# The notion of H-IFS -an approach for enhancing query capabilities in Oracle10g.

**Panagiotis Chountas**[1]
**Ermir Rogova**[1]
**Sadiq Mohammed**[1]
**Krassimir Atanassov**[2]

[1] Harrow School of Computer Science, University of Westminster
[2] CLBME, Bulgarian Academy of Sciences

# The Notion of H-IFS -An Approach for Enhancing Query Capabilities in Oracle10g

Panagiotis Chountas, Ermir Rogova, Sadiq Mohammed, Krassimir Atanassov

***Abstract*—Query answering requirements for a knowledge based treatment of user requests led us to introduce the concept of closure of an Intuitionistic fuzzy set over a universe that has a hierarchical structure. We introduce the automatic analysis of queries according to concepts defined as part of a knowledge based hierarchy in order to guide the query answering as part of an integrated database environment with the aid of hierarchical Intuitionistic fuzzy sets, H-IFS. In this paper based on the notion of H-IFS we propose an ad-hoc utility build on top of Oracle10g that allows us to enhance the query capabilities of by providing better and knowledgeable answers to user's requests. The theoretical aspects as well the practical issues and achieved results are presented throughout the rest of the paper.**

***Index Terms*— Query answering Systems, Hierarchical Intuitionistic Fuzzy Sets, OLAP, Oracle10g**

## I. Introduction

Over the past years we have witnessed an increasing interest in expressing user or domain preferences[1] inside database queries. First, it appeared to be desirable property of a query system to offer more expressive query languages that can be more faithful to what a user intends to say. Second, a classical query in the sense of relational paradigm may also have a restricted answer or sometimes an empty set of answers, while a relaxed version of the query enhanced with background or domain knowledge might be matched by some items in the database.

Frequently integrated DBMSs contain incomplete data which we may represent using hierarchical background knowledge to declare support contained in subsets of the domain. These subsets may be represented in the database as partial values, which are derived from background knowledge using conceptual modelling to re-engineer the integrated DBMS.

Concerning query enlargement, several works such as [2] use a lattice of concepts to generalize unsolvable queries. An extended relational model for assigning possible values to an attribute value has been proposed by [3]. This approach may be used either to answer queries for decision making or for the extraction of answers and knowledge from relational databases. It is therefore important that appropriate functionality is provided for database systems to handle such information

In studies about possibilistic ontologies [4], each term of an ontology is considered as a linguistic label and has an associated fuzzy description. Fuzzy pattern matching between different ontologies is then computed using these fuzzy descriptions.

Studies about fuzzy thesauri have discussed different natures of relations between concepts. Fuzzy thesauri have been considered, for instance, in [5].

Recently in OLAP systems a need has been identified for enhancing the query scope with the aid of kind of relation that describe knowledge as well as ordering of the elements of a domain or a hierarchical universe.

However, in our context, the terms of the hierarchy and the relations between terms are not fuzzy. These observations led us to introduce the concept of closure of the H-IFS which is a developed form defined on the whole hierarchy. The definition domains of the Hierarchical Intuitionistic Fuzzy sets, H-IFS that we propose below are subsets of hierarchies composed of elements partially ordered by the "kind of" relation. Intuitively, in the closure of the H-IFS, the "kind of" relation is taken into account by propagating the degree associated with an element to its sub-elements more specific elements in the hierarchy.

Based on the above observations, in this research, we particularly focus on incorporating hierarchical preferences expressed in the form of background-domain knowledge with the aim on enhancing the query scope and in return to get richer answer, closer to user requests.

We developed an ad-hoc utility 'IF-Oracle' implemented on top of Oracle10g that allow us firstly to define and secondly incorporate hierarchical knowledge in the form of H-IFS as part of the standard SQL queries. We demonstrate the benefits of the 'IF-Oracle' by comparing the respective enhanced query answers against the Oracle10g standard query answers.

The rest of the paper is organised as follows; In Section II we define the basic properties of Intuitionistic Fuzzy sets and H-IFS. In Section III we define the extended SQL aggregators. In Section IV we present and discussed the main concepts involved in the designing and implementation the 'IF-Oracle' ad-hoc utility and also demonstrate the potential of 'IF-Oracle' utility when it comes to query answering that requires utilisation of the domain knowledge in order to receive answer close to the user's intent. Finally we conclude and we point to future research aims and targets.

P. Chountas is with the DKMG – HSCS, University of Westminster, Watford Road Northwick Park, London, HA1 3TP, UK (e-mail: chountp@wmin.ac.uk)
E. Rogova is with the DKMG – HSCS, University of Westminster, Watford Road Northwick Park, London, HA1 3TP, UK (e-mail: rogovae@wmin.ac.uk)
S. Mohammed is with DKMG – HSCS, University of Westminster, Watford Road Northwick Park, London, HA1 3TP, UK
K. Atanassov is with CLBME, Bulgarian Academy of Sciences, Bl. 105, Sofia-1113, BULGARIA (e-mail: krat@argo.bas.bg)

978-1-4244-1739-1/08/$25.00 © 2008 IEEE 

## II. INTUITIONISTIC FUZZY LOGIC-NOTION OF H-IFS

Each element of an Intuitionistic fuzzy [6, 7] set has degrees of membership or truth ($\mu$) and non-membership or falsity ($\nu$), which don't sum up to 1.0 thus leaving a degree of hesitation margin ($\pi$).

As opposed to the classical definition of a fuzzy set given by $A' = \{< x, \mu_{A'}(x) > | x \in X\}$ where $\mu_A(x) \in [0, 1]$ is the membership function of the fuzzy set A′, an Intuitionistic fuzzy set A is given by

$$A = \{< x, \mu_A(x), \nu_A(x) > | x \in X\}$$

$$\mu_A : X \rightarrow [0, 1] \text{ and } \nu_A : X \rightarrow [0, 1]$$

such that $0< \mu_A(x) + \nu_A(x)<1$ and $\mu_A(x)\ \nu_A(x) \in [0, 1]$ denote a degree of membership and a degree of non-membership of x ε A, respectively. Obviously, each fuzzy set may be represented by the following Intuitionistic fuzzy set

$A=\{<x, \mu_A'(x), (x), 1- \mu_A'(x)>|x \in X\}$

For each Intuitionistic fuzzy set in X, we will call $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ an Intuitionistic fuzzy index (or a hesitation margin) of $x \in A$ which expresses a lack of knowledge of whether x belongs to A or not. For each x ε A $0<\pi_A(x)<1$.

**Definition 1.** Let A and B be two intuitionistic fuzzy sets defined on a domain X. A is included in B (denoted $A \subseteq B$) if and only if their membership functions and non-membership functions satisfy the condition:

$$(\forall x \in X)\ \ (\mu A(x) \leq \mu B(x)\ \ \&\ \ \nu A(x) \geq \nu B(x))$$

**Definition 2**. Let Q and D be two intuitionistic fuzzy sets defined on a domain X and representing, respectively, a flexible query and an ill-known datum:

Two scalar measures are classically used in classical fuzzy pattern matching to evaluate the compatibility between an ill-known datum and a flexible query, known as

- a possibility degree of matching, **Π(Q; D)**
- a necessity degree of matching, **N(Q/ D)**

The possibility degree of matching between Q and D, denoted Π(Q; D), is an "optimistic" degree of overlapping that measures the maximum compatibility between Q and D, and is defined by:

$$\Pi(Q/D) = <\sup_{x \in X} (\min(1-\nu_Q(x), 1-\nu_D(x)),$$
$$\inf_{x \in X} (\max(\nu_Q(x), \nu_D(x))>,$$

The necessity degree of matching between Q and D, denoted N(Q; D), is a "pessimistic" degree of inclusion that estimates the extent to which it is certain that D is compatible with Q, and is defined by:

$$N(Q/D) = <\inf_{x \in X} (\max(\mu_Q(x), \mu_D(x)),$$
$$\sup_{x \in X} (\min(1-\mu_Q(x), 1- \mu_D(x))>$$

### *A. H-IFS*

The definition domains of the hierarchical fuzzy sets [8, 9, 10] that we propose below are subsets of hierarchies composed of elements partially ordered by the "kind of" relation. An element $l_i$ is more general than an element $l_j$ (denoted $l_i \sim l_j$), if $l_i$ is a predecessor of $l_j$ in the partial order induced by the "kind of" relation of the hierarchy. An example of such a hierarchy is given in Fig. 3. A hierarchical intuitionistic fuzzy set is then defined as follows.

**Definition 3.** Let F be a H-IFS defined on a subset D of the elements of a hierarchy L. It degree is denoted as $<\mu, \nu>$. The closure of F, denoted clos(F), is a H-IFS defined on the whole set of elements of L and its degree $<\mu, \nu>_{clos(F)}$ is defined as follows.

For each element **l** of L, let $S_L = \{l_1, \ldots, l_n\}$ be the set of the smallest super-elements in D.

**If $S_L$ is not empty,**

$$<\mu, \nu>_{clos(F)}(S_L) = <\max_{1\leq i\leq n}(\mu(L_i)), \min_{1\leq i\leq n}(\nu(L_i))>$$

**else, $<\mu, \nu>_{clos(F)}(S_L) = <0, 0>$**

In other words, the closure of a H-IFS F is built according to the following rules. For each element $l_I$ of L:

- If $l_I$ belongs to F, then $l_I$ keeps the same degree in the closure of F (case where $S_L = \{ l_I \}$).
- If $l_I$ has a unique smallest super-element $l_I$ in F, then the degree associated with $l_I$ is propagated to L in the closure of F, $S_L = \{ l_I \}$ with $l_{I} > l_I$)

If L has several smallest super-elements $\{l_I, \ldots, l_n\}$ in F, with different degrees, a choice has to be made concerning the degree that will be associated with $l_I$ in the closure. The proposition put forward in definition 3, consists of choosing the maximum degree of validity μ and minimum degree of non validity v associated with $\{l_I, \ldots, l_n\}$. We refer to as the *Optimistic strategy*.

We can also utilise a *Pessimistic strategy* which consists of choosing the minimum degree of validity μ and maximum degree of non validity v associated with $\{l_I, \ldots, l_n\}$.

**If $S_L$ is not empty,**

$$<\mu, \nu>_{clos(F)}(S_L) = <\min_{1\leq i\leq n}(\mu(L_i)), \max_{1\leq i\leq n}(\nu(L_i))>$$

**else, $<\mu, \nu>_{clos(F)}(S_L) = <0, 0>$**

Alternatively, an *Average strategy* could be utilised, which consists of calculating the IF-Average and applying it to the degrees of validity μ and non-validity v.

It has been observed that two different H-IFSs, defined on the same hierarchy, can have the same closure, as in the following example.

**Example.** The H-IFSs Q={Wine<1,0>, Red Wine<0.7,0.1>, Brown Wine<1,0>, White Wine <0.4,0.3>} and

R ={Wine<1,0>, Red Wine<0.7,0.1>, Brown Wine<1,0>, Pinot Noir <0.4,0.3>} have the same closure, represented Fig.1 below.

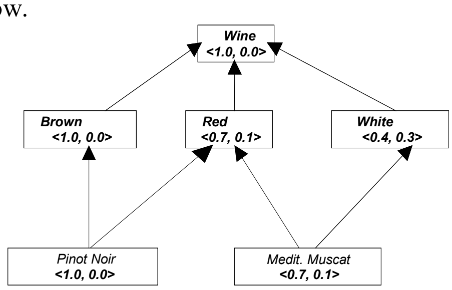



Fig.1. Common closure of the H-IFSs Q and R

Such H-IFSs form equivalence classes with respect to their closures.

**Definition 4.** Two H-IFSs Q and R, defined on the same hierarchy, are said to be equivalent Q≡R if and only if they have the same closure.

**Property** Let Q and R be two equivalent Intuitionistic hierarchical fuzzy sets. If $l_I \in dom(Q) \cap dom(R)$, then $<\mu,\nu>(Q.l_I) = <\mu,\nu>(R.l_I)$

**Proof** According to the definition of the closure of a H-IFS F, definition 3, the closure of F preserves the degrees that are specified in F. As Q and R have the same closure (by definition of the equivalence), an element that belongs to Q and R necessarily has the same degree <μ,ν> in both.

We can note that R contains the same element as Q with the same <μ,ν>, and also one more element Pinot Noir<1,0>. The <μ,ν> associated with this additional element is the same as in the closure of Q. Then it can be said that the element, Pinot Noir<1,0> is derivable in R through Q.

The same conclusions can be drawn in the case of Medit. Muscat <0.7, 0.1>

**Definition 5.** Let F be a hierarchical fuzzy set, with dom(F) = $\{l_1, \ldots, l_n\}$, and $F_{-k}$ the H-IFS resulting from the restriction of F to the domain dom(F) \ $\{l_k\}$. $l_k$ is deducible in F if

$$<\mu, \nu>clos_{(F\text{-}k)}\ (l_k) = <\mu, \nu>clos_{(F)}\ (l_k)$$

As a first intuition, it can be said that removing a derivable element from a hierarchical fuzzy set allows one to eliminate redundant information. But, an element being derivable in F does not necessarily mean that removing it from F will have no consequence on the closure: removing k from F will not impact the degree associated with k itself in the closure, but it may impact the degrees of the sub-elements of k in the closure.

For instance, if the element Brown Wine is derivable in Q, according to definition 5, removing Brown Wine <1,0> from Q would not modify the degree of Brown Wine itself in the resulting closure, but it could modify the degree of its sub-element Pinot Noir. Thus, Brown Wine <1,0> cannot be derived or removed. This remark leads us to the following definition of a minimal hierarchical fuzzy set.

**Definition 6.** In a given equivalence class (that is, for a given closure C), a hierarchical fuzzy set is said to be **minimal** if its closure is C and if none of the elements of its domain is derivable**.**

*B. Obtaining the Minimal H-IFS*

*Step 1:* Assign Min-H-IFS ← ∅. Establish an order so that the sub-elements $\{l_1, \ldots, l_n\}$ of the hierarchy L are examined after its super-elements.

*Step 2:* Let $l_1$ be the first element and $\mathbf{(l_1)/<\mu, \nu>} \neq \mathbf{(l_1)/<0, 0>}$ then add $\mathbf{l_1}$ to Min-H-IFS and $\mathbf{<\mu, \nu>_{clos(Min\text{-}HIFS)}\ (l_1) = (l_1)/<\mu, \nu>}$.

*Step 3:* Let us assume that K elements of the hierarchy L satisfy the condition $\mathbf{<\mu, \nu>_{clos(Min\text{-}HIFS)}\ (l_i) = (l_i)/<\mu, \nu>}$. In this case the Min-H-IFS do not change. Otherwise go to next element $\mathbf{l_{k+1}}$ and execute Step 4.

*Step 4:* The $\mathbf{l_{k+1}/<\mu_{k+1}, \nu_{k+1}>}$ associated with $\mathbf{l_{k+1}}$. In this case $\mathbf{l_{k+1}}$ is added to Min-H-IFS with the corresponding $\mathbf{<\mu_{k+1}, \nu_{k+1}>}$.

Step 5: Repeat steps three and four until $\mathbf{clos_{(Min\text{-}HIFS)} = C}$.

For instance the H-IFSs $S_1$ and $S_2$ are **minimal** (none of their elements is derivable). They cannot be reduced further.

$S_1$= Wine<1,0>
$S_2$={Wine<1,0>, Red Wine<0.7,0.1>, Pinot Noir<1,0>, White Wine <0.4, 0.3>}

*C. Representing H-IFS as concept relations*

The structure of any H-IFS can be described by a domain concept relation DCR = (Concept, Element), where each tuple describes a relation between elements of the domain on different levels. The DCR can be used in calculating recursively [14] the different summarisation or selection paths as follows:

$$PATH \leftarrow DCR_{\{x=1\ldots(n-2)\,|\,n>2\}} \bowtie DCR_x$$

If n≤2, then DCR becomes the Path table as it describes all summarisation and selection paths. These are entries to a knowledge table that holds the metadata on parent-child relationships. An example is presented below

TABLE 1. DOMAIN CONCEPT RELATION

| DCR | |
|---|---|
| **Concept** | **Element** |
| **Wine <1.0, 0.0>** | Brown Wine <1.0, 0.0> |
| **Wine <1.0, 0.0>** | Red Wine <0.7, 0.1> |
| **Wine <1.0, 0.0>** | White Wine <0.4, 0.3> |
| **Brown Wine <1.0, 0.0>** | Pinot Noir <1.0, 0.0> |
| **Red Wine <0.7, 0.1>** | Pinot Noir <1.0, 0.0> |
| **Red Wine <0.7, 0.1>** | Medit. Muscat <0.7, 0.1> |
| **White Wine <0.4, 0.3>** | Medit. Muscat <0.7, 0.1> |

Table 1 shows how our Wine hierarchy knowledge table is kept. Paths are created by running a recursive query that reflects the 'PATH' algebraic statement. The hierarchical IFS used as example throughout this paper comprises of 3 levels, thus calling for the SQL-like query as below:

*SELECT A.Concept as Grand-concept, b.concept, b.element*
*FROM DCR as A, DCR as B*
*WHERE A.child=B.parent;*

This query will produce the following paths:

TABLE 2.PATH TABLE

| Path | | | |
|---|---|---|---|
| **Grand-concept** | **Concept** | **Element** | **Path Colour** |
| **Wine <1.0, 0.0>** | Brown Wine <1.0, 0.0> | Pinot Noir <1.0, 0.0> | Red |
| **Wine <1.0, 0.0>** | Red Wine <0.7, 0.1> | Pinot Noir <1.0, 0.0> | Blue |
| **Wine <1.0, 0.0>** | Red Wine <0.7, 0.1> | Medit. Muscat <0.7, 0.1> | Green |
| **Wine <1.0, 0.0>** | White Wine <1.0, 0.0> | Medit. Muscat <0.7, 0.1> | Brown |

Fig. 2 presents a pictorial view of the four distinct summarisation and selection paths. These paths will be used in fuzzy queries to extract answers that could be either definite or possible. This will be realised with the aid of the predicate (θ).A predicate (θ) involves a set of atomic predicates ($\theta_1, \ldots, \theta_n$) associated with the aid of logical operators *p* ( i.e. ∧, ∨, etc.). Consider a predicate θ that takes the value "Red Wine", θ = "Red Wine".

After utilizing the IFS hierarchy presented in Fig.7, this predicate can be reconstructed as follows:

$$\theta = \theta_1 \vee \theta_2 \vee \ldots \vee \theta_n$$

In our example, $\theta_1$="Red Wine", $\theta_2$="Pinot Noir" and $\theta_n$="Medit. Muscat". The reconstructed predicate θ = (Red Wine ∨ Pinot Noir ∨ Medit. Muscat) allows the query mechanism to not only definite answers, but also possible answers [11].

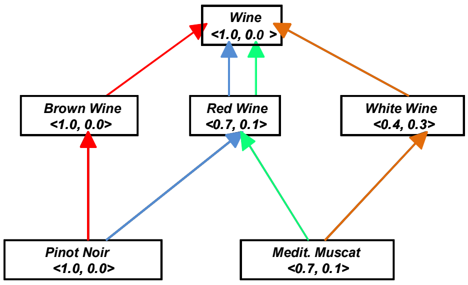



Fig. 2. Pictorial representation of paths

In terms a query retrieving data from a summary table, the output contains not only records that match the initial condition, but also those that satisfy the reconstructed predicate. Consider the case where no records satisfy the initial condition (Red Wine). Traditional aggregation query would have returned no answer, however, based on our approach, the extended query would even in this case, return an answer, though only a possible one, with a specific belief and disbelief $<\mu, \nu>$ . It will point to those records that satisfy the reconstructed predicate θ, more specifically, "Pinot Noir and Medit. Muscat".

Following the representation of H-IFS as concept relations and the definition of summarisation paths, there is still a need to extend the traditional aggregation operators in order to cope with flexible hierarchies of data organisations.

## III. EXTENDED RELATIONAL AGGREGATION OPERATORS

**Aggregation *(A):*** An aggregation operator *A* is a function *A(G)* where $G = \{<x, \mu_F(x), \nu_F(x)> | x \in X\}$ where $x=<att_1, ..., att_n>$ is an ordered tuple belonging to a given universe *X*, $\{att_1, ..., att_n\}$ is the set of attributes of the elements of *X*, $\mu_F(x)$ and $\nu_F(x)$ are the degree of membership and non-membership of *x*. The result is a bag of the type $\{<x', \mu_F(x'), \nu_F(x')> | x' \in X\}$. To this extent, the bag is a group of elements that can be duplicated and each one has a degree of μ and ν.

- *Input:* $R_i = (l, F, H)$ and the function *A(G)*
- *Output:* $R_o = (l_o, F_o, H_o)$ *where*
- *l* is a set of levels $l_1, ..., l_n$, that belong to a partial order $\leq O$ To identify the level *l* as part of a hierarchy we use *dl.*
- $l_\perp$: base level, $l_\top$: top level

For each pair of levels $l_i$ and $l_j$ we have the relation

$$\mu_{ij} : l_i \times l_j \rightarrow [0,1], \quad \nu_{ij} : l_i \times l_j \rightarrow [0,1] \quad 0 < \mu_{ij} + \nu_{ij} < 1$$

- *F* is a set of fact instances with schema $F = \{<x, \mu_F(x), \nu_F(x)> | x \in X\}$, where $x=<att_1, ..., att_n>$ is an ordered tuple belonging to a given universe *X*, $\mu_F(x)$ and $\nu_F(x)$ are the degree of membership and non-membership of *x* in the fact table *F* respectively.
- *H* is an object type history that corresponds to a structure*( l, F, H')* which allows us to trace back the evolution of a structure after performing a set of operators i.e. aggregation

The definition of the extended group operators allows us to define the extended group operators ***Roll up (Δ), and Roll Down (Ω).***

**Roll up (Δ):**The result of applying Roll up over dimension $d_i$ at level $dl_r$ using the aggregation operator A over a relation $R_i=(l_i, F_i, H_i)$ is another relation $R_o=(l_o, F_o, H_o)$

*Input:* $R_i = (l_i, F_i, H_i)$

*Output:* $R_o = (l_o, F_o, H_o)$

An object of type history is a recursive structure:

- ω is the initial state of the relation.
- *(l, A, H')* is the state of the relation after performing an operation on it.

The structured history of the relation allows us to keep all the information when applying *Roll up* and get it all back when *Roll Down* is performed. To be able to apply the operation of *Roll Up* we need to make use of the $IF_{SUM}$ aggregation operator.

***Roll Down (Ω):*** This operator performs the opposite function of the *Roll Up* operator. It is used to roll down from the higher levels of the hierarchy with a greater degree of generalization, to the leaves with the greater degree of precision. The result of applying *Roll Down* over a relation $R_i = (l, F, H)$ having $H=(l', A', H')$ is another relation $R_o= (l', F', H')$.

*Input:* $R_i=(l, F, H)$

*Output:* $R_o=(l', F', H')$ where *F' → set of fact instances defined by operator A.*

To this extent, the *Roll Down* operative makes use of the recursive history structure previously created after performing the *Roll Up* operator.

The definition of aggregation operator points to the need of defining the IF extensions for traditional group operators [12], such as *SUM, AVG, MIN and MAX.* Based on the standard group operators, we provide their IF extensions and meaning.

***$IF_{SUM}$:*** The $IF_{sum}$ aggregate, like its standard counterpart, is only defined for numeric domains. The relation *R* consists of tuples $R_i$ *with* $1 \leq i \leq m$. The tuples $R_i$ are assumed to take Intuitionistic Fuzzy values for the attribute $att_{n-1}$ for $i = 1$ *to* $m$ we have $R_i[att_{n-1}] = \{<\mu_i(u_{ki}), \nu_i(u_{ki})> / u_{ki} \mid 1 \leq k_i \leq n\}$ . The $IF_{sum}$ of the attribute $att_{n-1}$ of the relation R is defined by:

$$IF_{SUM}((att_{n-1})(R)) = \{<u>/y \mid (\forall_{k1, ...km} : 1 \leq k1, ...km \leq n)\ ((u=(\min_{i=1}^{m} \mu_i(u_{ki}), \max_{i=1}^{m} \nu_i(u_{ki}))) \wedge (y = \sum_{ki=k1}^{km} u_{ki}))\}$$

***$IF_{AVG}$* :** The $IF_{AVG}$ aggregate, like its standard counterpart, is only defined for numeric domains. This aggregate makes use of the $IF_{SUM}$ that was discussed previously and the standard *COUNT.* The $IF_{AVG}$ can be defined as:

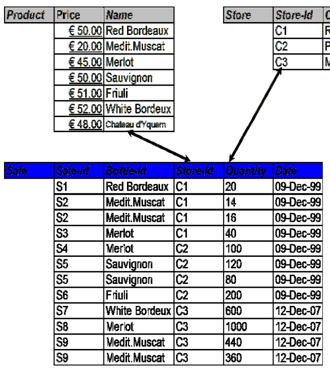
$$IF_{AVG}((att_{n-1})(R)) = IF_{SUM}((att_{n-1})(R)) \,/\, COUNT((att_{n-1})(R))$$

In the next section we demonstrate the usefulness of the H-IFS notion and the extended aggregation operators for extending the query capabilities of Oracle10g. We developed an ad-hoc utility 'IF-Oracle' implemented on top of Oracle10g that allow us to

- Define an H-IFS hierarchy
- Incorporate hierarchical knowledge in the form of H-IFS as part of the standard SQL queries.
- Enhance the scope of query answers against the Oracle10g standard query answers.

## IV. THE IF-ORACLE AN H-IFS BASED AD HOC UTILITY

IF-Oracle has been developed using Visual Studio.Net as an ad-hoc utility that is attached to and enhances Oracle10g DBMS query capabilities. For demonstrating the functionality of IF-Oracle let us consider a sample multidimensional model, (Fig.3) in the form of a star schema that describes sales of Vitis Vinifera type wines.



| Product | Price | Name |
|---|---|---|
| | € 50.00 | Red Bordeaux |
| | € 20.00 | Medit.Muscat |
| | € 45.00 | Merlot |
| | € 50.00 | Sauvignon |
| | € 51.00 | Friuli |
| | € 52.00 | White Bordeux |
| | € 48.00 | Chateau d'Yquem |

| Store | Store-Id | City |
|---|---|---|
| | C1 | Rome |
| | C2 | Paris |
| | C3 | Moscow |

| Sale | Sale-id | Bottle-Id | Store-Id | Quantity | Date |
|---|---|---|---|---|---|
| | S1 | Red Bordeaux | C1 | 20 | 09-Dec-99 |
| | S2 | Medit.Muscat | C1 | 14 | 09-Dec-99 |
| | S2 | Medit.Muscat | C1 | 16 | 09-Dec-99 |
| | S3 | Merlot | C1 | 40 | 09-Dec-99 |
| | S4 | Merlot | C2 | 100 | 09-Dec-99 |
| | S5 | Sauvignon | C2 | 120 | 09-Dec-99 |
| | S5 | Sauvignon | C2 | 80 | 09-Dec-99 |
| | S6 | Friuli | C2 | 200 | 09-Dec-99 |
| | S7 | White Bordeux | C3 | 600 | 12-Dec-07 |
| | S8 | Merlot | C3 | 1000 | 12-Dec-07 |
| | S9 | Medit.Muscat | C3 | 440 | 12-Dec-07 |
| | S9 | Medit.Muscat | C3 | 360 | 12-Dec-07 |

Fig. 3. Sample of a Star Schema

Figure 4 shows a sub-hierarchy that has been derived from the Vitis Vinifera domain for testing purposes. On the left it is shown the tree structure view as displayed in IF-Oracle, while on the right we have shown the tree representation.

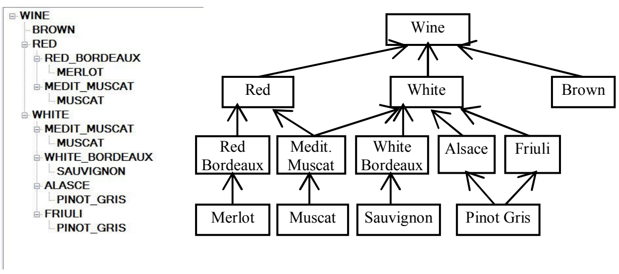



Fig. 4. Vitis Vinifera sub-hierarchy views

After forming the structure and storing it as a concept relation in Oracle10g, we perform the calculation of the hierarchical closure of the H-IFS and its weights.

The user now has the choice of selecting three different strategies: ***Optimistic, Pessimistic*** or ***Average*** as defined on section II paragraph A.

Let's assume that the user's interest lays on finding information about Red, White and Brown wines.

Figure 5 below shows the hierarchy after weights have been calculated and assigned reflecting the user's intent.

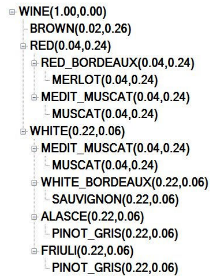



Fig. 5. Vitis Vinifera sub-hierarchy view with weights

We can observe that the principle of the H-IFS closure *(see definition 3)* has been preserved when propagating the degree of validity μ and non-validity ν from super-elements to sub-elements by using the optimistic strategy.

The degree of validity and non-validity <μ, ν > are calculated as follows:

$$\mu = \frac{|c_l|}{|c_{l-1}|} \qquad \nu = \frac{|\neg c_l|}{|c_{l-1}|}$$

Where $c_l$ corresponds to those elements from the fact table that absolutely satisfy the selection criteria with reference to a node in the hierarchy. $C_{l-1}$ represents the elements children elements of that selection on a lower level that satisfy the selection condition to some extent. It is obvious that

$$\pi = 1 - (\mu + \nu)$$

After adding the hierarchy into the repository and automatically calculating the weights for the requested nodes, the user can utilize the ad-hoc interface for execution of queries either in standard SQL or make use of the enhanced Select clause and features that IF-Oracle provides.

Figure 6 shows the results of a user request for "Red" wine executed in standard SQL provided by Oracle10g.

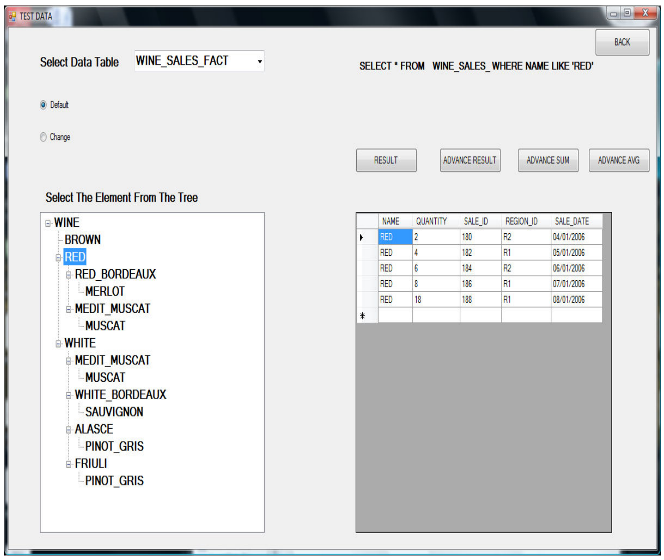



Fig. 6. Standard SQL output for "Red" wine

In contrast, figure 7 shows the output after executing the same query, but this time using the IF-Oracle utility.

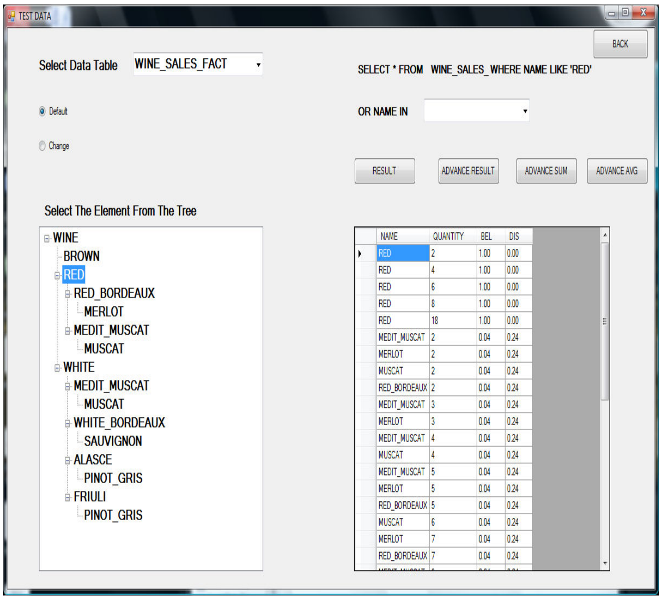



Fig. 7. Enhanced SQL output for "Red" wine

By comparing the two figures, one can observe that IF-Oracle produces a knowledge-based answer instead of mindlessly matching the records against the word "Red".

The results show that IF-Oracle not only retrieves sales of "Red" bottles, but also sales of bottles that are classified as red wines by the knowledge represented in the H-IFS hierarchy as "Merlot", "Red Bordeaux", "Medit. Muscat", etc. with indicative degrees of $<\mu, \nu>$ relevant to the user's preference.

## V. CONCLUSIONS

In this paper, we focus on integrating hierarchical preferences expressed in the form of background-domain knowledge with the aim on enhancing the query scope and in return to get richer answer, closer to user requests. We provide the means of using background knowledge to re-engineer query processing and answering with the aid of H-IFS and Intuitionistic Fuzzy relational representation.

The hierarchical links defined on the basis of the H-IFS closure are representing knowledge in the form of enhanced "kind of, $\leq$" relation. The membership of an element in a H-IFS has consequences on the membership and non-membership of its sub elements in this set.

We demonstrated the simplicity and implement-ability of the H-IFS notion by adding an ad-hoc utility 'IF-Oracle' in Oracle10g that allow us to enrich the scope of query and receive answers closer to user's intent and preferences even when answers are not obvious when using the standard SQL provided by Oracle10g.

Future research efforts will concentrate on incorporating knowledge arriving from external sources either semi structured or unstructured i.e. WordNet or Wikipedia considering the web as such as source. Furthermore considering the notion of the minimal H-IFS one could consider to devise new optimisation techniques for making query processing more efficient

## REFERENCES

[1] Slawomir Zadrozny, Janusz Kacprzyk: Bipolar Queries and Queries with Preferences. *DEXA Workshops 2006*: 415-419

[2] S. Rice, J. F. Roddick, "Lattice-Structured Domains, Imperfect Data and Inductive Queries", *LNCS, DEXA*, 2000, pp. 664-674

[3] D. Bell, J. Guan, S. Lee "Generalized union and project operations for pooling uncertain and imprecise information". *DKE 18* (1996), pp.89-117

[4] G. Pasi, F. Crestani, "Evaluation of Term-Based Queries Using Possibilistic Ontologies," *Soft Computing for Information Retrieval on the Web, Springer-Verlag, 2005*.

[5] S. Miyamoto, K. Nakayama, "Fuzzy Information Retrieval Based on a Fuzzy Pseudothesaurus," *IEEE Trans. Systems, Man and Cybernetics, 1986,* vol. 16, no. 2, pp. 278-282

[6] K. Atanassov „Intuitionistic Fuzzy Sets", *Heidelberg, Springer-Verlag, 1999.*

[7] K. Atanassov "Remarks on the Intuitionistic fuzzy sets". – *Fuzzy Sets and Systems, vol. 51, 1992*, No 1, 117-118.

[8] E. Rogova, P. Chountas, K. Atanassov "Flexible hierarchies and fuzzy knowledge-based OLAP". – *FSKD 2007, IEEE Computer Society Press* vol.2, pp. 7-11

[9] P. Chountas "On intuitionistic fuzzy sets over universes with hierarchical structures". – *Notes on Intuitionistic Fuzzy Sets, vol. 13, 2007,* No 1, 52-56.

[10] E. Rogova, P. Chountas "On imprecision intuitionistic fuzzy sets & OLAP – The case for KNOLAP*", IFSA 2007, Springer-Verlag ISBN: 3540724338, 2007*, pp.11-23

*[11]* [14] A. Silberschatz, H. Korth, S. Sudarshan " Database System Concepts". *McGraw-Hill*

[12] E. Rundensteiner L. Bic. "Aggregates in posibilistic databases", *VLDB'89,* pp. 287-295, 1989