



**HAL**  
open science

# A Conceptual Model for Handling Personalized Hierarchies in Multidimensional Databases

Yoann Pitarch, Anne Laurent, Pascal Poncelet

► **To cite this version:**

Yoann Pitarch, Anne Laurent, Pascal Poncelet. A Conceptual Model for Handling Personalized Hierarchies in Multidimensional Databases. MEDES: Management of Emergent Digital EcoSystems, Oct 2009, Lyon, France. 10.1145/1643823.1643843 . lirmm-00426501

**HAL Id: lirmm-00426501**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00426501>**

Submitted on 18 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Conceptual Model for Handling Personalized Hierarchies in Multidimensional Databases

Yoann Pitarch  
LIRMM  
University Montpellier 2 -  
CNRS  
Montpellier, France  
pitarch@lirmm.fr

Anne Laurent  
LIRMM  
University Montpellier 2 -  
CNRS  
Montpellier, France  
laurent@lirmm.fr

Pascal Poncelet  
LIRMM  
University Montpellier 2 -  
CNRS  
Montpellier, France  
poncelet@lirmm.fr

## ABSTRACT

Hierarchies are extensively used in data warehouses, OLAP systems and more recently in data stream summarization systems. They indeed allow decision makers to consider information at multiple granularity levels and they enable efficient compression mechanisms. However, even if numerous models of hierarchies have been proposed, some hierarchies arising in real-world situations are still not manageable by the current systems. For instance, in medical applications, determining the *normality* of an arterial pressure measure is infeasible without considering the patient's characteristics (e.g. age). In this paper, we thus propose to model such context-dependent hierarchies by introducing *Personalized Hierarchies*. Firstly, we motivate this new category by presenting the lacks of existing approaches and we propose a conceptual model for modeling personalized hierarchies. Finally, a first logical model for handling such context-dependent hierarchies is proposed.

## Keywords

Data Warehouses, OLAP, Hierarchies

## 1. INTRODUCTION

Nowadays decision makers extensively use data analysis in order to be more and more competitive. Data warehouses are particularly well adapted for this purpose. Defined as “a collection of subject-oriented, integrated, non-volatile, and time-variant data to support management's decisions” [2], data warehouses offer a multidimensional view of data. Their structure is usually modeled using a *star* or a *snowflake* schema. Attributes representing the elements of analysis are called *measures*. These measures can be explored and analyzed from different perspectives thanks to the *dimensions*. A *dimension* can either be descriptive (e.g., smoker or not) or can form a hierarchy. The main advantage of the hierarchies is that data can be observed at different levels of granularity. Thus, users can navigate from a general view

of data to a more detailed view thanks to the drill-down operator. Inversely, the roll-up operator allows to observe detailed measure at a higher level of granularity.

Recently, the concept of hierarchy has been employed for multidimensional data stream summarization. Indeed, more and more multidimensional data are daily generated (web logs, sensor networks, ...) and it is impossible to store the whole history of such streams. Methods are thus proposed in order to provide decision makers with a bounded and representative summary of the stream. Among the different techniques employed, two approaches [1, 5] exploit the dimension hierarchies according the following principle: the older the data, the coarser their storage level.

Although hierarchies are very powerful and useful, their categorization is still an open-problem. Moreover, current OLAP or data stream summarization tools manage a restrictive subset of existing types of hierarchies in comparison with real-world situations. Moreover, traditional data cube definitions do not always easily handle hierarchies on measures. This leads to a lack of expressivity. To illustrate these arguments, we rely on the three following scenarios.

**SCENARIO 1 (THE PURCHASE ANALYSIS).** *Let us consider the classical situation of the customer market basket analysis. A typical query could be “Who bought a lot of surfboards during the last week?” Here, the difficulty is to exactly define what ‘a lot of’ means. In fact, this concept clearly depends on several parameters such as the temporal period, the considered product, the store's location... It should be noted that current systems do not enable to manually define such informations.*

**SCENARIO 2 (SENSOR NETWORK).** *Nowadays, sensor networks are very common. Most of sensors send physical measures to a centralized site which summarizes the incoming data. Let us suppose that sensors send intern temperature data of hydraulic pumps. A supervisor system would like to classify these incoming values according a normality degree. The matter is that this classification is not static since these measures depend on some exterior parameters such as the external temperature, the altitude, ... As a consequence, considering a unique scale of normality can lead to critical mistakes in monitoring, alarm detection or stream summarization systems.*

SCENARIO 3 (MEDICAL EQUIPMENT). *Nowadays, more and more hospitals use medical information systems. Some of them summarize patient vital parameter streams in order to perform medical alert detection. However, some vital parameters (arterial pressure, heart rate, ...) are highly correlated with some physiologic human characteristics. For instance, the average heart rate depends on the patient's age. Table 3 presents the children heart rate average depending to the age of the children [6]. Another characteristic which impacts a lot the heart rate is the global physic condition. Indeed, the average heart rate for athletic men is about 60 bpm at rest. A tachycardia crisis detection system must be aware of such knowledge in order to be efficient.*

Age Group	Heart Rate
New born	140
6 months	130
1 year	115
2 years	110
6 years	103
8 years	100
10 years	95

**Table 1: Children's heart rate**

*The arterial pressure also depends on some human characteristics. For instance, smoker's arterial pressure dramatically increases just after they have smoked a cigaret. The age of the patient is also a parameter which impacts a lot the arterial pressure (globally the arterial pressure of an  $X$ -year-old person is  $100 + X$ ). Table 3 presents the arterial pressure of some patients presenting different characteristics. Generally, a normal arterial pressure is considered between 13 and 15. With such a general knowledge, none of the patient has a normal arterial pressure (i.e., the patients 1 and 2 would be considered as suffering from hypotension and the patients 3 and 4 would be consider as suffering from hypertension). Nevertheless, according to the above-mentionned specific knowledge, we conclude that the patients 1 and 4 have a normal arterial pressure regarding theirs characteristics.*

IdPatient	Age	Smoker	Arterial pressure
1	5	No	10
2	25	Yes	10
3	40	Yes	17
4	70	No	17

**Table 2: Arterial pressure of different patients**

As presented in these scenarios, many hierarchies defined on measures are context-dependent in real-world situations. Thus the roll-up and the drill-down operators must take external characteristics into account. Even if these scenarios are realistic and often occur, the literature does not provide any formal model for this kind of hierarchy.

In this paper, we thus tackle this lack by introducing a new category of hierarchy: *personalized hierarchies*. In this study, we take Scenario 3 as a typical scenario in order to illustrate the proposed model.

The rest of this paper is organized as follows. Section 2 presents the main existing categories of hierarchies. In Section 3, the personalized hierarchies are defined. In Section 4, we propose a first logical model for handling such personalized hierarchies. Finally, we conclude and draw up some perspectives in Section 5.

## 2. A BRIEF OVERVIEW OF THE EXISTING CATEGORIES OF HIERARCHIES

We recall here that we consider the multidimensional framework. In this respect, dimensions are defined, some of them being of particular interest: the measures. For instance, the measure *number of sales* is considered and analyzed with respect to dimensions such as *city*, *product*, *month*.

DEFINITION 1 (DIMENSION). *A dimension  $D$  is defined as a pair  $(n, dom)$  where  $n$  is its name, and  $dom$  is its domain, i.e. the list of possible values (also called members).*

Some dimensions can come with hierarchies. For instance, months can be merged into quarters, and semesters, years... For this purpose, the domain of a dimension can be divided into the so called *levels (of granularity)*, each level containing a subset of the domain values.

DEFINITION 2 (LEVEL). *Let  $D = (n, dom)$  be a dimension.  $L$  is said to be a level of  $D$  if  $L \subseteq dom$ .*

Levels are usually disjoint : for any level  $L$  defined on  $D$ ,  $\exists L'$  such that  $L \cap L' \neq \emptyset$ .

In this section, we present an overview of the standard models of hierarchies described in the litterature. We extensively refer to [4] and [3] for the notations and the terminology.

DEFINITION 3 (HIERARCHY). *Let  $D$  be a dimension and  $L$  be the set of levels defined on  $D$ . A hierarchy  $\mathcal{H}$  is defined as a set of binary relations between these levels. The sequence of the levels is called a (hierarchical) path and is noted  $H^0, \dots, H^k$ . Its size,  $k + 1$ , is called the length of the path. The most specific level in a path,  $H^0$ , is called the leaf and the coarsest,  $H^k$ , is called the root. We denote by  $x \in dom(H^j)$  if  $x$  is a member of  $H^j$  (i.e., if  $x$  is an instance of the level  $H^j$ ).*

We define operators in order to retrieve members at different levels of granularity.

DEFINITION 4. *The operator  $up(x)$  returns the set of the direct generalizations of  $x$ :  $up(x) = X$  where  $x \in dom(H^j)$  ( $0 \leq j \leq k - 1$ ) and  $X \in dom(H^{j+1})$ . Conversely, the operator  $down(x)$  returns the set of the direct specializations of  $x$ :  $down(x) = X$  where  $x \in dom(H^j)$  ( $0 < j \leq k$ ) and  $X \in dom(H^{j-1})$ .*

EXAMPLE 1. *For instance, if we consider the hierarchy shown in the Figure 1, we have  $up(Lyon) = \{France\}$  and  $down(USA) = \{LA, NY\}$ .*

Several kinds of hierarchies have been studied, as described below.

## 2.1 Simple hierarchies

*Simple hierarchies* are hierarchies where the hierarchy can be seen as a list of levels and members can be organized as a tree. Such hierarchies can be either categorized into symmetric or asymmetric hierarchies.

Figure 1 exhibits an example of a symmetric simple hierarchy. At the instance level, the members form a balanced tree (branches have the same length).

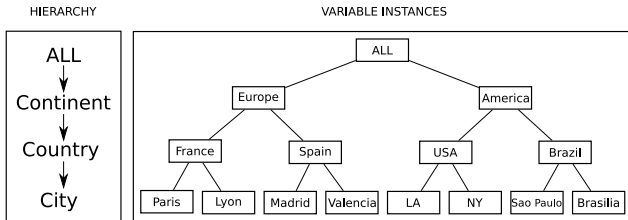


Figure 1: An example of a symmetric simple hierarchy (instance level)

Asymmetric hierarchies are hierarchies where the members from the instance level form a non-balanced tree. Figure 2 (extracted from [4]) shows an example of such an asymmetric hierarchy. We can observe that there exist agencies with no ATM (Automatic Teller Machine).

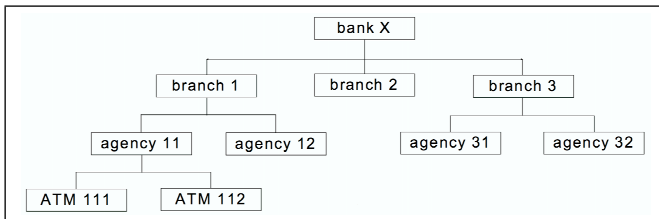


Figure 2: An example of an asymmetric simple hierarchy (instance level)

## 2.2 Multiple hierarchies

*Multiple hierarchies* represent the situation where the schema level is not composed by only one path but forms a directed graph. A typical example of multiple hierarchy is the temporal dimension's hierarchy. Figure 3 shows an example of multiple hierarchy. Indeed, the temporal dimension can be considered either along the path  $day \rightarrow week \rightarrow ALL$  or along the path  $day \rightarrow month \rightarrow year \rightarrow ALL$ .

## 2.3 Parallel hierarchies

A dimension can contain different analysis criteria. If these criteria have associated hierarchies, the hierarchy of this dimension is said to be parallel, which can be seen as a special case of multiple hierarchies. A parallel hierarchy can be composed of the different above-described hierarchies. There exist two subcategories of parallel hierarchies: dependent and independent. In a parallel independent hierarchy, the different hierarchies do not share any level. On the contrary,

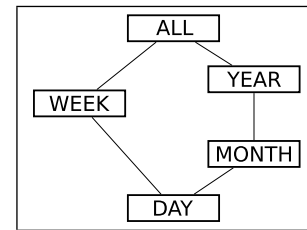


Figure 3: A temporal dimension's schema level

parallel dependent hierarchies have some hierarchies sharing some levels. Figure 4 shows an example of a parallel independent hierarchy.

## 2.4 Non-strict hierarchies

For the above-mentioned simple hierarchies, we assumed that a child member  $c$  is related to at most one parent ( $|up(c)| \leq 1$ ) and a parent member  $p$  can be related to several children ( $|down(c)| \geq 0$ ). This situation is restrictive and there exist many real-world situations where  $|up(c)| > 1$ . For instance, a product can belong to several categories. A hierarchy is called *non-strict* if there exists at the instance level at least one member where  $|up(c)| > 1$ . On the contrary, a hierarchy is called *strict* if  $|up(c)| \leq 1$  for all the members.

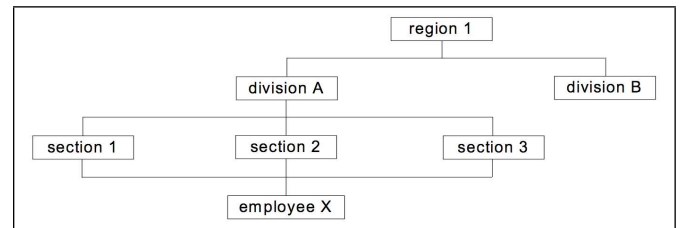


Figure 5: default

## 2.5 Fuzzy hierarchies

Fuzzy hierarchies are a special case of non strict hierarchies. They allow for the definition of granularity levels where elements can gradually belong to upper level elements. For instance, a city can be considered as belonging to some extent to the eastern part of a country *and* to another extent to the western part of the same country. Two main formal frameworks can be used for defining such fuzzy hierarchies: fuzzy graphs and fuzzy partitions, as described in [3].

## 3. A CONCEPTUAL MODEL FOR PERSONALIZED HIERARCHIES

In the previous section, we presented the principal hierarchy categories and saw that a lot of real-world applications can be handled with such categories. Nevertheless, none of the above-mentioned scenarios can be successfully modeled with these categories because none of them models the context-dependency. One may think that non-strict and fuzzy hierarchies handle such a dependency. However, let us consider the scenario 3 to convince ourselves that it is not true. Modeling the heart beat with a fuzzy hierarchy (and any kind of the above-mentioned hierarchies) would indeed lead to

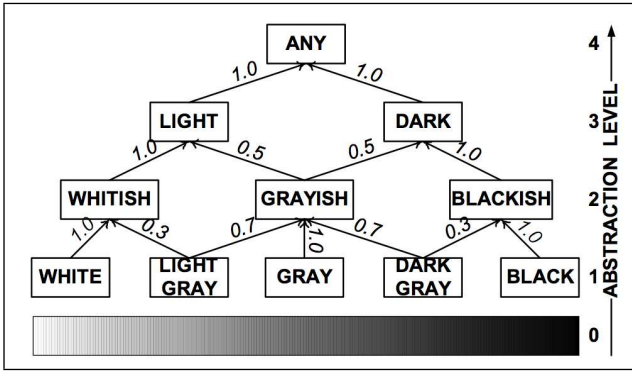


Figure 6: An illustration of a fuzzy hierarchy

consider the same membership function for all the patients. This lack of model can lead to dramatic mistakes. So, this challenge has to be met. In this section, we thus propose a new category of hierarchy which takes into account the context-dependency: *personalized hierarchies*.

Traditionally, a data cube is defined as a mapping between  $N$  dimensions of analysis  $\mathcal{D} = (\mathcal{D}_1, \dots, \mathcal{D}_N)$  and a set of measures  $\mathcal{M} = (\mathcal{M}_1, \dots, \mathcal{M}_K)$ . Thanks to hierarchies, the dimensions of analysis can usually be considered at multiple levels of granularity. On the contrary, measures are often numerical values and it is not possible to observe these values at coarser levels of granularity. As a consequence, in such traditional approaches, it is not possible to answer the query “Who got a very low arterial pressure one hour ago?” or the answer would be wrong as the current systems would consider only one way to define *low* although it depends on many parameters (e.g. age).

As previously discussed, some dimension values can impact the measure’s hierarchies (e.g. age on the normal arterial pressure). Those dimensions are referred to as the *contextual dimensions*. We denote by  $\mathcal{AC} = AC_1, \dots, AC_k$  the set of all the contextual values.

DEFINITION 5 (CONTEXTUAL DIMENSION). A dimension  $\mathcal{D}$  is said to be contextual if  $\mathcal{D}$  possesses at least one contextual value. We denote by  $\mathcal{D}_C$  the set of all the contextual dimensions.

It should be noted that if a hierarchy  $\mathcal{H}_{AC_i}$  is defined on a contextual value  $AC_i$ , it is not obvious that all the most precise values have a significant impact on the measure’s hierarchy. For instance, the precise age may not be useful for describing a normal arterial pressure. Thus, we can consider a higher level of granularity such as “baby, young, adult, old”. We denote this significant level of granularity by  $H_{AC_i}^s$ .

DEFINITION 6 (CONTEXT). Let  $c$  be a context defined on dimensions  $D_1, \dots, D_k$ .  $c$  is defined as  $c \in \text{dom}(D_1) \times \dots \times \text{dom}(D_k)$ . We call  $c$  a context and we denote the set of all the possible contexts by  $C$ .

It should be noted that all the dimensions are not always

present in a context definition. Moreover, the values taken from the domains  $\text{dom}(D_i)$  usually belong to the same level of granularity. We thus have  $c$  defined as a combination of values from  $H_{AC_i}^s$  ( $i = 1, \dots, k$ ).

EXAMPLE 2. Referring back to scenario 3, dimensions Age and Smoker are contextual and a possible context  $c$  is  $c = (\text{Adult}, \text{yes})$ .

With such new concepts, the *up* and *down* functions have to be redefined. Indeed, the traditional *up* function takes an element  $x$  and returns its father. As now the father of  $x$  depends on a context  $c$ , we consider context-aware operators *up* and *down* and we have:  $up_P(x, c)$  returns the set of the direct generalizations of  $x$  according to the context  $c$ . Conversely, the operator  $down_P(x)$  returns the set of the direct specializations of  $x$  with the associated context.

Now that we have made clear what we call a *context*, we define the concept of personalized hierarchy.

DEFINITION 7 (PERSONALIZED HIERARCHY). Let  $\mathcal{H}_i$  be the hierarchy associated to the dimension  $\mathcal{D}$ . The hierarchy  $\mathcal{H}_i$  is said to be personalized if there exist at least two contexts  $c_1$  and  $c_2$  and  $x \in \text{dom}(\mathcal{D})$  such that  $up(x, c_1) \neq up(x, c_2)$ .

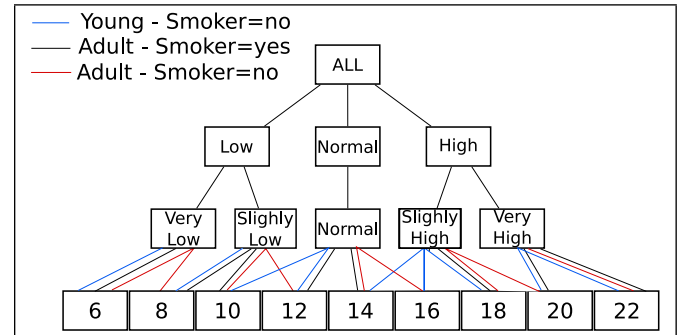


Figure 7: An example of a Personalized Hierarchy

DEFINITION 8 (PERSONALIZED MEASURE). A measure  $\mathcal{M}_i$  is said to be personalized if its associated hierarchy  $\mathcal{H}_i$  is personalized. The set of the personalized measures is denoted by  $\mathcal{M}_P$ .

EXAMPLE 3. Considering the scenario 3, dimensions Heart rate and Arterial pressure are both personalized dimensions.

DEFINITION 9 (NEUTRAL DIMENSION AND MEASURE). A dimension  $\mathcal{D}_i$  is said to be neutral if it is not contextual. The set of neutral dimensions is denoted by  $\mathcal{D}_N$ . In the same way, a measure is said to be neutral if it is not personalized. The set of neutral measures is denoted by  $\mathcal{M}_N$ .

Thus, according to the proposed model, an analysis dimension is either *neutral* or *contextual* and a measure is either *neutral* or *personalized*. With these concepts, we can define a *personalized cube*.

DEFINITION 10 (PERSONALIZED DATA CUBE). *A Personalized Data Cube is defined as a mapping such as:*

$$\text{dom}(\mathcal{D}_C) \times \text{dom}(\mathcal{D}_N) \rightarrow \text{dom}(\mathcal{M}_P) \times \text{dom}(\mathcal{M}_N)$$

We now study how to manage such personalized hierarchies, measures and cubes.

## 4. HANDLING PERSONALIZED HIERARCHIES

In the previous section, we propose a conceptual model for context-dependent hierarchies: the personalized hierarchies. Here, we focus on how to manage these hierarchies. Thus, we firstly describe the storage cost if we materialize all the hierarchies. Then, we propose a first approach for storing them efficiently.

### 4.1 Exploiting redundancy

Personalized hierarchies are powerful and model many real-world situations. Here, we consider the storage cost if all the personalized hierarchies are materialized (i.e, physically stored). Let  $M$  be a personalized hierarchy and  $\mathcal{AC} = AC_1, \dots, AC_k$  be the set of all the contextual attributes. The number of personalized hierarchies to store is  $|\mathcal{C}| = |\text{dom}(\mathcal{AC}_1^a)| \times \dots \times |\text{dom}(\mathcal{AC}_k^s)|$ . Unfortunately, this number can be huge. For instance, parameters impacting the arterial pressure are numerous. Considering the scenario 3, Figure 8 presents the relational table used if we made the choice of the full materialization. Some mechanisms have to be proposed in order to be space-saving.

ARTERIAL PRESSURE HIERARCHY
Measure (PK)
Age Category (PK)
Smoker (PK)
AP_Category
Other attributes

Figure 8: The logical representation of the  $upp$  function

Generally, only the generalizations between the leaves of the hierarchy and their father are variable. Indeed, the difficulty of personalized hierarchy is to give the appropriate semantic of numerical values. As a consequence, excluding the first generalization, the rest of the personalized hierarchies is static. As a consequence, it is useless to materialize it  $|\mathcal{C}|$  times. Thus, a naive solution would be to duplicate the leaves such as shown in Figure 9. This solution is not optimal since there exist some situations where a single item has the same generalization with two contexts  $c_1$  and  $c_2$  (i.e.,  $upp(x, c_1) = upp(x, c_2)$ ). This is particularly the case

with the terminal values. For instance, an arterial pressure equal to 6 is very low for any context. Figure 10 displays a graphical representation of common generalizations. We can observe that there exist numerous intersections. We thus take advantage of these intersections in order to reduce the storage cost.

We describe here the proposed methodology for handling the redundancy and minimize the storage cost.

1. Create the table *Hierarchy*(*IdContextGroup* (PK), *generalization*).
2. Make the intersection of the different personalized hierarchies (Figure 10).
3. Give an Id to every pair  $\langle x, C_x \rangle$  (where  $x$  is a leaf and  $C_x$  is a set of contexts so that  $\forall c_1, c_2 \in C_x$  we have  $upp(x, c_1) = upp(x, c_2)$ ). Figure 11 illustrates this step.
4. Insert the tuple  $(id, upp(x, c_1))$  in the *Hierarchy* table.
5. Add to the fact table an attribute *IdContextGroup* (FK). At each insertion in the fact table, this attribute can be easily computed thanks to the foreign keys composing the fact table.

With such a protocol, we both reduce the storage cost and allow an effective generalization.

## 5. CONCLUSION

In this paper, we address the problem of defining personalized hierarchies in multidimensional databases. In current systems, hierarchies are thus defined on dimensions and do not depend on the part of the data being considered. However, it is often necessary, in real-world applications, to consider hierarchies which depend on the values on other dimensions. For instance, the *normal* arterial pressure depends on the age of a patient. In this paper, we focus on the conceptual definitions. Further work include the detailed study of the implementation and indexing of such personalized hierarchies.

## 6. REFERENCES

- [1] J. Han, Y. Chen, G. Dong, J. Pei, B. W. Wah, J. Wang, and Y. D. Cai. Stream cube: An architecture for multi-dimensional analysis of data streams. *Distributed Parallel Databases*, 18(2), 2005.
- [2] W. H. Inmon. *Building the data warehouse*. Wiley, 2005.
- [3] A. Laurent. Querying fuzzy multidimensional databases: Unary operators and their properties. *International Journal IJUFKS*, 11:31–45, 2003.
- [4] E. Malinowski and E. Zimanyi. OLAP hierarchies: A conceptual perspective. In *Advanced Information Systems Engineering*, pages 477–491. 2004.
- [5] Y. Pitarch, A. Laurent, M. Plantevit, and P. Poncelet. Multidimensional data streams summarization using extended tilted-time windows. In *FINA*, 2009.
- [6] www.etooolsage.com. Heart Rate Calculator. [http://www.etooolsage.com/calculator/Heart\\_Rate\\_Calculator.asp?toolsort=%1500](http://www.etooolsage.com/calculator/Heart_Rate_Calculator.asp?toolsort=%1500).

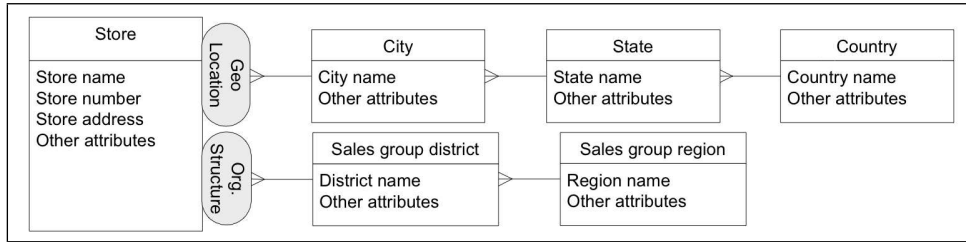


Figure 4: Parallel independent hierarchies associated to one dimension

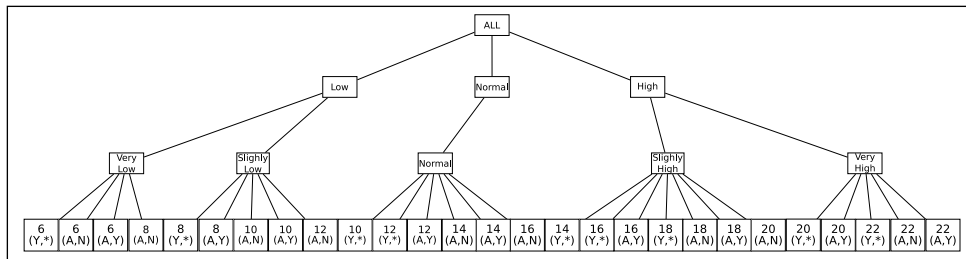


Figure 9: A naive solution for handling personalized hierarchies

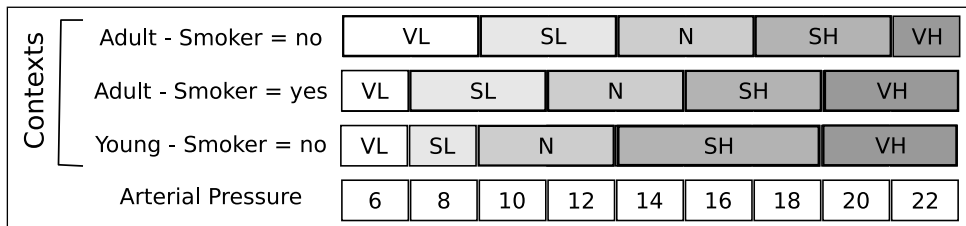


Figure 10: Graphical representation of the personal hierarchy intersection

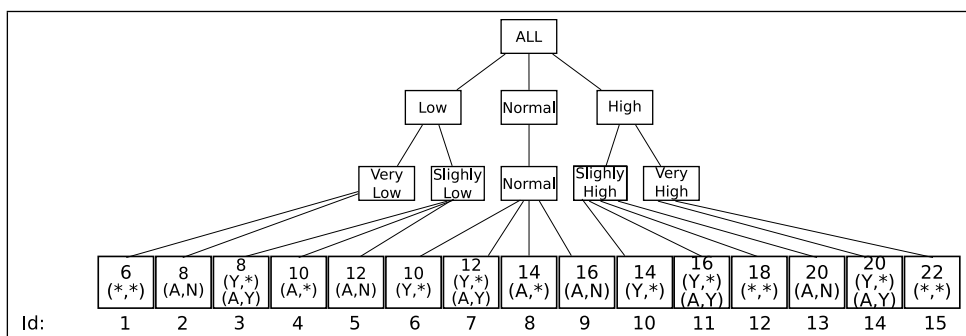


Figure 11: Proposed Methodology