7)

- *Possibility theory based approach*

This approach allows a generalization of classical database relation, accepting various degrees of tuple-belonging-to-relation in (0,1], not too far from probabilities approach.
This kind of representation assumes that all attribute values are crisp, but the membership of a tuple to a relation is fuzzy.
A more interesting advantage of the possibility theory framework is to attach a possibility degree to every value of every attribute, that is possibility distributions (Prade H. and Testemale C., 1984). In this case, a particular query language must be able to compute the degree to which a certain tuple satisfies a certain condition.

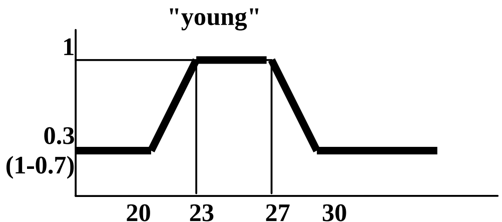

Fig. 7. Representation of the attribute fuzzy value "young" affected by a 0.7 uncertainty degree.

In a more complex situation, when imprecision and uncertainty are both present in databases, possibility theory is able to combine them during the query processing. In figure 7, the possibility distribution of a vague and uncertain value of an attribute is represented. For example, Peter's age, recorded as label "young" (fuzzy attribute value), has a 0.7 uncertainty degree.

## 4. CONCLUSIONS

The theoretical issues presented here have inspired a new project consisting in an intelligent interface to process flexible queries addressed to relational databases. It is in working in our laboratory.
The module that manages fuzzy terms, **FuzzyKAEE**, is a tool able to build fuzzy knowledge (fuzzy sets, fuzzy numbers, fuzzy connectives, fuzzy modifiers, fuzzy quantifiers), using a very friendly graphical interface. FuzzyKAEE is also able to evaluate fuzzy expressions as required by the user, in accordance with fuzzy terms previously defined. It works with respect to the fuzzy logic approach presented in the previous sections.
An other component of the system, **RoLQuery**, is a natural language (Romanian) interface for database querying. It is able to connect to any relational database for which a specific three-component knowledge base exists: lexicon, phrase translation rules base and database model. RoLQuery receives a natural language query, translates it in a SQL query, sends the query to the server and presents results to the user.
In the end, the entire system will be able to interpret and evaluate flexible queries, ensuring three major facilities: to record imprecise data in database, to formulate queries in natural language (Romanian) and to include vague terms in selection criteria.

## REFERENCES

Bosc, P. *et al.* (1995). *Rapport d'activité scientifique 1995. Project BADINS. Bases de données multimédia et interrogation souple*. Institut de recherche en informatique et systèmes aléatoires, Rennes.

Bosc, P. and H. Prade (1997). An Introduction to Fuzzy Set and Possibility Theory-based Approaches to the Treatment of Uncertainty and Imprecision in Data Base Management Systems. In *Uncertainty Management in Information Systems: from Needs to Solutions* (A. Motro and P. Smets, (Eds)), pp. 285-324, Kluwer Academic Publishers.

Codd, E.F. (1979). Extending the database relational model to capture more meaning, In *ACM Transaction on Database Systems*, **4**, pp 397-434

Dey, D. and Sarkar, S. (1998). PSQL: A query language for probabilistic relational data, In *Data & Knowledge Engineering*, **28**, pp. 107-120, Elsevier

Dobrzynski, W., J. Kacprzyk and S. Zadrozny (1997). An example of fuzzy querying using Microsoft's active server pages tools. In *EUFIT'97 Proceedings*, pp. 1181-1185, Aachen

Dubois, D. and Prade, H. (1988). *Théorie des possibilités. Applications à la représentation des connaissances en informatique*, MASSON, Paris

Dubois, D. and Prade, H., (1994). Possibility theory and data fusion in poorly informed environments, *Control Engineering Practice*, IFAC, **2(5)**, pp 811-823

Dubois, D. and Prade, H. (1996). *Using fuzzy sets in flexible querying : Why and how ?,* FQAS'96, pp. 89-103, Roskilde, Denmark

Kacprzyk, J. and S. Zadrozny (2001). Computing with words in intelligent database querying: standalone and Internet-based applications. *Information Sciences*, 134, pp.71-109, Elsevier

Liétard, L. (1995) *Contribution à l'interrogation flexible de bases de données - Etude des propositions quantifiées floues*. thèse, Université de Rennes I.

Lipski, W. (1979). On semantic issues connected with incomplete information databases, In *Proceedings of the International Conference on Knowledge Representation and Reasoning*, pp 245-255, Toronto

Motro, A. (1988). VAGUE :a user interface to relational databases that permits vague queries, *ACM Transaction on Office Information Systems*, **6**, pp. 187-214

Pivert, O. (1991) *Contribution à l'interrogation flexible de bases de données - Expression et évaluation de requêtes floues*. thèse, Université de Rennes I.

Prade, H. and Testemale, C. (1984). Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries, *Information Sciences*, **34**, pp. 115-143, Elsevier Science, NY

Rundensteiner, E. A., Bic, L. (1991), *Evaluating Aggregates in Possibilistic Relational* , NSF grant, January 1991

Tudorie, C., Segal, C., Bumbaru, S. (2001) Imperfect data in database context. Flexible queries, In *SIMSIS-11 Proceedings*, pp. 189-193, Galati

Zicari, R. (1993). Databases and Incomplete Information, In *Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems: From Needs to Solutions*, Catalina Island, CA