Rev. Téc. Ing. Univ. Zulia. Vol. 33, Nº 3, 263 - 271, 2010

Reclassification queries in a geographical data warehouse

Francisco J. Moreno[1], Jaime Echeverri[2], Fernando Arango[1]

[1]School of Systems, University National of Colombia. Carrera 80 Nº 65-223.
[2]Universidad de Medellín. Carrera 87 Nº 30-65. Medellin, Antioquia, Colombia.
Phone: (094) 425-5376. {fjmoreno, farango}@unal.edu.co, jaecheverri@udem.edu.co

Abstract

A data warehouse is a specialized database designed to support decision-making and is usually modeled using a multidimensional view of data. A multidimensional model includes dimensions that are composed of levels. The levels of a dimension are organized in a hierarchy, *e.g.*, salespersons are grouped into stores. Throughout its lifespan a member (instance) of a level can be associated with several members of a higher level of the hierarchy, *e.g.*, the salespersons can rotate between the stores. This succession of associations enables the formulation of queries such as: "How much did a salesperson sell in his n-th season (stay) in the store X?" In this paper, we enrich this type of query, known as *season queries*, with spatial features. This enhancement enables the formulation of queries such as: "How much did a salesperson sell in his n-th season in a given geographic region?" (A spatial query window that contains a set of stores.) In order to facilitate their formulation, we propose and incorporate an operator into a multidimensional query language to demonstrate their feasibility of implementation.

Key words: Temporal data warehouses, spatial data warehouses, OLAP, members' reclassification, season queries.

Consultas de temporada espacial en un modelo multidimensional

Resumen

Una bodega de datos es una base de datos especialmente diseñada para soportar la toma de decisiones y es usualmente modelada en forma multidimensional. Un modelo multidimensional posee dimensiones las cuales se componen de niveles. Los niveles de una dimensión se organizan jerárquicamente, *e.g.*, los vendedores se agrupan en tiendas. A través de su existencia un miembro (instancia) de un nivel se puede asociar con varios miembros pertenecientes a un nivel superior en la jerarquía, *e.g.*, los vendedores pueden rotar entre las tiendas. Esta sucesión de asociaciones posibilita la formulación de consultas como: "Cuánto vendió un vendedor en su enésima temporada (estadía) en la tienda X?" En este artículo, se enriquece este tipo de consultas, conocidas como *consultas de temporadas*, con aspectos espaciales. Esta mejora posibilita la formulación de consultas como: "Cuánto vendió un vendedor en su enésima temporada (estadía) en una región geográfica" (Una región espacial que cubre un conjunto de tiendas.) Para facilitar su formulación, se propone e incorpora un operador en un lenguaje de consulta multidimensional para demostrar la viabilidad de su implementación.

Palabras clave: Bodegas de datos temporales, bodegas de datos espaciales, OLAP, reclasificación de miembros, consultas de temporadas.

1. Introduction

A data warehouse [1-2] is a specialized database designed to support decision-making and is usually modeled using a multidimensional view of data. Although there are several multidimensional models [3-11]; they share a set of key concepts such as dimension, hierarchy, level, fact, and measure, among others.

A dimension is associated with a subject of analysis, called *fact*. For example, SALESPERSON, TIME, PRODUCT, and CUSTOMER dimensions can be associated with a *sale*. A multidimensional collection of data arranged in this way is commonly referred to as a *data cube* [12] (to be referred to hereinafter simply as *cube*.)

A dimension represents a business perspective to analyze the facts and it is composed of a non-empty set of levels. For example, in Figure 1 Salesperson, Store, City, and State are levels of the SALESPERSON dimension. A level in turn has attributes [11], which provide supplementary information about the level. For example, Name and Salary are attributes of the Salesperson level. For simplicity, we do not show attributes of levels in Figure 1.

On the other hand, a fact has *measures*, *i.e.*, business metrics that analysts want to evaluate and report on, *e.g.*, number of units of a product sold and sale value, are measures of a sale.

The levels of a dimension are structured as a hierarchy according to the analysis needs [13]. The hierarchical relationship between the levels captures their *full containment* [9]. For example, in our SALESPERSON dimension, a salesperson is fully contained in a store, a store is fully contained in a city, and a city is fully contained in a state.

A member (instance) of a level can be associated with several members of a higher hierarchical level throughout its lifespan, *i.e.*, a member can be reclassified, *e.g.*, a salesperson can rotate between the stores, a product can change its category. These reclassifications originate the concept of *season* [14]. Informally, a season is a maximum interval during which a member of a level is associated with a member of a higher level. For example, suppose a salesperson Sp_1 is associated with the store St_1 from 2009-01-01 to 2009-03-15. Note that throughout his lifespan a salesperson can experience several (disjoint) seasons in the same store. Thus, the ordering of the seasons between two members originates the notion of the *n-th* season, *e.g.*, the *first* season of Sp_1 in St_1, the *second* season of Sp_1 in St_1, the *first* season of Sp_1 in St_2, and so on.

The seasons can originate queries such as: "How much did Sp_1 sell in his n-th season in St_1?" These types of queries are called *season queries* [14]. However, in [14] season queries that involve spatial features are not supported, *e.g.*, how much did Sp_1 sell in his n-th season in region R_1? (Where R_1 is a spatial query window that contains a set of stores.) In this paper, we propose an operator to support this type of query, *i.e.*, *spatial season queries*. To the best of our knowledge, there is no language or operator that allows one to formulate spatial season queries in a concise and simple way.

Although over the last years both spatial and temporal data warehouses have been an active field of research [15-16], the notion of spatial season queries is not present in any work we have found in the literature. The works closest to ours are the following. In [17] the authors focus on solving queries such as: obtain the total sales of all stores that are inside a given region (a spatial query window); however, they do not deal with members' reclassifications. Shekhar [18] proposes an operator that supports spatial aggregation in the context of a spatial multidimensional database; however, this work also does not deal with members' reclassifications. Other works [8], [19-22] deal with reclassifications but they do not consider spatial features or season queries.

This paper is organized as follows: in Section 2 we present a motivating example. In Section 3 we propose an operator to support spatial season queries and in Section 4 we give examples. Finally, in Section 5 we present conclusions and outline future work.

2. Motivating example

Consider a consortium with stores in the cities of a country. The country is divided territorially into states, which group the cities. One of the most important subjects of analysis for the

consortium are the sales of products, since from their behavior may raise strategies for production, distribution, purchasing, inventory management, marketing, among others. The products are classified into categories, *e.g.*, cosmetics, meat and dairy products, and the customers are classified by gender and age groups.

A multidimensional model for representing this scenario is shown in Figure 1. We use notations from Malinowski [15] based on the entity-relationship graphical notations. Note that, since the cardinality of every level (rectangles) participating in a fact relationship (grey diamond) is zero-to-many (crowfoot connector), such cardinalities are omitted. Note also that Store, City, and State are *spatial levels*. A spatial level is a level that the application needs to keep its spatial characteristics [15]. This is captured by its geometry represented using spatial types [23], such as Point and Region. In addition, the symbol between two spatial levels, see Figure 1, represents the topological relationship *inside* [23-24], *e.g.*, a store is inside a city and a city is inside a state.

The salespersons of the consortium tend to rotate between the stores in periods of days. The rotation is due to factors such as salesperson's experience, skills, greater number of people in certain stores at certain times, distribution and launch of products, management of replacements due to vacation, permissions, sick leaves of the salespersons. Thus, a salesperson associated with store St_1, may go on training, and later be associated with store St_2 and then return to store St_1. The temporal association between salespersons and stores is shown in Figure 1 by means of μ = day [22] (temporal unit to trace their assignments.)

The consortium is interested in analyzing how the rotation affects the performance in sales of its salespersons, e.g., analyzing the effect on the sales of a salesperson when he/she returns to the stores of a given region. For example, compare the total sales of a salesperson in his n-th season in a region regarding his previous seasons in that region. Note that factors such as knowledge acquired in previous seasons or training received before returning to a region, can influence the performance of a salesperson. The results could help identify the training that is beneficial and when it should take place, the periods for launching products, and the stays (frequency and duration) of the salespersons in the stores, all with the aim of increasing sales.

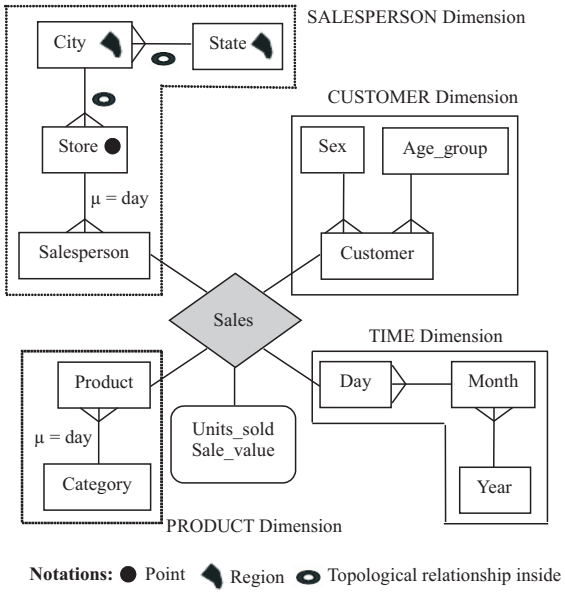

Figure 1. A multidimensional model for Sales.

Consider Figure 2 where the dashed region R_1 represents the set of stores in the western region of a country and consider the query: obtain the total sales made by salesperson Sp_1 in his first season in the stores in the western region of the country. To answer this query, according to Figure 2, the sales made by Sp_1 in his first and second seasons in St_1 and in his first season in St_2, St_3, and St_4 should be considered. Note that the sales made by Sp_1 corresponding to his *second* season in St_1 are included in the result because he *has not left* R_1. Thus, while Sp_1 rotates between the stores of R_1 without leaving this region, his sales will be part of the total requested. Eventually, when Sp_1 leaves R_1 and then returns to some store in that region, he begins his *second* season in R_1 (in Figure 2, when Sp_1 returns to St_3 from St_6.) Note that more specialized queries can be formulated, *e.g.*, obtain the total sales of cosmetics made to middle-aged women by Sp_1 in his first season in the western stores.

Similar queries to the previous ones can be applied in other fields. In the military field, where the military units perform missions at strategic sites, the following query can be formulated: In its third season when the Red Unit performed

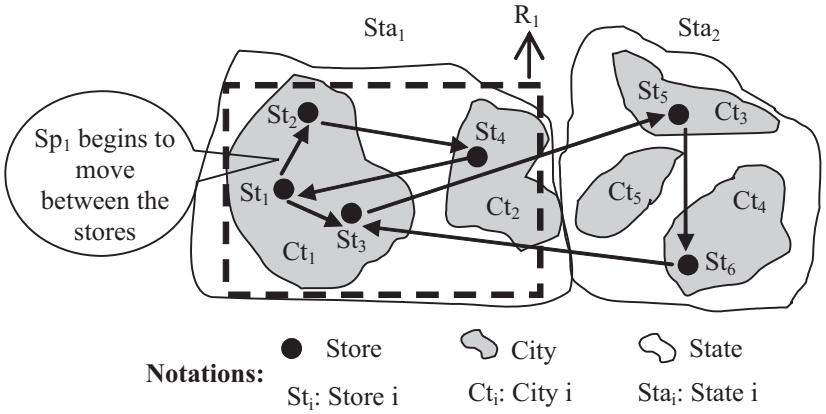

Figure 2. Rotation of salesperson Sp_1 between the stores and spatial query window R_1.

missions in sites within the southern region, did the number of casualties decrease compared to the previous two seasons? In the fishing field where the boats are regularly assigned to certain fishing spots, the following query may be formulated: What was the total salmon catch by all the boats in their first three seasons in the Polar region (where the Polar region contains a set of specific fishing spots)? In the next section we present an operator to facilitate the formulation of this type of query.

3. The Spatial_Season operator

In order to design an operator to facilitate the formulation of the queries outlined in Section 2, we identify the arguments required by such an operator. Consider Figure 3 and the query: obtain the total sales made by salesperson Sp_1 in his first season in region R_2. Assume that the SALESPERSON dimension is formed as shown in Figure 1.

Let $\mathcal{Q}$ be one of the sales made by Sp_1 in his first season in St_1. $\mathcal{Q}$ contributes to the total requested since St_1 is inside R_2. Assume now that the SALESPERSON dimension is formed as shown in Figure 4 and that the $\mathcal{Q}$ sale was made by Sp_1 when he *lived* in neighborhood N_1, see Figure 5. Thus, the $\mathcal{Q}$ sale is characterized, from the geographic point-of-view, by store St_1 that is inside R_2 (and therefore it *contributes* to the total requested) and by the neighborhood N_1 which is outside R_2 (and therefore it does not c*ontribute* to the total requested.) Therefore, the statement of the query should be clarified to avoid this ambiguity.

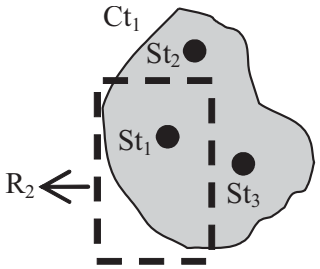

Figure 3. Spatial query window R_2.

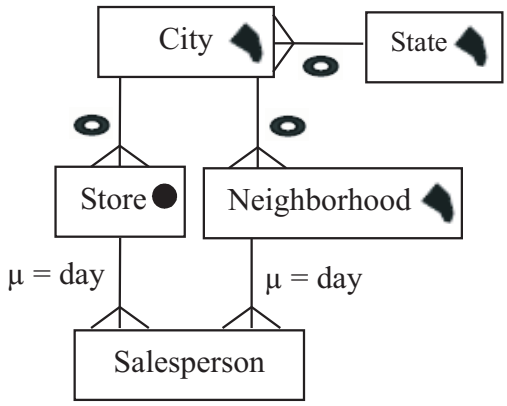

Figure 4. Alternative SALESPERSON dimension.

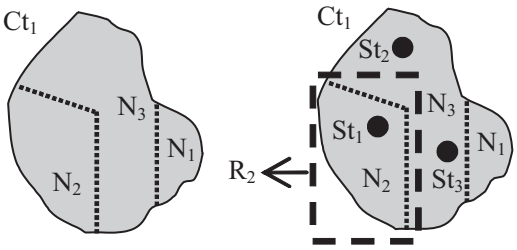

Figure 5. Neighborhoods of city Ct_1 and spatial query window R_2.

Thus, the user must specify in the statement of his query, the corresponding *geographic context*: i) to obtain the total sales made by Sp_1 in his first season in the *stores* of R_2 or ii) obtain the total sales made by Sp_1 in his first season in the *neighborhoods* of R_2. That is, the geographic context indicates the geographic elements of interest associated with the facts and that are included in a given spatial query window. Note that in the second interpretation, and according to Figure 5, the sales made by Sp_1 in St_1, St_2, and St_3, contribute to the total requested if they were made when Sp_1 lived his first season in N_2.

On the other hand, note that it is possible that in a moment in time, a salesperson lives in a neighborhood of a city different from the city of the store where he works, *i.e.*, in time *t* the city associated with a salesperson along the path that goes by the Store level might be different from the city associated with the salesperson along the path that goes by the Neighborhood level. Thereby, *Salesperson.Store.City* and *Salesperson.Neighborhood.City* represent different geographic contexts. The first refers to the city where the salesperson *works* (store) and the second refers to the city where he *lives* (neighborhood). Thus, a sale might be referred to two cities: the city of the store where the sale was made and the city where the salesperson that made the sale lives.

In order to facilitate the formulation of this type of query, we define a Spatial_Season operator. Our operator receives as arguments: i) a spatial query window, ii) a geographic context, c) a list of aggregates, where an aggregate is an aggregate function applied to a measure, and iv) a cube. Our operator returns a cube as well. For example, consider a cube corresponding to the schema of Figure 1, the region R_2 of Figure 3 and the geographic context Salesperson.Store. For each salesperson in Sales his seasons in R_2 are calculated. Each season has its start and end time (SStart and SEnd attributes) and its corresponding order number (SNumber attribute). The facts are grouped into their respective seasons along with the aggregates requested.

For example, assume that the facts (D_5, $Prod_1$, Sp_1, $Cust_1$, 10, 50$), ($D_{28}$, $Prod_1$, Sp_1, $Cust_1$, 15, 75$), ($D_{19}$, $Prod_2$, Sp_1, $Cust_1$, 8, 80$), and ($D_{35}$, $Prod_2$, Sp_1, $Cust_1$, 5, 50$) are the only ones that are part of the first season of salesperson Sp_1 in region R_2, a season that takes place between day D_1 and day D_{40}. When applying the Spatial_Season operator with the aggregate list {SUM(Sale_value)} the operator generates two facts: (S_1, $Prod_1$, Sp_1, $Cust_1$, 125$) and ($S_1$, $Prod_2$, Sp_1, $Cust_1$, 130$). This indicates that the salesperson Sp_1 sold in his first season (S_1) in region R_2 to customer $Cust_1$, 125$ of the product $Prod_1$ and 130$ of the product $Prod_2$.

We define the following syntax for the operator Spatial_Season: $Spatial\_Season_{SQW,\ GC,\ AL}(C) = C'$, where: i) SQW (Spatial Query Window): is the spatial query window, *e.g.*, region R_1 in Figure 2 and region R_2 in Figure 3, ii) GC (Geographic Context): is a path expression that indicates the geographic context. The expression is formed by the level names separated by dots, it starts with the bottom level of one dimension and ends with a spatial level of the same dimension, *e.g.*, Salesperson.Store and Salesperson.Store. City are valid geographic contexts, iii) AL (Aggregate List): is a list of elements *af(m)* where *af* is an aggregate function such as SUM, MAX, COUNT and *m* is a measure, *e.g.*, {SUM (Sale_value), MAX(Units_sold)}. Each *af(m)* generates a measure with name *afm*, *e.g.*, the previous list generates the names SUMSale_value and MAXUnits_sold, iv) C (Cube): is the cube on which the Spatial_Season operador is applied, and v) C': is the resulting cube.

Note that our operator takes a cube (C) as an argument and returns a new cube (C'), thus facilitating its integration into a multidimensional query language, and enabling the composition of queries and the integration of their results, see Section 4.

The corresponding schema for the resulting cube C' is generated as follows: i) we preserve all the dimensions of the schema of the original cube (C) except the TIME dimension, ii) the TIME dimension is replaced by a SEASON dimension with a homonymous level with attributes SStart, SEnd, and SNumber, and iii) the measures are generated from the aggregate list as explained in iii) in the previous list.

Figure 6 outlines the original and the resulting schema generated by Spatial_Season and in Table 1 we outline an algorithm to generate the

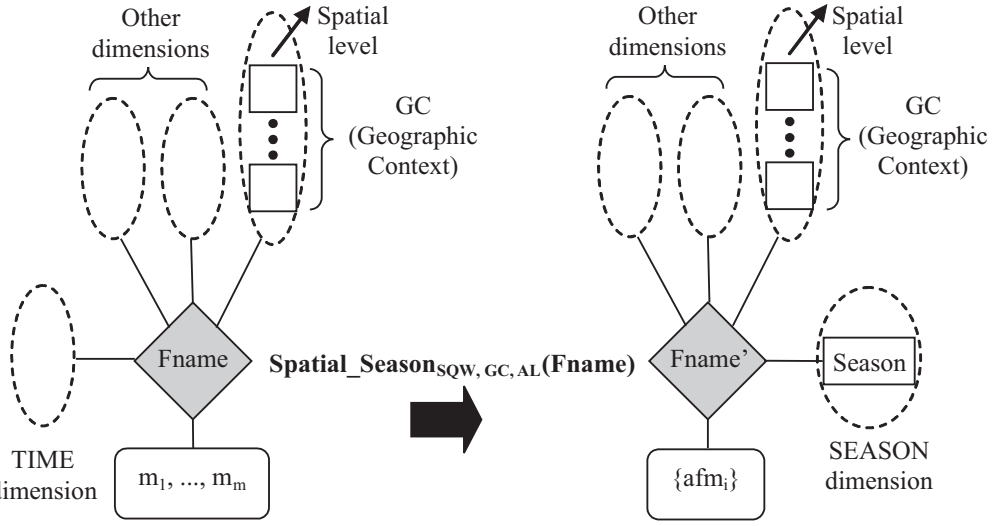

Figure 6. Spatial_Season operator: original schema and resulting schema. AL = {af(m)$_i$} where m is a measure of Fname.

Table 1
Spatial_Season operator algorithm

$Spatial\_Season_{SQW, GC, AL}(C) = C'$

Input: Cube C **Output:** Cube C'

Procedure:

Step 1. Let *last_level* be the last level of GC. Identify the members of *last_level*, which is a spatial level, that are contained in the spatial query window SQW.

Step 2. Let *first_level* be the first level of GC. For each member of *first_level* compute its seasons with regard to the region SQW and considering the members identified in the Step 1. For this purpose, consider the periods of association between the members of *first_level* and *last_level*. The results make up a SEASON dimension with a level Season and attributes SStart, SEnd, and SNumber.

Step 3. Insert the SEASON dimension, generated in Step 2, into the cube C. Each fact instance $f \in C$ is associated with a member of Season level as follows. Let SM be the set of members of Season level corresponding to the member *f.first_level*. The member $sm \in SM$ associated with the fact instance *f* is {*sm* | *sm* ∈ SM AND *sm*@SeasonStart <= *f*.time_level <= *sm*@SeasonEnd}, where *f*.time_level refers to the member of the bottom level of the TIME dimension associated with *f*. We use the symbol @ to access the attributes of a level.

Step 4. Remove the TIME dimension from C and aggregate the measures, according to the aggregate list AL, for the rest of dimensions. The output is a cube C'.

resulting cube C'. We describe how C is transformed into C'; however, in an actual implementation C should remain intact.

Consider, *e.g.*, the facts of Table 2 corresponding to the schema of Figure 1 and region R_1 of Figure 2. Suppose the first season of Sp_1 in stores that are inside R_1 is [$\mathbf{D_1}$, $\mathbf{D_{225}}$]. The results of the operation $Spatial\_Season_{R1,\ Salesperson.Store,\ \{SUM(Sale\_value)\}}(Sales)$ are shown in Table 3.

The results of Table 3 show, *e.g.*, that Sp_1 sold during his first season in R_1, that elapsed between day D_1 and day D_{225}, a total of \$ (70 + ... + 50) of $Prod_2$ to $Cust_2$.

4. Spatial season queries examples

Table 4 presents formulations to some spatial season queries using our spatial season operator. We use the multidimensional query language of Datta [25], which operates with cubes as the basic unit of input and output for all operators. This language includes typical multidimensional operators such as selection (σ) and aggregation (α). σ allows us to specify values for dimensions. Notation: $\sigma_P(Cube_1) = Cube_2$, where P is a

Table 2
Sample data of Sales fact table: Sales made by Sp_1 in his first season in stores inside R_1

Day	Salesperson	Product	Customer	Units_sold	Sale_value ($)
D_1	Sp_1	$Prod_1$	$Cust_1$	8	40
D_1	Sp_1	$Prod_2$	$Cust_2$	7	70
D_2	Sp_1	$Prod_1$	$Cust_1$	13	65
...	Sp_1	...	...	...	...
D_{225}	Sp_1	$Prod_2$	$Cust_2$	5	50

Table 3
Results of $Spatial\_Season_{R1,\ Salesperson.Store,\ \{SUM(Sale\_value)\}}(Sales) = Sales'$

Season (SStart, SEnd, SNumber)	Salesperson	Product	Customer	SUMSale_value ($)
$S_1 = (\mathbf{D_1}, \mathbf{D_{225}}, 1)$	Sp_1	$Prod_1$	$Cust_1$	40 + 65 + ...
$S_1 = (\mathbf{D_1}, \mathbf{D_{225}}, 1)$	Sp_1	$Prod_2$	$Cust_2$	70 + ... + 50

Table 4
Spatial season queries examples

User request	Query
	Let $Spatial\_Season_{R1,\ Salesperson.Store,\ \{SUM(Sale\_value)\}}(\mathbf{Sales}) = \mathbf{Cube_1}$
Obtain the total sales made by Sp_1 in his first season in the stores in the western region (R_1).	$\alpha_{[SUM(SUMSale\_value),\ \{Salesperson\}]}\sigma_{Salesperson = Sp1\ AND\ Season@SNumber\ =\ 1}(\mathbf{Cube_1})$
Obtain the total sales of cosmetics made to middle-aged women by Sp_1 in his first season in the stores in the western region (R_1).	$\alpha_{[SUM(SUMSale\_value),\ \{Salesperson\}]}\sigma_{Salesperson = Sp1\ AND\ Season@SNumber = 1\ AND\ ?Product.Category = Cosmetics\ AND\ ?Customer.Sex = Female\ AND\ Customer.Age\_group = Middle\text{-}aged}(\mathbf{Cube_1})$
Obtain the total sales made by Sp_1 in all his seasons in the stores in the western region (R_1).	$\alpha_{[SUM(SUMSale\_value),\ \{Salesperson\}]}\sigma_{Salesperson = Sp1}\ (\mathbf{Cube_1})$
Obtain the total sales made by all salespersons in their three first seasons in the stores in the western region (R_1).	$\alpha_{[SUM(SUMSale\_value),\ \{Salesperson\}]}\sigma_{Season@SNumber\ <\ 4}(\mathbf{Cube_1})$
Obtain the total sales made by Sp_1 in his second season in the neighborhoods from region R_2.	i) $Spatial\_Season_{R2,\ Salesperson.Neighborhood,\ \{SUM(Sale\_value)\}}(\mathbf{Sales}) = \mathbf{Cube_2}$ ii) $\alpha_{[SUM(SUMSale\_value),\ \{Salesperson\}]}\sigma_{Salesperson = Sp1\ AND\ Season@SNumber\ =\ 2}(\mathbf{Cube_2})$

predicate. α applies aggregate functions to measures with one or more levels of a dimension specified as grouping elements. Notation: $\alpha_{[AL, GDL]}(Cube_1) = Cube_2$. AL is a list of elements $af_i(m_i)$ where af_i is an aggregate function applied to measure m_i, and GDL is a set of grouping dimensions levels.

In order to access the attributes of levels, we propose the notation LevelName@AttributeName, *e.g.*, Season@SNumber, Salesperson@Salary; since Datta's language lacks this feature.

5. Conclusions and future work

In this paper we proposed an operator to facilitate the formulation of spatial season queries within the context of a multidimensional model, *i.e.*, queries such as: What were the total sales of cosmetics made by a salesperson to middle-aged women in his first season in the stores of a given geographic region? (A spatial query window.) This type of query can help evaluate the performance of the salespersons in the wake of their rotation between the stores. Furthermore, these queries can be useful in other domains too, where several phenomena are involved in a recurring manner in a geographic scenario, *e.g.*, analyze both the material and human losses caused by a hurricane in its n-th season in a city, state, or country. This can help not only to take preventive measures but also to evaluate their effectiveness.

As future work, we plan to incorporate our operator in a multidimensional query language such as MDX [26], a language which in recent years has become a *de facto* standard to query multidimensional data. However, in principle there are two drawbacks that ought to be considered: first MDX has no spatial features and second MDX does not support temporal relationships between levels.

On the other hand, the temporality that exists between two levels can generate a complex data type: a trajectory. For example, a salesperson rotation between the stores defines a trajectory. We believe that the management of a trajectory as a *first-class* concept in a data warehouse can similarly generate interesting queries, *e.g.*, to analyze the performance of the salespersons that have followed similar trajectories, where the notion of similarity of trajectories should be defined. For example, two salesperson trajectories could be considered similar if they have in common at least 75% of stores visited. The works [27-30] are points of departure for these issues.

Finally, we plan to experiment with real data in several domains and analyze the results in order to discover trends that may be associated with spatial seasons.

References

1. Inmon W. H.: "Building the Data Warehouse", Wiley, New York, 2005.
2. Kimball R., Ross M., Thornthwaite W., Mundy J., and Becker B.: "The Data Warehouse Lifecycle Toolkit", Wiley, New York, 2008.
3. Agrawal R., Gupta A., and Sarawagi S.: "Modeling multidimensional databases". 13th ICDE'97, Birmingham (1997) 232-243.
4. Gyssens M., and Lakshmanan L.: "A foundation for multi-dimensional databases". 23rd VLDB'97, Athens (1997) 106-115.
5. Vassiliadis P.: "Modeling multidimensional databases, cubes and cube operations". 10th SSDBM, Capri (1998) 53-62.
6. Golfarelli M., and Rizzi S.: "A methodological framework for data warehouse design". 1st DOLAP, Washington D.C. (1998) 3-9.
7. Lehner W., Albrecht J., and Wedekind H.: "Normal forms for multidimensional databases", 10th SSDBM'98, Capri (1998) 63-72.
8. Pedersen T. B., Jensen C. S., and Dyreson C. E.: "A foundation for capturing and querying complex multidimensional data". Information Systems, Vol. 26, No. 5 (2001) 383-423.
9. Jensen C. S., Kligys A., Pedersen T. B., and Timko I.: "Multidimensional data modeling for location-based services". 10th GIS 2002, McLean (2002) 55-61.
10. Timko I., Dyreson C. E., and Pedersen T. B.: "Probabilistic Data Modeling and Querying for Location-based Data Warehouses". 17th SSDBM, Santa Barbara (2005) 273-282.
11. Kumar N., Gangopadhyay A., Bapna S., Karabatis G., and Chen Z.: "Measuring interestingness of discovered skewed patterns in data cubes". Decision Support Systems, Vol. 46, No. 1 (2008), 429-439.

12. Jarke M., Lenzerini M., Vassiliou Y., and Vassiliadis P.: "Fundamentals of Data Warehouses", Springer, New York, 2003.
13. Torlone R.: Conceptual multidimensional models. In: M. Rafanelli (ed), Multidimensional Databases: Problems and Solutions. Idea Group, USA(2003), 69-90.
14. Moreno F., Arango F., and Fileto R.: "Season queries on a temporal multidimensional model". 11th IM2, Valencia (2009) to appear.
15. Malinowski E., Zimányi E.: "Advanced Data Warehouse Design: from Conventional to Spatial and Temporal Applications", Springer, New York, 2008.
16. Golfarelli M., and Rizzi S.: "A survey on temporal data warehousing". International Journal of Data Warehousing and Mining, Vol. 5, No. 1 (2009), 1-17.
17. Rao F., Zhang L., Yu X., Li Y., and Chen Y.: "Spatial hierarchy and OLAP-favored search in spatial data warehouse". 6th DOLAP, New Orleans (2003), 48-55.
18. Shekhar S., Lu C. T., Tan X., and Chawla S.: Map cube: a visualization tool for spatial data warehouses. In: H. J. Miller, J. Han (eds), Geographic Data Mining and Knowledge Discovery. Taylor and Francis, USA(2001), 73-108.
19. Chamoni P., Stock S.: "Temporal structures in data warehousing". 1st DaWaK, Florence(1999) 353-358.
20. Mendelzon A., and Vaisman A.: "Temporal queries in OLAP". 26th VLDB, Cairo (2000) 242-253.
21. Malinowski E., Zimányi E.: "A conceptual solution for representing time in data warehouse dimensions". 3rd APCCM 2006, Hobart (2006) 45-54.
22. Moreno F., Arango F., and Fileto R.: "A multigranular temporal multidimensional model". 1st miproBIS, Opatija (2009) 1-6.
23. Parent C., Spaccapietra S., and Zimányi E.: "Spatio-temporal conceptual models: data structures + space + time". 7th ACM-GIS, Kansas (1999) 26-33.
24. Schneider M.: "Computing the topological relationship of complex regions". 15th DEXA, Zaragoza (2004) 844-853.
25. Datta A., and Thomas H.: "The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses". Decision Support Systems, Vol. 27, No.3 (1999), 289-301.
26. Whitehorn M., Zare R., and Pasumansky M.: "Fast Track to MDX", Springer, New York, 2006.
27. Brakatsoulas S., Pfoser D., and Tryfona N.: "Modeling, storing, and mining moving object databases". 8th IDEAS, Coimbra (2004) 68-77.
28. Orlando S., Orsini R., Raffaeta A., and Roncato A.: "Trajectory data warehouses: design and implementation issues". Journal of Computing Science and Engineering, Vol. 1, No. 2 (2007), 211-232.
29. Marketos G., Frentzos E., Ntoutsi I, Pelekis N., Raffaetà A., and Theodoridis Y.: "Building real world trajectory warehouses". 7th MobiDE'08, Vancouver (2008) 1-8.
30. Spaccapietra S., Parent C., Damiani M. L., Fernandes de Macêdo J. A., Porto F., and Vangenot C.: "A conceptual view on trajectories". Data & Knowledge Engineering, Vol. 65, No. 1 (2008), 126-146.

Recibido el 26 de Agosto de 2009
En forma revisada el 4 de Octubre de 2010