Context-Aware Generalization for Cube Measures

Yoann Pitarch, Cécile Favre, Anne Laurent, Pascal Poncelet

Context-Aware Generalization for Cube Measures

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

Cécile Favre
ERIC - University Lyon 2
Lyon, France
cecile.favre@univ-lyon2.fr

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

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

ABSTRACT

Hierarchies are crucial for analysis in data warehouses. But they can hardly be defined on measure attributes. In this paper, we tackle this issue and we show that measure generalizations often depend on a context. For instance, a given blood pressure can be either low, normal or high regarding not only the collected measure but also characteristics of the patient such as the age. The contribution of this paper is threefold. (1) Thanks to an external database storing the expert knowledge, we propose an effective solution for considering these hierarchies. (2) In order to efficiently manage this knowledge, a Rich Internet Application is developed. (3) Finally, in order to provide a flexible analysis, a query rewriting module is proposed. Thus, it is possible to answer queries such as: "*Who had a low blood pressure last night?*".

Categories and Subject Descriptors

H.2.7 [**Database Management**]: Database Administration—*Data warehouse and repository*

General Terms

Design

Keywords

Data Warehouse, Modeling, Measure Generalization, Context, Expert Knowledge

1. INTRODUCTION

Observing data at different granularity levels (getting summarized or detailed data) is very helpful for the decision makers and can thus improve the decision support process. Nevertheless, in practice, most of current data warehousing models suffer from two major weaknesses in terms of their hierarchies management: (1) Few solutions allow to define a hierarchy on a measure attribute; (2) Hierarchies are supposed to be independant (orthogonal). Thus, the generalization process only depends on the specific value. Therefore, hierarchies expressing the fact that external characteristics may impact the aggregation link of a value cannot be modeled. This limitation is critical in some situations.

Let us consider a medical data warehouse recording observations of patients from an Intensive Care Unit (ICU). A hierarchy defined on the *blood pressure (BP)* measure (*e.g.,* $BP \rightarrow CatBP \rightarrow NormalityBP \rightarrow ALL_{BP}$) could be very relevant to perform tasks such as medical alarm detection (*e.g., trigger an alarm when a BP is becoming very low*), monitoring (e.g., *what is the general evolution of the patient X for the last 2 hours?*). As previously mentioned, hierarchies cannot be defined on measure attributes in the existing data warehouse models. Moreover, the categorization of a given BP depends on some physiological characteristics (*e.g.,* age of the patient, smoker or not smoker). Therefore, a given BP can be differently generalized on the $CatBP$ level depending on the analysis context. For instance, a BP of 120 mmHg (millimeters of mercury) is either normal for a smoker adult or high for a non-smoker adult. Throughout this example, we argue that expert knowledge must be taken into account to correctly generalize such measures.

More generally, in this paper, we put the emphasis on *how to generalize numerical attribute value to their symbolic generalization* (*e.g.,* how to generalize BP numerical values to the $CatBP$ level). In fact, we assume that the generalization of a symbolic level to a higher symbolic level is non-context-dependant. For instance, a *low* BP (where $low \in Dom(CatBP)$) is *abnormal* (where $abnormal \in Dom(NormalityBP)$) whatever the patient. This symbolic level to symbolic level generalization problem is not tackled here since it can be obviously figured out.

Languages such as MDX [6] allow users to define calculated elements within a query, displaying expressions computed at runtime. Most OLAP products also offer the capability to declare calculated elements in the cube schema. Thus, these calculated elements could appear sufficient to correctly perform this generalization. Indeed, these calculated elements can be seen as customized measures and can be created by combining cube data, arithmetic operators,

numbers, and functions. Thus, rules such as (**IF** $attr_1 = val_1, \ldots, attr_n = val_n$ **THEN** *generalized measure* $= k$) could be defined for materializing the expert knowledge. Nevertheless, this solution suffers from a major drawback: its inflexibility. Indeed, these rules can only be specified before the cube computation. Consequently, adding, removing or modifying one of them is very critical since it implies to recalculate the data cube. Referring back to the ICU example, one cannot predict the relevant rules to be considered since patients cannot be known in advance. Moreover, expert knowledge is evolving with respect to the advances in medicine. So, customized measures are not totally adequate to consider this evolving expert knowledge.

To the best of our knowledge, the problem of context-aware hierarchisation of measure hasn't been dealt yet. In [5], the authors propose to contextualize data warehouses with documents. However, it does not focus on the measures themselves. That constitutes a specific problem, dealing with the aggregation step, which requires an adequate representation. In [4], the authors proposed a conceptual formalization of the various types of hierarchies. This work shows a large expressive power to represent the various types of hierarchies. However, it concerns only dimension hierarchies. Moreover it cannot be adapted to our problem since it concerns only non-context-dependant hierarchies.

Since the problem is to take into account a certain context of the measure hierarchy, we were interested in proposals that introduce a flexibility during the aggregation process. In [2], the authors proposed a rule-based language to manage exceptions during the aggregation process. However, it can be summed up to a modification of the aggregation path. In our case, we need to take into account a context to define the right aggregation path (based on expert knowledge), and more precisely for measure hierarchy. In [3], a previous work proposed a rule-based model in order to allow the creation of new granularity level in dimension hierarchies, providing a solution to analyses personalisation according to users' knowledge themselves. The new levels were created over the first level of the dimension hierarchies. However, we remain in the context of classical hierarchies, where aggregation paths are predefined and are not context-dependant.

Thus, we address the problem of the context-dependency of measure generalization. Formally introduced by [7], these contextual hierarchies are not yet established in any data warehouse model. In this paper we tackle the problem of efficiently model, store, manage and use these contextual hierarchies with the following contributions:

1. The expert domain knowledge is efficiently represented thanks to an external database;

2. Since this knowledge can evolve, it is necessary to allow users to easily manage this database. So, we provide an easy-to-use Rich Internet Application (RIA) to facilitate the management of the database by the domain experts. Indeed, our interface enables insertion, update and deletion of the expert knowledge. Our solution does not lead the recalculation of the data cube;

3. Finally, in order to provide a flexible, efficient and appropriate analysis of such contextualized hierarchies, we propose a query rewriting module. Thanks to this mechanism, it is possible to correctly answer queries such as: "*Who had a low BP last night?*".

Section 2 presents a case study on medical data. Section 3 presents our contextual hierarchy formalization. In Section 4, we develop our proposal to handle the expert knowledge. The query rewriting module is presented in Section 5. Finally, we conclude in Section 6.

2. CASE STUDY

Let us consider a medical data warehouse[1] recording some observations and the dosage of each patient staying in an ICU. Such a data warehouse allows the storage of BP, heart rate and prescribed drugs for each patient at any time according to the star schema displayed on Figure 1. It should be noted that two implicit hierarchies exist in the model. The first hierarchy enables to consider the age of the patient at two levels of granularity ($Age \rightarrow CatAge \rightarrow ALL_{Age}$) and the second hierarchy enables the categorization of drugs ($IdDrug \rightarrow CatDrug \rightarrow ALL_{Drug}$).

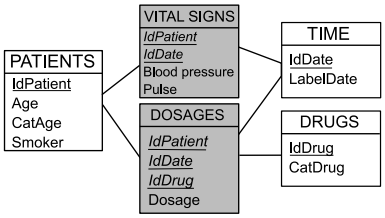

Figure 1: Schema of the data warehouse.

Typically, such medical information systems are designed to satisfy several crucial needs [8]. Among them, the medical alarm detection and the effective monitoring of patients require the correct characterization of measure's generalization. For instance, regarding the medical alarm detection scenario, it is necessary to detect when a BP is becoming abnormally high. In the same way, it would be suitable to allow doctors to formulate queries such as "Q_1 =*Who had a high BP at* t_0*?*" or "Q_2 =*Who received a high quantity of drug X this night?*".

Unfortunately, this model does not allow to formulate queries on measure's generalization due to the reasons explained in the introduction. An expert knowledge is required in order to correctly generalize a measure since the concepts of high BP or high dosage depends on some characteristics (*e.g.,* the age of the patient or the drug category). For instance, having a BP of 130 mmHg is *normal* for an adult and *high* for a baby. Table 1 presents some expert knowledge on the BP[2] categorization based on three attributes: a collected BP, an age category and a smoker attribute.

CatAge	Smoker	BP (mmHg)	CatBP
Baby	Yes or No	>110	High
Adult	Yes	>140	High
Senior citizen	Yes or No	> 140	High
Baby	Yes or No	Between 90 and 110	Normal
Adult	Yes	Between 100 and 140	Normal
...	...	...	...

Table 1: Expert knowledge sample.

[1]This work is a part of the MIDAS project founded by the french ANR agency (ANR- 07-MDCO-008).
[2]In the rest of this paper, we focus on the blood pressure.

This expert knowledge is fundamental to correctly answer to queries concerning BP generalization. For instance, let us consider the above-mentionned query Q_1. Assuming that a high BP is strictly greater than 130, this query can be expressed in SQL term as follows:

```
SELECT IdPatient FROM VITAL_SIGNS WHERE IdDate=1 AND BP>130;
```

PATIENTS			
IdPatient	Age	Smoker	CatAge
1	2	No	Baby
2	0.5	No	Baby
3	30	Yes	Adult
4	40	No	Adult
5	70	Yes	SeniorCitizen
6	80	No	SeniorCitizen
...	...	...	...

VITAL SIGNS			
IdPatient	IdDate	BP	Pulse
1	1	130	70
2	1	120	90
3	1	100	100
4	1	140	65
5	1	80	80
6	1	140	70
1	2	100	80
3	2	160	70
6	2	100	100
...	...	...	...

Figure 2: Samples of the tables.

Regarding the fact table `VITAL_SIGNS` of Figure 2, this query would return the following set of IdPatient: $\{4, 6\}$. In other words, a medical alarm detection system would have triggered an alarm for patients 4 and 6 at t_0. Let us now consider the specific knowledge presented in Table 1. Taking this knowledge into account implies to rewrite Q_1 such as:

```
SELECT IdPatient FROM VITAL_SIGNS V, PATIENTS P
WHERE  IdDate=1 AND (
 (P.IdPatient=V.IdPatient AND P.CatAge IN ('Baby')
  AND Smoker IN ('Yes','No') AND BP>110)
 OR (P.IdPatient=V.IdPatient AND P.CatAge IN ('Adult')
  AND Smoker IN ('Yes') AND BP>140)
 OR (P.IdPatient=V.IdPatient
  AND P.CatAge IN ('SeniorCitizen')
  AND Smoker IN ('Yes','No') AND BP>140));
```

Regarding the fact table of Figure 2, this query would return the following set of IdPatient: $\{1, 2\}$. In a medical alarm detection scenario, 2 alarms would have been triggered for patients whose BP was normal (*i.e.,* patients 4 and 6) whereas two patients would have suffered from non-detected high BP (*i.e.,* patients 1 and 2). This case study illustrates the need to model, store and exploit expert knowledge in order to correctly generalize context-dependant measure.

3. FORMALIZATION

Here, we propose a more flexible data warehouse model allowing to take contextualization into account.

Definition 1 (Dimensions)**.** *We note $\{D_s, 1 \leq s \leq N\}$ the set of N dimension tables, $A_s = \{a_{sg}/1 \leq s \leq N, 1 \leq g \leq h\}$ the set of the h attributes of the dimension D_s and id_{D_s} the attribute of A_s identifying the dimension D_s.*

Example 1. *In our study, $N = 3$: $D_1 \equiv PATIENTS$, $D_2 \equiv TIME$ and $D_3 \equiv DRUGS$.*
$A_1 = \{IdPatient, Age, CatAge, Smoker\}$,
$A_2 = \{IdDate, LabelDate\}$,
$A_3 = \{IdDrug, CatDrug\}$,
$id_{D_1} \equiv IdPatient$, $id_{D_2} \equiv IdDate$ and $id_{D_3} \equiv IdDrug$.

Definition 2 (Facts)**.** *A fact table is defined by a set of dimensions and measures.*

$\mathcal{F} = \{F_i, ß \geq 1\}$ is the set of fact tables of the data warehouse, some of them can share common dimensions.

A fact table F_i is defined as a pair $(\mathcal{D}_i, \mathcal{M}_i)$ where $\mathcal{D}_i = \{id_{D_s}, 1 \leq s \leq t_i\}$ are the identifiers of the t_i dimensions describing $F_ß$ (which are a subset of the N dimension tables) and $\mathcal{M} = \{M_u, 1 \leq u \leq v_i\}$ represents the set of the v_i measures of F_i.

Example 2. *For our study, we have:*
$F_1 = (\{IdPatient, IdDate\}, \{BP, Pulse\})$
and $F_2 = (\{IdPatient, IdDate, IdDrug\}, \{Dosage\})$.

As previously discussed, attributes can impact on the generalization of some measures. Now, we define the concepts of *contextualised* and *contextualising attributes* in order to introduce the concepts of *context* and *instance of context.*

Definition 3 (Contextualised and Contextualising Attributes)**.** *An attribute L is said to be* contextualising *if its value impacts on a measure generalization. This generalized attribute is thus said to be* contextualised.

Example 3. *In the BP study, $CatBP$ is a contextualised attribute since its value depends on the contextualising attributes $CatAge$, $Smoker$ and BP.*

Definition 4 (Context)**.** *A context c_i is defined by $c_i = (\{K_{\Omega_i}\}, \{L_{\Psi_i}\})$ where $\{K_{\Omega_i}\}$ is a subset of contextualising attributes and $\{L_{\Psi_i}\}$ a subset of contextualised attributes so that $\Omega_i > 1$ and $\Psi_i \geq 1$.*

Example 4. *$c_1 = (\{CatAge, Smoker, BP\}, \{CatBP\})$.*

Notice that $\{L_{\Psi_i}\}$ can be reduced to a singleton (one contextualised attribute per context). In general, $\{K_{\Omega_i}\}$ contains at least two attributes. Otherwise, we come down to the classic conception of hierarchies.

Definition 5 (Context instance)**.** *Let c_i^j be the j^{th} instance of the context c_i. A context instance corresponds to the instantiation of each attribute taking place in a context. In term of a database implementation, this instantiation should respect the SQL syntax.*

Example 5. *An instance of the context c_1 defined in the previous example could be:*
$c_1^1 = (\{=' Baby', IN('Yes',' No'), > 12\}, \{=' High'\})$.

4. HANDLING THE KNOWLEDGE

In this section, solutions for handling the expert knowledge are developed. To maintain the consistency of the model, values of the generalized measure attributes cannot be stored neither in the dimension table nor in the fact table. Moreover, since the expert knowledge takes place in the aggregation process, it cannot be taken into account during the classical loading phase but should be considered in the analysis phase. These considerations motivate the use of an external database to store the expert knowledge. Second, since this knowledge could evolve or be enriched over time by experts, it is necessary to provide them with a simple and efficient solution for the knowledge insertion, deletion or update. To this aim, a RIA was developed and is briefly exposed. Finally, the adopted representation is discussed.

4.1 Knowledge Storage

The adopted storage solution must fulfill two requirements. First, several contextual hierarchies can coexist in the same data warehouse. For instance, regarding the ICU scenario,

another possible contextual hierarchy could be defined on the *dosage* measure attribute (*e.g., Dosage* $\rightarrow$ *CatDosage* $\rightarrow$ *NormalityDosage* $\rightarrow$ ALL_D). In such a hierarchy, the set of the contextualising attributes (*i.e.,* the attributes impacting on the generalization between the *Dosage* level and the *CatDosage* level) is different (in terms of cardinality and composition) to the set of the contextualising attributes of the *BP* hierarchy. Therefore, it would be suitable to *generically* store both contexts and their associated instances. Second, since the useful expert knowledge can evolve, it is necessary to provide a *flexible* solution for storing the expert knowledge.

To fulfill these needs of genericity and flexibility, the use of an external database composed by two tables is proposed. The Knowledge Meta Table (KMT) generically stores the different contexts existing in the data warehouse (their structure). The Knowledge Table (KT) stores the instances of contexts. Notice that we consider the case of a relational implementation of the data warehouse (ROLAP).

KMT

Context	Attribute	Table	Type
BP	CatAge	PATIENTS	Context
BP	Smoker	PATIENTS	Context
BP	BP	VITAL SIGNS	Context
BP	CatBP	VITAL SIGNS	Result
Dosage	...	...	...

KT

Context	Context instance	Attribute	Values
BP	1	CatAge	IN (Baby)
BP	1	Smoker	IN (Yes,No)
BP	1	BP	>120
BP	1	CatBP	IN (High)
BP	2	CatAge	IN (Adult)
BP	2	Smoker	IN (Yes)
BP	2	BP	>140
BP	2	CatBP	IN (High)
....	...	...	..

Figure 3: Samples of the tables KMT and KT.

Definition 6 (`KMT`). *Let* `KMT` = ($Context, Attribute, Table, Type$) *be a table such that:*

- ***Context*** *points out the context identifier* (c_i)*;*
- ***Attribute*** *points out an attribute taking place in the context* c_i (i.e., $Attribute \in K_\Omega \cup L_\Psi$)*;*
- ***Table*** *points out the table where* Attribute *is stored;*
- ***Type*** *indicates if* Attribute *is either* contextualising *or* contextualised *in* c_i. Type *equals* "Context" *if* Attribute *is* contextualising *and* "Result" *otherwise.*

Note that for the contextualised attribute, the table corresponds to the one of the measure to be generalized.

Definition 7 (`KT`). *Let* `KT` = ($Context, ContextInstance, Attribute, Values$) *be a table such that:*

- ***Context*** *points out the context identifier* (c_i) *;*
- ***ContextInstance*** *points out the context instance identifier (*i.e., *represents* j *in* c_i^j*) ;*
- ***Attribute*** *points out an attribute taking place in the context* c_i (i.e., $Attribute \in K_\Omega \cup L_\Psi$)*;*
- ***Values*** *is a SQL valid expression and represents the set of values of* Attribute *in* c_i^j.

Example 6. *Let us consider the context* $BloodPressure = (\{CatAge, Smoker, BP\}, \{CatBP\})$. *This context is stored in the table KMT. The left table in Figure 3 displays the sample of the table KMT associated to this context. As illustrated on Table 1, numerous rules exist to correctly generalize a given BP. While the structure of a context is stored in the KMT, the instances of a context (i.e., the expert knowledge) is stored in the table KT. The right table of the Figure 3 displays a sample of this table. For instance, let us consider the instance "*12 is a high BP for babies.*". In the table KT, this instance is represented by the four first tuples of the table. Among these tuples, the three first ones represent the conditions to gather and the last one store the value of the generalization.*

4.2 Knowledge Update

Providing to experts an easy way to manage both context and their instances is necessary to guarantee a correct generalization since the knowledge stored in the external database can evolve. So, our storage model is relevant because the use of two predefined tables facilitates the knowledge update. Indeed, an elementary consultation of the KMT suffices to determine the existing contexts. Consequently, it is not necessary to know in advance neither the structure nor the instances to update the knowledge. Thus, a generic interface can be easily implemented to manage the expert knowledge. This justifies the development of a RIA which enables some useful features such as: (1) the context creation, update and deletion; (2) the context instance creation, update and deletion. This application has been developed under the PostgreSQL DBMS with a Web interface coded with PHP. This implementation proves the feasibility of our proposal. However, due to lack of space, we do not give more details about this RIA.

4.3 Discussion

Our approach allows to address the problem of contexts representation and storage. It thus offers a way to take into account some kind of measure hierarchy, defined by a context. This approach has several advantages.

Different contexts can be modeled in the same way. Thus, we proposed a generic approach allowing the easy addition of contexts. The contexts are stored in one table ensuring that each context is represented by various tables, eventually facilitating the process of rewriting queries.

Another advantage of this proposal is also able to group instances to define the aggregate function rather than storing the path aggregation for each instance, which is an advantage in terms of complexity (for continuous attributes more particularly). For instance, each BP value will not be an expression of its own path aggregation but a path may be defined for a set of values. For instance, the expression "> 120" (for the BP attribute) incorporates a set of BPs.

The choice of a relational representation can then exploit the power of relational querying. In such an implementation, a query rewriting mechanism can be a relevant approach. In the next section, we develop such a mechanism for exploiting such knowledge.

5. EXPLOITING THE KNOWLEDGE

As discussed in Section 1, calculated measures cannot be used to model the expert knowledge since they must be defined during the data cube conception phase. This point is very critical and leads to an important lack of flexibility. For instance, let us consider that a new patient in the ICU does not match any of the stored expert rules. As a consequence,

to correctly perform the BP generalization, some knowledge must be inserted in the external database. Specifying new rules to compute the calculated measure representing the BP category would necessitate to recompute the data cube.

We propose a query rewriting method to correctly exploit the expert knowledge, considering only conjunctives queries [1]. Further investigations need to be performed in order to consider more general queries. Nevertheless, we claim that this category of query is enough to perform monitoring or alarm detection in our study case, that fits with the need of the ICU. For instance, they allow to formulate query such that "*List all patients who had a high BP at* t_1". The rewriting process is launched when at least one condition in the *WHERE* clause considers a contextualized attribute. Now, we define *contextual queries* to formalize this idea.

Definition 8 (Contextual conjunctive query)**.** *Let* $A = \{a_1, \ldots, a_k\}$ *be a set of attributes of a fact table* F_i *(*i.e., $A \subseteq \mathcal{D}_i \cup \mathcal{M}_i$*),* $T = \{t_1, \ldots, t_l\}$ *be a set of dimension or fact tables so that* $\{F_i, KT\} \subseteq T$ *and* $Cond = (\,(a_{m_1}, op_{m_1}, val_{m_1}), \ldots, (a_{m_n}, op_{m_n}, val_{m_n})\,)$ *(where* $a_{m_j} \in A$*,* op_{m_j} *is an SQL operator and* $val_{m_j} \in Dom(a_{m_j})$ *be a conjunction of criterions so that at least one criterion concerns a contextualized attribute (called a contextual criterion). A* contextual conjunctive query Q *is a triplet* $Q = \langle A, T, Cond\rangle$ *meaning SELECT A FROM T WHERE Cond.*

Example 7. *Regarding the Definition 8,* $Q =$ "List all patients who had a high BP at t_1*" is a contextual conjunctive query and can be denoted* $Q = \langle \{IdPatient\},\ \{VITAL\_SIGNS, KT\}, (\ (IdDate, =, t_1), (CatBP, =, High)\ )\rangle$.

Notice that to lighten the definition, *contextual conjunctive query* does not allow to query attributes which are not in a fact table. Consequently, it is not possible to formulate query such as "*List the age of the patients who had a high BP at* t_1". However, we claim that our query rewriting method can be easily extended to this general case. Now, the concept of contextual conjunctive query was introduced, we develop how to exploit the contextual hierarchies.

Taking contextual hierarchies into account, contextual conjunctive queries can be performed by rewriting contextual criterion. Here, the two different steps to rewrite a contextual criterion (a, op, val) are developed. First, IdC, the concerned context must be identified. It is realized thanks to the query $Q_{context}$:

```
SELECT Context FROM KMT WHERE Type='Result' AND Attribute='a';
```

This query looks for the context in KMT where a is a result attribute. Once the context is determined, the instances of context leading the correct generalization (i.e., satisfying the contextual criterion) are determined from KT thanks to the following query $Q_{Inst.}$:

```
SELECT ContextInstance, Attribute, Values
FROM KT,KMT WHERE
KT.Context='IdC' AND KT.Context=KMT.Context
AND KT.Attribute=KMT.Attribute
AND KT.Type='Context'
ORDER BY ContextInstance;
```

Thus, each of the returned instance of context represents a conjunction of criterions to gather in order to generalize the attribute a to the value val. Consequently, the criterion (a, op, val) can be rewritten as the following disjunction of conjunction of criterions: $(\ (a_{11}, op_{11}, val_{11})\ AND\ \ldots\ AND\ (a_{1n}, op_{1n}, val_{1n})\ )\ OR \ldots OR (\ (a_{k1}, op_{k1}, val_{k1})\ AND \ldots\ AND\ (a_{kn}, op_{kn}, val_{kn}))$.

We illustrate the query rewriting mechanism thanks to the query Q:

```
SELECT IdPatient FROM VITAL_SIGNS WHERE IdDate=1 AND CatBP='High';
```

So, we focus on the contextual criterion $(CatBP, =, High)$. Thanks to $Q_{context}$ the $BloodPressure$ context is identified. Then, the instances of the $BloodPressure$ context where the value of $CatBP$ equals $High$ are identified thanks to $Q_{Inst.}$. Finally, this leads to the following rewritten query Q':

```
SELECT IdPatient FROM VITAL_SIGNS V, PATIENTS P
WHERE  idDate=1 AND (
 (P.IdPatient=V.idPatient AND P.CatAge IN ('Baby')
 AND Smoker IN ('Yes','No') AND BP>110)
OR (P.IdPatient=V.IdPatient AND P.CatAge IN('Adult')
 AND Smoker IN ('Yes') AND BP>140)
OR (P.IdPatient=V.IdPatient
 AND P.CatAge IN('SeniorCitizen')
 AND Smoker IN ('Yes','No') AND BP>140));
```

In this section, a first query rewriting approach was introduced to take contextual hierarchies into account. This first approach must be extended to consider a wider class of queries than conjunctive queries.

6. CONCLUSION AND FUTURE WORK

In this paper, we addressed the problem of context-dependant measure generalization on data warehouses, which corresponds to a need in real application, such as in medical domain. To represent the various contexts generalization of measures, an external database storing the domain knowledge is built. To manage this knowledge, we develop an easy-to-use web interface. Then, we exploit the external database and proposed a query rewriting module to correctly answer to contextual conjunctive queries in a flexible way. The feasability of our proposal has been proved thanks to an implementation of a RIA supported by PostgreSQL/PHP.

The perspectives opened by this study are numerous. We provide here two among them. First, we have to deal with the consistency of expressed knowledge and the case where various experts are not necessarily of agreement. Second, we intend to study the possibility of discovering in an automatic way contexts thanks to data mining approaches.

7. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*.

[2] M. M. Espil and A. A. Vaisman. Efficient Intensional Redefinition of Aggregation Hierarchies in Multidimensional Databases. In *DOLAP'01, Atlanta, Georgia, USA*, pages 1–8, 2001.

[3] C. Favre, F. Bentayeb, and O. Boussaid. Dimension Hierarchy Updates in Data Warehouses: a User-driven Approach. In *ICEIS'07, Funchal, Madeira, Portugal*, pages 206 – 211, 2007.

[4] E. Malinowski and E. Zimányi. OLAP Hierarchies: A Conceptual Perspective. In *CAiSE'04, Riga, Latvia*, 2004.

[5] J. Manuel Pérez-Martínez, R. Berlanga-Llavori, M. J. Aramburu-Cabo, and T. B. Pedersen. Contextualizing data warehouses with documents. *Decision Support System*, 45(1):77–94, 2008.

[6] Microsoft. MDX language reference (MDX), 2007.

[7] Y. Pitarch, A. Laurent, and P. Poncelet. A Conceptual Model for Handling Personalized Hierarchies in Multidimensional Databases. In *MEDES'09, Lyon, France*, pages 107–111, 2009.

[8] J. Van Bemmel, M. Musen, and J. Helder. *Handbook of Medical Informatics*. Bohn Stafleu Van Loghum, 1997.