

En español

In English

Operadores para consultas relacionadas con reclasificaciones en un modelo temporal multidimensional

Francisco Moreno¹, Fernando Arango²

RESUMEN

Usualmente las dimensiones de una bodega de datos son consideradas estáticas porque su esquema y datos tienden a no cambiar. Sin embargo, tanto el esquema como los datos de las dimensiones pueden cambiar. Este artículo se enfoca en un tipo de cambio dimensional denominado *reclasificación*, que ocurre cuando un miembro de un nivel cambia de miembro en un nivel superior de la dimensión, ejemplo, cuando un producto cambia de categoría (es reclasificado). Este tipo de cambios da lugar al concepto *período de clasificación* y a un tipo de consultas que pueden ser útiles para la toma de decisiones. Verbi-gracia, ¿cuál fue el total vendido del producto ajedrez durante su primer periodo de clasificación en la categoría juguete? Para facilitar el planteamiento de este tipo de consultas se propone un conjunto de operadores y se muestra como éstos se incorporan en SQL, un lenguaje familiar para los desarrolladores de bases de datos. También se demuestra la expresividad de los operadores propuestos, ya que la formulación de esas consultas sin usar estos operadores usualmente conduce a soluciones complejas y poco intuitivas.

Palabras claves: bodegas de datos temporales, OLAP, reclasificación, período de clasificación.

Recibido: febrero 01 de 2010

Aceptado: febrero 7 de 2011

Introducción

En los últimos años las bodegas de datos (BD) (Kumar, 2008; Golfarelli, 2009a) han demostrado su utilidad en la integración de sistemas de información y en el soporte a la toma de decisiones. Las BD se modelan usualmente de manera multidimensional. Un modelo multidimensional posee un conjunto de *dimensiones* asociadas con un fenómeno de interés para una organización denominado *hecho*. Una dimensión se compone de *niveles*, organizados jerárquicamente de acuerdo a las necesidades de la información (Torlone, 2003). Así, en una dimensión *tiempo* con niveles *día*, *mes* y *año*, los días se agrupan en meses y los meses en años; en una dimensión *producto* con niveles *producto* y *categoría*, los productos se agrupan en categorías.

Los hechos típicamente incluyen *medidas*, es decir, indicadores

Operators for reclassification queries in a temporal multidimensional model

Francisco Moreno³, Fernando Arango⁴

ABSTRACT

Data warehouse dimensions are usually considered to be static because their schema and data tend not to change; however, both dimension schema and dimension data can change. This paper focuses on a type of dimension data change called *reclassification* which occurs when a member of a certain level becomes a member of a higher level in the same dimension, e.g. when a product changes category (it is reclassified). This type of change gives rise to the notion of *classification period* and to a type of query that can be useful for decision-support. For example, What were total chess-set sales during first classification period in Toy category? A set of operators has been proposed to facilitate formulating this type of query and it is shown how to incorporate them in SQL, a familiar database developer language. Our operators' expressivity is also shown because formulating such queries without using these operators usually leads to complex and non-intuitive solutions.

Keywords: temporal data warehouse, OLAP, reclassification, classification period.

Received: February 01th 2010

Accepted: February 7th 2011

Introduction

Data warehouses (DW) (Kumar, 2008; Golfarelli, 2009a) have proved their usefulness in integrating information systems and supporting decision-making during recent years. DWs are usually modelled using a multidimensional view of data. A multidimensional model has a set of *dimensions* associated with a subject of interest for an organisation called *fact*. A dimension is composed of *levels* structured as a hierarchy according to analysis needs (Torlone, 2003). For example, in a **time** dimension with day, month, and year levels, days are grouped into months and months into years; whereas in a **product** dimension having product and category levels, products are grouped into categories.

¹ Ingeniero de sistemas. Ph.D. en Ingeniería, Universidad Nacional de Colombia, Sede Medellín. fmoreno@unal.edu.co

² Ingeniero Civil. M.Sc y Ph.D. Universidad Nacional de Colombia, Sede Medellín. farango@unal.edu.co

³ Systems Engineer. Ph.D. Engineering, Universidad Nacional de Colombia, venue Medellín. fmoreno@unal.edu.co

⁴ B.Sc. Civil Engineering. M.Sc and Ph.D., Universidad Nacional de Colombia, venue Medellín. farango@unal.edu.co

En español

que permiten evaluar actividades específicas de una organización (Malinowski, 2008). Estas medidas se pueden agregar por medio de los niveles de las dimensiones, lo que permite analizar la información desde diferentes niveles de granularidad. Por ejemplo, considérese un hecho *venta*, con una medida total vendido, asociado con las dimensiones *tiempo* y *producto*. Esta organización de la información facilita consultas tales como “obtener el total vendido de un producto, diaria, mensual o anualmente”.

Usualmente las dimensiones se consideran estáticas, mientras que a los hechos se los interpreta dinámicos, ya que nuevos hechos se insertan periódicamente en la BD. Sin embargo, el esquema y los datos de una dimensión también pueden cambiar (Golfarelli, 2009b). Este artículo se enfoca en un tipo de cambios llamado *reclasificación* y en un tipo de consultas que se originan a raíz de dichos cambios. Una reclasificación sucede cuando un miembro de un nivel cambia de miembro en un nivel superior de la dimensión, por ejemplo, cuando un producto clasificado como juguete pasa a ser clasificado como educativo.

Las reclasificaciones son frecuentes en diversos escenarios: un producto cambia de categoría, un vendedor rota por las tiendas, una tienda o un cliente cambian de estatus, un jugador cambia de equipo, un huracán se mueve de una región a otra, entre otros. En particular, en la reclasificación de productos es posible que un producto experimente varias reclasificaciones durante su existencia. Por tanto, un producto ajedrez puede ser clasificado como juguete durante un periodo (Per_1), luego ser clasificado como educativo durante otro periodo (Per_2), y posteriormente volver a ser clasificado como juguete. Esta sucesión de reclasificaciones puede originar consultas como: ¿Cuál fue el total vendido del producto ajedrez en el primer periodo cuando fue clasificado como juguete? ¿En sus tres primeros períodos cuándo fue clasificado como educativo? ¿Cuál ha sido el período de clasificación en el que el total vendido del producto ajedrez han sido las más altas y en cuál categoría estaba? ¿Mejoran las ventas de un producto en los períodos cuando se clasifica como educativo con respecto a los periodos en que se clasifica como juguete?

Las respuestas podrían ayudar a identificar los períodos apropiados para reclasificar los productos con el fin de incrementar las ventas. En otros escenarios, consultas análogas podrían servir para ajustar las políticas de rotación de personal o evaluar el desempeño de los jugadores a medida que rotan por los equipos, evaluar el impacto de fenómenos naturales cíclicos y de las políticas diseñadas para minimizar sus estragos, por ejemplo: ¿disminuyó el área de cultivos destruidos la segunda vez que el fenómeno de El Niño afectó a Perú? En este artículo se propone un conjunto de operadores para facilitar la formulación de este tipo de consultas. El resto del artículo se organiza así: en la sección 2 se presentan trabajos relacionados; en la sección 3 se ofrece el modelo formal temporal multidimensional; en la sección 4 se introduce y formaliza el concepto de período de clasificación; en la sección 5 se proponen y ejemplifican operadores para consulta de períodos de clasificación; en la sección 6 se incorporan los operadores en SQL; y finalmente, en la sección 7, se plantean las conclusiones y el trabajo futuro.

Trabajos relacionados

Aunque se han propuesto lenguajes de consulta multidimensionales (Cabibbo, 1997; Datta, 1999; Pedersen 2001; Jensen,

In English

A fact usually includes *measures*, namely, indicators allowing one to evaluate an organisation's specific activities (Malinowski, 2008). Measures can be aggregated throughout dimension levels, allowing information from different granularity levels to be analysed. For example, consider a fact *sale*, having a total sold measure associated with time and product dimensions. This way of organising data facilitates formulating queries such as, “What was the total daily, monthly or annual sale of a certain product?”

Usually, dimensions are considered static in DW while facts are considered dynamic because new facts are periodically added to DW. However, dimension schema and dimension data can also change (Golfarelli, 2009b). This paper focuses on a type of dimension data change called *reclassification* and on a type of query arising from such changes. Reclassification occurs when a member of a certain level changes its parent (a member of a higher level from the same dimension), e.g. when a product classified as a toy is reclassified as being educational.

Reclassification is common in several domains: a product changes category, a salesperson shifts position in a store, a store or a customer changes status, a player changes team, a hurricane moves from one region to another. In product reclassification, it is possible for a product to undergo several reclassifications during its lifespan. For example, a chess set can be classified as a toy during period Per_1 , then be classified as educational during period Per_2 , and again be reclassified as a toy. This succession of reclassifications may give rise to queries such as, “What was the total sale of chess sets during the first period when it was classified as a toy?” “What was the total sale of chess sets during the three first periods when it was classified as educational?” “What was the classification period when total chess sets sold has been the highest and which category were they in?” “Are sales of a product better in periods when classified as educational compared to periods when classified as a toy?”

The answers to the previous queries can help identify suitable periods for reclassifying products to increase their sales. In other domains, analogous queries can help adjust personnel shifting policies, evaluate players' performance as they shift through teams, evaluate the impact of cyclical natural phenomena as well as policies designed to minimise their damage, e.g. “Did the area of destroyed crops decrease the second time El Niño affected Peru?” A set of operators is proposed in this paper for facilitating the formulation of this type of query.

Related work

Although several multidimensional query languages have been proposed (Cabibbo, 1997; Datta, 1999; Pedersen 2001; Jensen,

2004; Whitehorn, 2005), éstos carecen de elementos que consideren el aspecto dinámico de las dimensiones. Sin embargo, existen trabajos que consideran este aspecto. Hurtado (1999) y Blaschka (1999) proponen operadores para borrar, insertar y actualizar el esquema y los datos de las dimensiones. Kaas (2004) propone operadores para cambiar el esquema de una BD, entre ellos operadores para insertar y borrar dimensiones y niveles. Otros autores (Eder, 2001; Body, 2002; Morzy, 2004; Golfarelli, 2006; Ravat, 2006; Rechy-Ramirez, 2006; Wrembel, 2007), se enfocan en el problema de versionamiento de una BD, es decir, cómo transformar o consultar datos que abarcan varias versiones de una BD originadas a partir de cambios en las dimensiones. Un estado del arte reciente sobre BD que consideran aspectos temporales, se puede ver en Golfarelli (2009b).

Unos pocos trabajos tratan específicamente con reclasificaciones. Chamoni (1999), Pedersen (2001) y Malinowski (2008) usan intervalos para registrarlas; no obstante, no plantean operadores que faciliten la formulación de consultas como las enunciadas en la sección 1. Por otro lado, en (Mendelzon, 2000) y (Vaisman, 2004) se propone un modelo multidimensional que soporta cambios en el esquema y en los datos de las dimensiones. También se propone un lenguaje de consulta denominado Tolap. Una consulta como “cuál fue el total vendido del producto ajedrez cuando ha sido clasificado como juguete” se puede formular en Tolap, pero este lenguaje no está orientado a resolver consultas como las enunciadas en la sección 1. Finalmente, en Moreno (2009a) se identifica informalmente este tipo de consultas, y en Moreno (2010) se propone un operador que puede ser considerado como una primera aproximación para resolver algunas de estas consultas. Un enfoque más completo y diferente se plantea aquí. Además, este nuevo enfoque puede ser fácilmente incorporado en SQL.

Modelo temporal multidimensional

El modelo multidimensional que se presenta a continuación se basa en el trabajo de Moreno (2009b), que a su vez se basó en el de Mendelzon (2000). \mathbb{N}_0 representa el conjunto de los números naturales que incluyen el cero y $\mathbb{N} = \mathbb{N}_0 - \{0\}$.

Dimensiones

Un *esquema de dimensión* es una 5-tupla (DNOMBRE, ND, \preceq , All, \perp), donde: i) DNOMBRE es nombre de la denominación del esquema de dimensión, ii) ND es un conjunto de niveles. Cada nivel $n \in ND$ se asocia con un conjunto de miembros (valores del nivel), es decir, un dominio denotado $dom(n)$. Un miembro puede incluir atributos (Golfarelli, 2009b), los cuales dan información adicional sobre el nivel, iii) \preceq es un orden parcial en el conjunto ND. Sean $n_1, n_2 \in ND$, $n_1 \preceq n_2$ significa que n_1 rolls up to n_2 (n_1 se agrupa en n_2). Se denota \preceq' a la reducción transitiva de \preceq , iv) All $\in ND$ es el nivel superior del orden parcial \preceq , es decir, $\forall n \in ND, n \preceq All$; $dom(All) = \{all\}$ y v) $\perp \in ND$ es el nivel inferior del orden parcial \preceq , es decir, $\forall n \in ND, \perp \preceq n$.

Ejemplo 1. Sea el esquema de dimensión (PRODUCTO, {Producto, Categoría, All}, \preceq , All, Producto), donde $\preceq' = \{(Producto, Categoría), (Categoría, All)\}$. Un miembro del nivel Producto podría incluir atributos como nombre, precio y descripción.

2004; Whitehorn, 2005), they lack elements considering the dynamic aspect of dimensions. However, some studies have considered this aspect. Hurtado (1999) and Blaschka (1999) have proposed operators for deleting, inserting and updating dimension schema and dimension data. Kaas (2004) has proposed operators for changing DW schema, including operators for inserting and deleting dimensions and levels. Other authors (Eder, 2001; Body, 2002; Morzy, 2004; Golfarelli, 2006; Ravat, 2006; Rechy-Ramirez, 2006; Wrembel, 2007) have focused on DW versioning, i.e. how to transform and/or query data covering several DW versions arising from dimension changes. A recent survey on DWs considering temporal aspects can be found in Golfarelli (2009b).

A few works have specifically dealt with reclassification. Chamoni (1999), Pedersen (2001) and Malinowski (2008) have used intervals to record them. However, they have not proposed operators facilitating the formulation of queries like those outlined in section 1. On the other hand, a multidimensional model supporting changes in dimension schema and dimension data is proposed in (Mendelzon, 2000) and (Vaisman, 2004). They have also proposed a query language called TOLAP. A query such as, “What was the total sale of chess sets when classified as a toy?” can be formulated in TOLAP; however, this language is not orientated towards solving queries such as those outlined in section 1. Moreno (2009a) informally identified this type of query and Moreno (2010) proposed an operator that can be considered a first approach to solving some of these queries. A different and more complete approach is followed here; such new approach can easily be incorporated into SQL.

Temporal multidimensional model

The multidimensional model presented here was based on work by Moreno (2009b), in turn based on work by Mendelzon (2000). \mathbb{N}_0 represented the set of natural numbers including zero and $\mathbb{N} = \mathbb{N}_0 - \{0\}$.

Dimensions

A *dimension schema* was a 5-tuple (DNAME, ND, \preceq , All, \perp) where: i) DNAME was the name of the dimension schema, ii) ND was a set of levels. Each level $n \in ND$ was associated with a set of members (level values), i.e. a domain denoted by $dom(n)$. A member could include attributes (Golfarelli, 2009b) providing supplementary information about a level, iii) \preceq was a partial order in set ND. $n_1, n_2 \in ND$, $n_1 \preceq n_2$ meant that n_1 rolled up to n_2 (n_1 was grouped into n_2). \preceq' was denoted as the transitive reduction of \preceq , iv) All $\in ND$ was the top level of partial order \preceq . $\forall n \in ND, n \preceq All$; $dom(All) = \{all\}$, and v) $\perp \in ND$ was the bottom level of partial order \preceq . $\forall n \in ND, \perp \preceq n$.

Example 1. Consider dimension schema (PRODUCT, {Product, Category, All}, \preceq , All, Product), where $\preceq' = \{(Product, Category), (Category, All)\}$. A member of level Product could include attributes such as name, price and description.

Adición de tiempo a las dimensiones

El tiempo se representa en forma discreta, o sea que un punto en la línea del tiempo corresponde a un entero positivo. Un entero positivo representa una instancia de una unidad temporal, por ejemplo, una hora, un día, un mes. Por claridad, se escribirá 'día 1' (o un valor como 'enero 1 de 2000') en vez de simplemente un 1. Además, sean $x, y, z \in \mathbb{N}$, $[x, z]$ representa un intervalo que corresponde a un conjunto de enteros positivos: $x \leq y \leq z$. Las funciones Inicio(l) y Fin(l) devuelven el primer y el último entero positivo de un intervalo l , respectivamente.

Existe una relación *ser más fino que* entre algunas unidades temporales. Por ejemplo, cada día está incluido en un mes específico, así la unidad temporal *Día* es más fina que la unidad temporal *Mes*. Por otro lado, *Semana* no es más fina que *Mes*, y *Mes* no es más fino que *Semana*. Sean μ_1 y μ_2 unidades temporales, si μ_1 es más fina que μ_2 , se escribe $\mu_1 \sim \mu_2$.

El tiempo se incorpora a las dimensiones (excepto a la dimensión TIEMPO) de dos maneras. La primera es adicionando un intervalo a cada miembro de un nivel con el fin de registrar su tiempo de existencia en la BD. La segunda es adicionando un intervalo entre dos miembros, con el propósito de registrar los períodos correspondientes a sus asociaciones (clasificaciones).

Sea un esquema de dimensión (DNOMBRE, ND , \preceq , All , \perp). Una pareja de niveles $(n_1, n_2) \in \preceq$, $n_2 \neq All$ se puede asociar con una unidad temporal μ , la cual define la granularidad temporal de reclasificación (GTR) entre n_1 y n_2 . Se dice entonces que la pareja (n_1, n_2) es *temporal*.

Ejemplo 2. Considérese el ejemplo 1. Un producto se clasifica en una categoría durante un período de días. Por lo tanto, se asocia una GTR $\mu = Día$ con la pareja (*Producto*, *Categoría*) con el objetivo de registrar las asociaciones entre productos y categorías sobre una base diaria.

Una *instancia de un esquema de dimensión* es una 2-tupla (D , FR), donde D es un esquema de dimensión y FR un conjunto de funciones *rollup*. Sea ND el conjunto de niveles de D ; $n_1, n_2 \in ND$, y \preceq el orden parcial en el conjunto ND entonces: i) para cada pareja temporal $(n_1, n_2) \in \preceq$ con GTR μ , existe una función *rollup* $RUP_{n_1, n_2}: dom(n_1) \times dom(\mu) \rightarrow dom(n_2)$ y ii) para cada pareja no-temporal $(n_1, n_2) \in \preceq$, existe una función *rollup* $RUP_{n_1, n_2}: dom(n_1) \rightarrow dom(n_2)$. Nótese que RUP_{n_1, n_2} es un metanombre, es decir, n_1 y n_2 se refieren a nombres de niveles.

Ejemplo 3. Sea una instancia del esquema de dimensión del ejemplo 2. Sean los dominios: $dom(Categoría) = \{Juguete, Educativo\}$, $dom(Producto) = \{Prod_1, Prod_2\}$, $dom(All) = \{all\}$ y $dom(Día) = \mathbb{N}$. Las funciones *rollup* entre los miembros de los niveles en el \preceq de ND se muestran en la tabla 1. Por ejemplo, $RUP_{Producto_Categoría}(Prod_1, día 1) = Juguete$.

Hechos

Un esquema de hecho es una 3-tupla (FNOMBRE, FL , M), donde: i) FNOMBRE es como se llama el esquema de hecho, ii) $FL = \{n_1, \dots, n_k\}$ es un conjunto de niveles. Cada nivel $n_i \in FL$ es el nivel inferior (\perp) en un esquema de dimensión y iii) $M = \{m_1,$

Adding time to dimensions

Time was represented as discrete: a point on a timeline corresponding to a positive integer. A positive integer represented an instance of a temporal unit, e.g. an hour, day or a month. For clarity, 'day 1' (or a value such as 'January 1 2011') has been written instead of just 1. In addition, $x, y, z \in \mathbb{N}$, $[x, z]$ represented an interval corresponding to a set of positive integers: $x \leq y \leq z$. Functions Start(l) and End(l) returned the first and the last positive integer of interval l , respectively.

There was a *finer than* relationship among temporal units. For example, a day is included in a specific month; thus temporal unit *day* is finer than temporal unit *month*. On the other hand, *week* is not finer than *month*, and *month* is not finer than *week*. μ_1 and μ_2 were temporal units, if μ_1 was finer than μ_2 , then this would be written as $\mu_1 \sim \mu_2$.

Time was incorporated into other dimensions (other than the TIME dimension) in two ways. The first was by adding an interval to each member of a level to record its timespan in the DW. The second way was by adding an interval between two members to record their corresponding association (classification) periods.

Consider a dimension schema (DNAME, ND , \preceq , All , \perp). A pair of levels $(n_1, n_2) \in \preceq$, $n_2 \neq All$, could be associated with temporal unit μ defining temporal reclassification granularity (TRG) between n_1 and n_2 . The pair (n_1, n_2) would then be said to be *temporal*.

Example 2. Consider Example 1. A product is classified in a category during a period of days. Therefore, TRG $\mu = Day$ would be associated with the pair (*Product*, *Category*) to record associations between products and categories on a daily basis.

A *dimension schema instance* was a 2-tuple (D , FR) where D was dimension schema and FR a set of rollup functions. ND was the set of levels of D ; $n_1, n_2 \in ND$, and \preceq partial order in the set ND then: i) for each temporal pair $(n_1, n_2) \in \preceq$ with TRG μ , there would be rollup function $RUP_{n_1, n_2}: dom(n_1) \times dom(\mu) \rightarrow dom(n_2)$ and ii) for each non-temporal pair $(n_1, n_2) \in \preceq$, there would be rollup function $RUP_{n_1, n_2}: dom(n_1) \rightarrow dom(n_2)$. Note that RUP_{n_1, n_2} was a *metaname*: n_1 and n_2 referred to level names.

Example 3. Consider the dimension schema given in Example 2 and domains: $dom(Category) = \{Toy, Educational\}$, $dom(Product) = \{Prod_1, Prod_2\}$, $dom(All) = \{all\}$, and $dom(Day) = \mathbb{N}$. The rollup functions between the members of levels in \preceq of ND are shown in Table 1. For example, $RUP_{Product_Category}(Prod_1, day 1) = Toy$.

Facts

A *fact schema* was a 3-tuple (FNAME, FL , M) where: i) FNAME was the name of the fact schema, ii) $FL = \{n_1, \dots, n_k\}$ was a set of levels. Each level $n_i \in FL$ was the bottom level (\perp) in a dimension schema, and iii) $M = \{m_1, \dots, m_m\}$ was a set of measures. Each measure m_i was associated with a domain $dom(m_i)$.

..., m_m } es un conjunto de medidas. Cada medida m_i se asocia con un dominio $dom(m_i)$.

Tabla 1. Ejemplos de funciones rollup para una instancia de esquema de dimensión PRODUCTO.

Pareja de niveles	Función rollup entre miembros de los niveles respectivos
(Producto, Categoría)	{((Prod ₁ , día 1), Juguete), ((Prod ₂ , día 1), Educativo), ((Prod ₁ , día 2), Juguete), ((Prod ₂ , día 2), Educativo), ..., ((Prod ₁ , día 46), Educativo), ((Prod ₂ , día 46), Educativo), ...}
(Categoría, All)	{(Juguete, all), (Educativo, all)}

Ejemplo 4. Sea el esquema de hecho (VENTAS, {Producto, Vendedor, Día}, {TotalVendido, UnidadesVendidas}), como se ilustra en la figura 1. Se usa la notación de Malinowski (2008) para representar el modelo conceptual, la cual se basa en la notación del modelo entidad-relación. Se adiciona la representación de las GRTs (Moreno, 2009b).

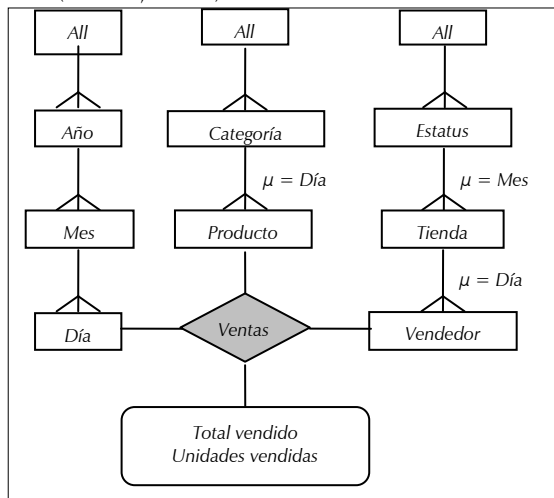


Figura 1. Un modelo temporal multidimensional de ventas.

Una instancia de hecho de un esquema de hecho (FNOMBRE, FL, M), $FL = \{n_1, \dots, n_k\}$ y $M = \{m_1, \dots, m_m\}$, es una 2-tupla (fl, m) donde $fl = \{\text{miembro}(n_1), \dots, \text{miembro}(n_k)\}$ es un conjunto de miembros y $m = \{\text{valor}(m_1), \dots, \text{valor}(m_m)\}$ es un conjunto de valores de medidas. Cada $\text{miembro}(n_i) \in dom(n_i)$ es un miembro de un nivel inferior (\perp) en una instancia de esquema de dimensión. Cada $\text{valor}(m_i) \in dom(m_i)$ es un valor para la medida m_i . Una tabla de hechos es un conjunto de instancias de hecho.

Ejemplo 5. Sean los dominios: $dom(Producto) = \{Prod_1, Prod_2\}$, $dom(Vendedor) = \{Vend_1, Vend_2\}$ y $dom(Día) = \mathbb{N}$. Una tabla de hechos Ventas correspondiente al esquema de hecho VENTAS del ejemplo 4 es $\{(\{Prod_1, Vend_1, día 1\}, \{1000, 2\}), (\{Prod_1, Vend_2, día 1\}, \{1500, 3\}), (\{Prod_2, Vend_1, día 2\}, \{2000, 4\}), (\{Prod_2, Vend_2, día 2\}, \{500, 1\})\}$.

Períodos de clasificación

Informalmente un período de clasificación (PC) es un intervalo durante el cual determinado miembro de un nivel se asocia con cierto miembro de un nivel superior, por ejemplo, un PC de un producto en una categoría, de un vendedor en una tienda, de una tienda en un estatus. A continuación se presenta una definición formal de un PC entre un miembro a de un nivel n_1 y un miembro b de un nivel superior n_k de n_1 . Sean $n_1, n_2, n_3, \dots, n_k$ niveles de un esquema de dimensión, $k > 1$, donde $n_1 \preceq n_2 \preceq$

Table 1. Examples of rollup functions for dimension schema instance - PRODUCT

Pair of levels	Rollup function between members of the respective levels
(Product, Category)	{((Prod ₁ , day 1), Toy), ((Prod ₂ , day 1), Educational), ((Prod ₁ , day 2), Toy), ((Prod ₂ , day 2), Educational), ..., ((Prod ₁ , day 46), Educational), ((Prod ₂ , day 46), Educational), ...}
(Category, All)	{(Toy, all), (Educational, all)}

Example 4. Consider the fact schema (SALES, {Product, Salesperson, Day}, {TotalSold, UnitsSold}), see Figure 1. Malinowski's notation (2008) was used for representing the conceptual model, which was based on entity-relationship model notations. TRG representation was included (Moreno, 2009b).

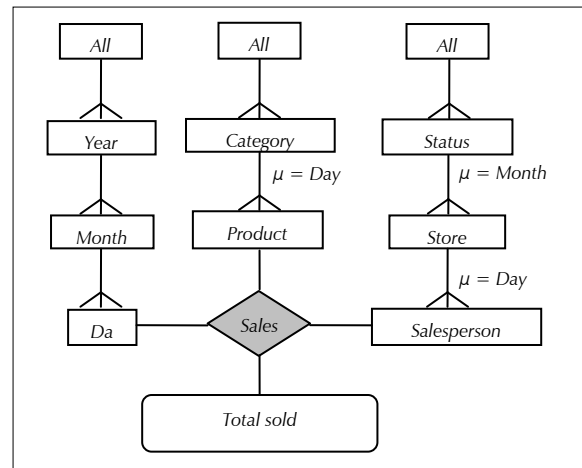


Figure 1. A temporal multidimensional sales' model

A fact instance of a fact schema (FNAME, FL, M), $FL = \{n_1, \dots, n_k\}$ and $M = \{m_1, \dots, m_m\}$, was a 2-tuple (fl, m) where $fl = \{\text{member}(n_1), \dots, \text{member}(n_k)\}$ was a set of members and $m = \{\text{value}(m_1), \dots, \text{value}(m_m)\}$ was a set of values from measures. Each $\text{member}(n_i) \in dom(n_i)$ was a member of a bottom level (\perp) in a dimension schema instance. Each $\text{value}(m_i) \in dom(m_i)$ was a value for measure m_i . A fact table was a set of fact instances.

Example 5. Consider domains: $dom(Producto) = \{Prod_1, Prod_2\}$, $dom(Salesperson) = \{Sp_1, Sp_2\}$, and $dom(Day) = \mathbb{N}$. A fact table Sales for the fact schema SALES in Example 4 was $\{(\{Prod_1, Sp_1, day 1\}, \{1000, 2\}), (\{Prod_1, Sp_2, day 1\}, \{1500, 3\}), (\{Prod_2, Sp_1, day 2\}, \{2000, 4\}), (\{Prod_2, Sp_2, day 2\}, \{500, 1\})\}$.

Classification period

Informally, a classification period (CP) is an interval during which a member of a level is associated with a member of a higher level, e.g. a CP for a product in a category, a CP for a salesperson in a store, a CP for a store regarding its status. A formal definition of a CP is presented between member a of level n_1 and member b from higher level n_k of n_1 . $n_1, n_2, n_3, \dots, n_k$ were levels of a dimension schema, $k > 1$, where $n_1 \preceq n_2 \preceq n_3 \dots \preceq n_k$. $U \neq \emptyset$ were the set of TRGs along path $n_1 \preceq n_2 \preceq n_3 \dots \preceq n_k$, $a \in dom(n_1)$ and $b \in dom(n_k)$. A CP of a in b was an interval I with

En español

In English

$n_3 \dots \leq' n_k$. Sea $U \neq \emptyset$ el conjunto de GTRs a lo largo del camino $n_1 \leq' n_2 \leq' n_3 \dots \leq' n_k$, $a \in \text{dom}(n_1)$ y $b \in \text{dom}(n_k)$. Un PC de a en b es un intervalo I con granularidad temporal μ' , donde: i) $\forall \mu \in U, \mu' \sim \mu$, ii) $\forall t \in I, \text{RUP}_{n_1, n_k}(a, t) = b$, iii) si $\text{RUP}_{n_1, n_k}(a, \text{Inicio}(I) - 1)$ está definida, entonces $\text{RUP}_{n_1, n_k}(a, \text{Inicio}(I) - 1) \neq b$, y iv) si $\text{RUP}_{n_1, n_k}(a, \text{Fin}(I) + 1)$ está definida, entonces $\text{RUP}_{n_1, n_k}(a, \text{Fin}(I) + 1) \neq b$.

Nótese que las condiciones iii) y iv) garantizan que I es un intervalo máximo durante el cual a se asocia con b . Si $U = \emptyset$ entonces durante su existencia en la BD a se asocia siempre con b ; por lo tanto, se considera a I como el único PC de a en b , donde $\text{Inicio}(I)$ y $\text{Fin}(I)$ corresponden al tiempo de existencia de a en la BD.

Ejemplo 6. Considérese el ejemplo 4. Los valores *rollup* (categorías) correspondientes a Prod_1 se muestran en la tabla 2.

Tabla 2. Valores *rollup* correspondientes (categorías) a Prod_1 .

Producto	Prod_1	Prod_1	Prod_1	Prod_1	Prod_1
Día	1 - 45	46 - 210	211 - 390	391 - 480	481 - 540
Categoría	Juguete	Educativo	Juguete	Educativo	Juguete

A partir de la tabla 2 se observa que [día 1, día 45], [día 211, día 390] y [día 481, día 540] son PC de Prod_1 en la categoría juguete. Ahora se considera el orden de los PC entre dos miembros y se define el concepto del n -ésimo PC. Sea I un PC de a en b . I es el primer PC de a en b si no existe un PC I' de a en b tal que $\text{Fin}(I') < \text{Inicio}(I)$. I es el segundo PC de a en b , si existe sólo un PC I' de a en b tal que $\text{Fin}(I') < \text{Inicio}(I)$. En general, sea $\mathbf{I} = \{I_1, I_2, \dots, I_n\}$ un conjunto de PC de a en b ; por lo tanto, $I_i \in \mathbf{I}$ es el n -ésimo PC de a en b , donde $n = |\{I_j \in \mathbf{I} \mid \text{Fin}(I_j) < \text{Inicio}(I_i)\}| + 1$. Este número se denomina número de PC. Por ejemplo, a partir del ejemplo 6 se tiene que [día 1, día 45] es el primer PC de Prod_1 en la categoría juguete, [día 211, día 390] es el segundo y [día 481, día 540] es el tercero.

Operadores de consulta para PC

Considérese el ejemplo 6. La operación *rollup* $\text{RUP}_{\text{Producto_Categoría}}(\text{Prod}_1, \text{día } 1)$ retorna la categoría del Prod_1 en el día 1 (juguete). Sin embargo, la operación no retorna ni el número ni el intervalo correspondiente al PC de Prod_1 en dicha categoría. Para obtener estos datos se propone un conjunto de operadores de consulta para PC. Los operadores deben responder a peticiones de usuario reales, como los descritos en la sección 1 y en la tabla 3.

Sean $n_1, n_2, n_3, \dots, n_k$ niveles de un esquema de dimensión, $k > 1$, donde $n_1 \leq' n_2 \leq' n_3 \dots \leq' n_k$. Sea $U \neq \emptyset$ el conjunto de GTRs a lo largo del camino $n_1 \leq' n_2 \leq' n_3 \dots \leq' n_k$; y μ' una unidad temporal arbitraria, donde $\forall \mu \in U, \mu' \sim \mu$. En la tabla 4 se presentan los operadores de consulta para PC propuestos. Si $U = \emptyset$, es decir, todas las parejas del camino $n_1 \leq' n_2 \leq' n_3 \dots \leq' n_k$ son no-temporales, entonces $\#\text{PC}_{n_1, n_k}$ y $\#\text{PCs}_{n_1, n_k}$ retornan 1, e IPC_{n_1, n_k} retorna un intervalo I con granularidad temporal μ' , donde $\text{Inicio}(I)$ y $\text{Fin}(I)$ corresponden al tiempo de existencia de a en la BD, donde a es el miembro perteneciente al dominio de n_1 .

temporal granularity μ' , where: i) $\forall \mu \in U, \mu' \sim \mu$, ii) $\forall t \in I, \text{RUP}_{n_1, n_k}(a, t) = b$, iii) if $\text{RUP}_{n_1, n_k}(a, \text{Start}(I) - 1)$ was defined then $\text{RUP}_{n_1, n_k}(a, \text{Start}(I) - 1) \neq b$ and iv) if $\text{RUP}_{n_1, n_k}(a, \text{End}(I) + 1)$ was defined then $\text{RUP}_{n_1, n_k}(a, \text{End}(I) + 1) \neq b$.

Note that conditions iii) and iv) guaranteed that I was a maximum interval during which a was associated with b . If $U = \emptyset$ then during its lifespan in the DW a was always associated with b ; therefore, I was considered as the unique CP of a in b , where $\text{Start}(I)$ and $\text{End}(I)$ corresponded to the lifespan of a in the DW.

Example 6. Consider again Example 4. The corresponding rollup values (categories) to Prod_1 are shown in Table 2.

Table 2. Correspondent rollup values (categories) for Prod_1 .

Product	Prod_1	Prod_1	Prod_1	Prod_1	Prod_1
Day	1 - 45	46 - 210	211 - 390	391 - 480	481 - 540
Category	Toy	Educational	Toy	Educational	Toy

Table 2 shows that [day 1, day 45], [day 211, day 390] and [day 481, day 540] were CPs of Prod_1 in Toy category. Now, considering the ordering of CPs between two members and defining the n^{th} CP concept. I would be a CP of a in b . I was the first CP of a in b if there were no CP I' of a in b so that $\text{End}(I') < \text{Start}(I)$. I was the second CP of a in b , if there were only CP I' of a in b so that $\text{End}(I') < \text{Start}(I)$. In general, $\mathbf{I} = \{I_1, I_2, \dots, I_n\}$ would be a set of CPs of a in b ; therefore, $I_i \in \mathbf{I}$ would be the n^{th} CP of a in b , where $n = |\{I_j \in \mathbf{I} \mid \text{End}(I_j) < \text{Start}(I_i)\}| + 1$. This number has been called CP number. For instance, in Example 6 [day 1, day 45] was the first CP of Prod_1 in Toy category, [day 211, day 390] was the second one, and [day 481, day 540] was the third one.

Query operators for CPs

Consider again Example 6. The rollup operation $\text{RUP}_{\text{Producto_Categoría}}(\text{Prod}_1, \text{day } 1)$ returned the category (Toy) of Prod_1 in day 1. However, this operation did not return the corresponding interval of Prod_1 in Toy category or the CP number. A set of query operators for CPs has thus been proposed for obtaining such data. Operators must be based on real users' requests, such as those described in Section 1 and Table 3.

$n_1, n_2, n_3, \dots, n_k$ were levels for a dimension schema, $k > 1$, where $n_1 \leq' n_2 \leq' n_3 \dots \leq' n_k$. $U \neq \emptyset$ was the set of TRGs along path $n_1 \leq' n_2 \leq' n_3 \dots \leq' n_k$; and μ' was an arbitrary temporal unit, where $\forall \mu \in U, \mu' \sim \mu$. Table 4 presents our query operators for CPs. If $U = \emptyset$, i.e. all pairs along path $n_1 \leq' n_2 \leq' n_3 \dots \leq' n_k$ were non-temporal, then $\#\text{CP}_{n_1, n_k}$ and $\#\text{CPs}_{n_1, n_k}$ returned 1, and ICP_{n_1, n_k} returned an interval I with temporal granularity μ' , where $\text{Start}(I)$ and $\text{End}(I)$ corresponded to the lifespan of a in the DW, where a was a member of the domain of n_1 .

Petición de usuario	Operadores sugeridos
¿Cuál ha sido el total vendido del producto Prod, cuando ha sido clasificado como juguete?	a) Operadores de agregación convencionales: SUM, AVG, MAX. b) <i>Rollup</i> temporal: obtener la categoría correspondiente a un producto en un tiempo específico (Mendelzon, 2000). c) Restricción y proyección de hechos con medidas agregadas.
¿Cuál fue el total vendido del producto Prod, en el primer PC cuando fue clasificado como juguete?	a) Los mismos que la petición anterior. b) Número de PC: obtener el número de PC correspondiente a un producto en una categoría en un tiempo específico.
¿Cuál fue el total vendido del producto Prod, en el último PC cuando fue clasificado como juguete?	a) Los mismos que la petición anterior. b) Número total de PC: obtener el número de PC que ha tenido un producto en una categoría específica.
¿Cuál fue el PC (fechas, número de PC y categoría) cuando el total vendido del producto Prod, ha sido el más alto?	a) Los mismos que la petición anterior. b) Intervalo del PC: obtener el intervalo correspondiente a un PC.

Tabla 3. Peticiones de usuario y operadores sugeridos para satisfacerlos.

Users' request	Proposed operators
What was the total sold of product Prod, when it was classified as a Toy?	a) Conventional aggregate functions: SUM, AVG, MAX. b) Temporal rollup: find the corresponding category associated with a product during a specific time (Mendelzon, 2000). c) Restriction and projection of facts with aggregated measures.
What was the total sold of product Prod, in the first CP when it was classified as a Toy?	a) The same as the previous request. b) CP number: find the corresponding CP number of a product in a category during a specific time.
What was the total sold of product Prod, in the last CP when it was classified as a Toy?	a) The same as the previous request. b) Total number of CPs: find the total number of CPs of a product in a specific category.
What was the CP (dates, CP number, and category) when the total sold of product Prod, has been the highest?	a) The same as the previous request. b) CP interval: find the interval of a CP.

Tabla 3. Users' requests and proposed operators

Operador	Signatura	Descripción	Ejemplo
RUP _{n₁,n_k}	$dom(n_1) \times dom(\mu') \rightarrow dom(n_k)$	<i>Rollup</i> temporal: retorna el valor <i>rollup</i> $\in dom(n_k)$ correspondiente a un valor $\in dom(n_1)$ en un tiempo específico $\in dom(\mu')$.	RUP_Producto_Categoría (Prod ₁ , day 300) = Juguete
#PC _{n₁,n_k}	$dom(n_1) \times dom(\mu') \rightarrow \mathbb{N}$	Número de PC: retorna el número de PC para un valor $\in dom(n_1)$ con respecto a su valor <i>rollup</i> $\in dom(n_k)$ en un tiempo específico $\in dom(\mu')$.	#PC_Producto_Categoría (Prod ₁ , day 300) = 2
#PCs _{n₁,n_k}	$dom(n_1) \times dom(n_k) \rightarrow \mathbb{N}$	Número de PCs: retorna el número de PCs de un valor $\in dom(n_1)$ con respecto a su valor <i>rollup</i> $\in dom(n_k)$.	#PCs_Producto_Categoría (Prod ₁ , Juguete) = 3
IPC _{n₁,n_k}	$dom(n_1) \times dom(\mu') \rightarrow CI$	Intervalo del PC: retorna el intervalo $\in CI$ (conjunto de todos los intervalos posibles) del PC para un valor $\in dom(n_1)$ con respecto a su valor <i>rollup</i> $\in dom(n_k)$ en un tiempo específico $\in dom(\mu')$.	IPC_Producto_Categoría (Prod ₁ , day 300) = [211, 390]

Tabla 4. Operadores de consulta para PC.

Operator	Signature	Description	Example
RUP _{n₁,n_k}	$dom(n_1) \times dom(\mu') \rightarrow dom(n_k)$	Temporal rollup: returns the rollup value $\in dom(n_k)$ corresponding to a value $\in dom(n_1)$ in a specific time $\in dom(\mu')$.	RUP_Product_Category (Prod ₁ , day 300) = Toy
#CP _{n₁,n_k}	$dom(n_1) \times dom(\mu') \rightarrow \mathbb{N}$	CP number: returns the CP number for a value $\in dom(n_1)$ with regard to its rollup value $\in dom(n_k)$ in a specific time $\in dom(\mu')$.	#CP_Product_Category (Prod ₁ , day 300) = 2
#CPs _{n₁,n_k}	$dom(n_1) \times dom(n_k) \rightarrow \mathbb{N}$	Number of CPs: returns the number of CPs of a value $\in dom(n_1)$ with regard to its rollup value $\in dom(n_k)$.	#CPs_Product_Category (Prod ₁ , Toy) = 3
ICP _{n₁,n_k}	$dom(n_1) \times dom(\mu') \rightarrow CI$	CP interval: returns the CP interval $\in CI$ (set of all the possible intervals) for a value $\in dom(n_1)$ with regard to its rollup value $\in dom(n_k)$ in a specific time $\in dom(\mu')$.	ICP_Product_Category (Prod ₁ , day 300) = [211, 390]

Tabla 4. Query Operators for CPs

Incorporación de los operadores de consulta para PC en SQL

En esta sección se muestra cómo se incorporan los operadores para PCs en SQL, un lenguaje familiar para los desarrolladores de bases de datos. Otra opción a considerar es MDX (Whitehorn, 2005), un lenguaje de consulta multidimensional (no temporal). En la figura 2 se muestran algunas tablas SQL del esquema de la figura 1. La tabla Prod_Cat representa la relación temporal entre producto y categoría. En la tabla 5 se registran consultas relacionadas con PC. A continuación se ofrece una solución en SQL (sin usar los operadores propuestos) para la última consulta de la tabla 5. Primero se enumeran los PC de cada producto en cada categoría:

Incorporating query operators for CPs in SQL

Here we show how to incorporate our query operators for CPs into SQL which is a familiar language for database developers. Another alternative is MDX (Whitehorn, 2005), a multidimensional query language (non-temporal). Figure 2 shows some SQL tables corresponding to the schema shown in Figure 1. Table Prod_Cat represents a temporal relationship between Product and Category. Table 5 presents some queries related to CPs. An SQL solution is now presented (without using our proposed operators) for the last query in Table 5. First, the CPs for each product in each category are enumerated:

En español

In English

```
CREATE VIEW PCs AS
SELECT IdProd, NomCat, ROW_NUMBER() OVER
(PARTITION BY IdProd, NomCat ORDER BY Dialnicio) AS NroPC, Dialnicio, DiaFin
FROM Prod_Cat;
```

Ahora se obtiene el total vendido de cada producto en cada PC:

```
SELECT IdProd, NomCat, NroPC, SUM(Total vendido) AS Total
FROM PCs, Ventas
WHERE IdProd = Product AND Dia BETWEEN Dialnicio AND DiaFin
GROUP BY IdProd, NomCat, NroPC;
```

Nótese que aunque la función analítica ROW_NUMBER facilita la formulación de esta consulta, la expresión correspondiente de la tabla 5 es más simple e intuitiva. Por otro lado, la simulación de una consulta que incluya operadores de PC entre dos niveles no adyacentes, por ejemplo, vendedor y estatus, es mucho más compleja y larga. Por límites de espacio no se presenta.

Producto		Prod_Cat			
IdProd	NomProd	IdProd	NomCat	Dialnicio	DiaFin
Prod1	Ajedrez	Prod1	Juguete	1	45
Prod2	Leche	Prod1	Educativo	46	210
Prod3	Legó	Prod1	Juguete	211	390
		Prod1	Educativo	391	480
		Prod1	Juguete	481	540
		Prod2	Alimento	1	540

Categoría		Prod_Cat			
NomCat	Importancia	IdProd	NameCat	DayStart	DayEnd
Educativo	Media	Prod1	Toy	1	45
Juguete	Baja	Prod1	Educational	46	210
Alimento	Alta	Prod1	Toy	211	390
		Prod1	Educational	391	480
		Prod1	Toy	481	540
		Prod2	Food	1	540

Ventas				
Día	Producto	Vendedor	Total vendido	Un. vendidas
1	Prod1	Vend1	10.000	1
1	Prod2	Vend2	20.000	10
2	Prod3	Vend1	8.000	1
2	Prod1	Vend2	20.000	2
2	Prod2	Vend1	12.000	6
3	Prod1	Vend1	10.000	1

Figura 2. Algunas tablas SQL del modelo temporal multidimensional de ventas.

```
CREATE VIEW CPs AS
SELECT IdProd, NameCat, ROW_NUMBER() OVER
(PARTITION BY IdProd, NameCat ORDER BY DayStart) AS NrCP, DayStart, DayEnd
FROM Prod_Cat;
```

Now we found the total sold of each product in each CP:

```
SELECT IdProd, NameCat, NrCP, SUM(TotalSold) AS Total
FROM CPs, Sales
WHERE IdProd = Product AND Day BETWEEN DayStart AND DayEnd
GROUP BY IdProd, NameCat, NrCP;
```

Note that although the analytic function ROW_NUMBER facilitated formulating this query, the corresponding expression in Table 5 is simpler and more intuitive. On the other hand, simulating a query that includes CP operators between two non-adjacent levels (e.g., salesperson and status) is much more complex and longer. This has not been shown due to space limitations.

Product		Prod_Cat			
IdProd	NameProd	IdProd	NameCat	DayStart	DayEnd
Prod1	Chess set	Prod1	Toy	1	45
Prod2	Milk	Prod1	Educational	46	210
Prod3	Legó	Prod1	Toy	211	390
		Prod1	Educational	391	480
		Prod1	Toy	481	540
		Prod2	Food	1	540

Category		Prod_Cat			
NameCat	Importance	IdProd	NameCat	DayStart	DayEnd
Educational	Medium	Prod1	Toy	1	45
Toy	Low	Prod1	Educational	46	210
Food	High	Prod1	Toy	211	390
		Prod1	Educational	391	480
		Prod1	Toy	481	540
		Prod2	Food	1	540

Sales				
Day	Product	Salesperson	TotalSold	UnitsSold
1	Prod1	Sp1	10000	1
1	Prod2	Sp2	20000	10
2	Prod3	Sp1	8000	1
2	Prod1	Sp2	20000	2
2	Prod2	Sp1	12000	6
3	Prod1	Sp1	10000	1

Figure 2. Some SQL tables of the temporal multidimensional model for sales

Tabla 5. Ejemplo de consultas relacionadas con PC en SQL.

Petición de usuario	Consulta en SQL
¿Cuál ha sido el total vendido del producto Prod, cuando ha sido clasificado como juguete?	SELECT SUM(Total vendido) AS Total FROM Ventas WHERE Product = 'Prod1' AND RUP_Producto_Categoría(Product, Día) = 'Juguete'
¿Cuál fue el total vendido del producto Prod, en el primer PC cuando fue clasificado como juguete?	SELECT SUM(Total vendido) AS Total FROM Ventas WHERE Product = 'Prod1' AND RUP_Producto_Categoría(Product, Día) = 'Juguete' AND #PC_Producto_Categoría(Product, Día) = 1
¿Cuál fue el total vendido del producto Prod, en el último PC cuando fue clasificado como juguete?	SELECT SUM(Total vendido) AS Total FROM Ventas WHERE Product = 'Prod1' AND RUP_Producto_Categoría(Product, Día) = 'Juguete' AND #PC_Producto_Categoría(Product, Día) = #PCs_Producto_Categoría(Product, Día)
¿Cuál ha sido el total vendido de cada producto en cada PC en cada categoría?	SELECT Product, RUP_Producto_Categoría(Product, Día), #PC_Producto_Categoría(Product, Día), SUM(Total vendido) AS Total FROM Ventas GROUP BY Product, RUP_Producto_Categoría(Product, Día), #PC_Producto_Categoría(Product, Día)

Table 5. Examples of queries related to CPs in SQL

User request	SQL query
What has the total sale for product Prod, been when it has been classified as a Toy?	SELECT SUM(TotalSold) AS Total FROM Sales WHERE Product = 'Prod1' AND RUP_Product_Category(Product, Day) = 'Toy'
What was the total sale for product Prod, in the first CP when it was classified as a Toy?	SELECT SUM(TotalSold) AS Total FROM Sales WHERE Product = 'Prod1' AND RUP_Product_Category(Product, Day) = 'Toy' AND #CP_Product_Category(Product, Day) = 1
What was the total sale for product Prod, in the last CP when it was classified as a Toy?	SELECT SUM(TotalSold) AS Total FROM Sales WHERE Product = 'Prod1' AND RUP_Product_Category(Product, Day) = 'Toy' AND #CP_Product_Category(Product, Day) = #CPs_Product_Category(Product, Day)
What has the total sale for each product been in each CP in each category?	SELECT Product, RUP_Product_Category(Product, Day), #CP_Product_Category(Product, Day), SUM(TotalSold) AS Total FROM Sales GROUP BY Product, RUP_Product_Category(Product, Day), #CP_Product_Category(Product, Day)

Conclusiones y trabajos futuros

Debido a las reclasificaciones que se pueden presentar entre los miembros de los niveles de una dimensión se introdujo el concepto período de clasificación. Este concepto da origen a un tipo de consultas que pueden ayudar en la toma de decisiones estratégicas en varios escenarios, tales como ventas, deportes, recursos humanos, fenómenos naturales, entre otros.

Se propuso un conjunto de operadores para facilitar la formulación de las consultas expuestas y se mostró cómo estos operadores se pueden incorporar en un lenguaje como SQL. También se expuso el esfuerzo de codificación requerido para simular los operadores propuestos en SQL, evidenciando así su expresividad.

Se plantea desarrollar los siguientes trabajos: i) extender los operadores al campo de las BD espaciales, de esta forma se podrían estructurar consultas como "obtener el total vendido por un vendedor en sus dos primeros PC cuando estuvo asociado con las tiendas delimitadas por una región determinada", ii) permitir PC superpuestos, por ejemplo, un producto podría pertenecer a varias categorías durante un mismo tiempo y iii) dividir un PC; por ejemplo, supóngase que se presenta la reelección de un gerente como presidente de un departamento. En vez de manejar un único PC para el presidente en ese departamento, se podrían definir dos PC: uno correspondiente a su primer mandato y otro a su segundo. De esta forma se podría evaluar su desempeño durante cada mandato.

Agradecimientos:

Este trabajo hace parte de los resultados del doctorado en Ingeniería - Sistemas, del primer autor.

Referencias / References

- Blaschka, M., Sapia, C., Höfling, G., On schema evolution in multidimensional databases., 1st DaWaK, Florencia, Italia, 1999, pp. 153 - 164.
- Body, M., Miquel, M., Bédard, Y., Tchounikine, A., A multidimensional and multiversion structure for OLAP applications., 5th DOLAP, McLean, USA, 2002, pp. 1 - 6.
- Cabibbo, L., Torlone, R., Querying multidimensional databases., 6th DBPL-6, Estes Park, USA, 1997, pp. 319-335.
- Golfarelli, M., Lechtenböcker, J., Rizzi, S., Vossen, G., Schema Versioning in Data Warehouses., Data and Knowledge Engineering, Vol. 59, No. 2, 2006, pp. 435-459.
- Golfarelli, M., Rizzi, S., Data Warehouse Design: Modern principles and methodologies., 1st Edn., McGraw-Hill Osborne Media, New York, 2009a.
- Golfarelli, M., Rizzi, S., A Survey on Temporal Data Warehousing., Int. Journal of Data Warehousing and Mining, Vol. 5, No. 1, 2009b, pp. 1-17.
- Hurtado, C. A., Mendelzon, A. O., Vaisman, A. A., Updating OLAP dimensions., ACM 2nd DOLAP, Kansas City, USA, 1999, pp. 60 -66.
- Jensen, C. S., Kligys, A., Pedersen, T. B., Timko, I., Multidimensional Data Modeling for Location-based Services., VLDB Journal, Vol. 13, No. 1, 2004, pp. 1-21.
- Kaas, C., Pedersen, T. B., Rasmussen, B., Schema evolution for stars and snowflakes., 6th ICEIS, Porto, Portugal, 2004, pp. 425-433.
- Kumar, N., Gangopadhyay, A., Bapna, S., Karabatis, G., Chen, Z. Measuring interestingness of discovered skewed patterns in data

Conclusions and future work

The concept of classification period was introduced due to re-classifications which may occur amongst dimension level members. This concept gives rise to a type of query which can help in decision-making in several domains such as sales, sports, human resources and natural phenomena.

A set of operators has been proposed to facilitate formulating CP queries and it has been shown how these operators can be incorporated into a language such as SQL. The encoding effort required to simulate our operators in SQL has also been shown, thus demonstrating its expressiveness.

The following work is planned. Extending our operators to the field of spatial DWs, so that queries may be formulated such as, "What was the total sold by a salesperson in his/her two first CPs when he/she was associated with the stores contained in a particular region?" Allowing overlapping CPs, e.g. a product could be associated with several categories simultaneously. It is also planned splitting a CP, e.g. suppose a manager is re-elected in a department. Instead of defining a unique CP for the manager in that department, two CPs could be defined: one corresponding to his first term and another to his/her second term. Performance in each term could thus be assessed.

cubes., Decision Support Systems, Vol. 46, No. 1, 2008, pp. 429-439.

Malinowski, E., Zimányi, E., Advanced data warehouse design: from conventional to spatial and temporal applications., Nueva York, Springer, 2008.

Mendelzon, A. O., Vaisman, A. A., Temporal queries in OLAP., 26th VLDB., El Cairo, Egipto, 2000, pp. 242-253.

Moreno, F., Arango, F., Fileto, R., Season Queries on a Temporal Multidimensional Model., 11th IM2 (presentación oral corta), Valencia, España, 2009a.

Moreno, F., Arango, F., Fileto, R., A multigranular temporal multidimensional model., 32nd Mipro, Opatija, Croacia, 2009b, pp. 206-210.

Morzy, T., Wrembel, R., On querying versions of multiversion data warehouse., ACM 7th DOLAP, Washington D.C., USA, 2004, pp. 92-101.

Pedersen, T. B., Jensen, C. S., Dyreson, C. E., A Foundation for Capturing and Querying Complex Multidimensional Data., Information Systems, Vol. 26, No. 5, 2001, pp. 383-423.

Ravat, F., Teste, O., Supporting Data Changes in Multidimensional Data Warehouses., Int. Review on Computers and Software, Vol. 1, No. 3, 2006, pp. 251-259.

Rechy-Ramirez, E., Benitez-Guerrero, E., A model and language for bitemporal schema versioning in data warehouses., 15th CIC, Ciudad de Mexico, Mexico, 2006, pp. 309-314.

Torlone, R., Conceptual Multidimensional Models., Multidimensional Databases: Problems and Solutions., M. Rafanelli (ed.), Nueva York, Idea Group, 2003, pp. 69 - 90.

Vaisman, A. A., Mendelzon, A. O., Ruaro, W., Cymerman, S. G., Supporting Dimension Updates in an OLAP Server, Information Systems, Vol. 29, No. 2, 2004, pp. 165-185.

Whitehorn, M., Zare, R., Pasumansky, M., Fast Track to MDX, Nueva York, Springer, 2005.

Wrembel, R., Bebel, B., Metadata Management in a Multiversion Data Warehouse., Journal of Data Semantics, Vol. 8, No. 1, 2007, pp. 118-157.