Calificación Difusa en Cubos OLAP

Fuzzy Qualified OLAP Cube

■ MSc. Rosseline Rodríguez
email: crodrig@ldc.usb.ve
Universidad "Simón Bolívar"

■ PhD. Leonid Tineo
email: leonid@usb.ve
Universidad Simón Bolívar, Venezuela

■ PhD. Angélica Urrutia
email: aurrutia@ucm.cl
Universidad Católica del Maule, Chile.

Fecha de Recepción: 5 de mayo de 2009
Fecha de Aceptación: 23 de septiembre de 2009

Resumen

En la actualidad, las herramientas para manejo de indicadores de gestión proveen de información cuantitativa en términos absolutos que suelen ser incomprensibles al usuario por ser lejanas a su forma de pensar, de manera que requieren un esfuerzo extra en su comprensión y uso en el proceso de toma de decisiones. Para solucionar este problema de rigidez, es conveniente dotar a tales sistemas de la habilidad de dar respuestas en términos del lenguaje natural, para lo cual la teoría de conjuntos difusos resulta adecuada. En este artículo aportamos un nuevo modelo de conjuntos difusos auto-ajustables que permite dar a los términos lingüísticos vagos una semántica acorde al contexto y la percepción del usuario, de manera que sea factible usar estos términos en la calificación del cubo OLAP. Asimismo proponemos una metodología para el desarrollo de Data Warehouse que incorpora este concepto y permite su implementación en un manejador de bases de datos tradicional. Mostramos en detalle el aporte a través de un caso de estudio real.

Palabras Clave — Consultas Difusas, Cubo Difuso, Metodología Data Warehouse, OLAP, Indicadores Gerenciales, Etiquetas Lingüísticas

TEKHNE.indb 12 1/2/10 09:11:47

Abstract

At present time, management indicators analysis tools give quantitative information in absolute numeric representation. The user might have difficulties to understand this information because it is far from human thinking. Thus extra efforts must be done in order to use it in decision making process. In order to solve this rigidity problem, such tools might be able to give answers in natural language terms. Fuzzy sets theory is an adequate for that. In this paper we contribute with a new model for auto-adjustable fuzzy sets. This model provides linguistic vague terms with a context-depend semantics expressing user perception. This semantics enables the use of these terms in OLAP cube qualification. We propose a new methodology for Data Warehouse development that incorporates this concept and allows its implementation in a traditional database management system. We show in detail how does it work throw a real life study case.

Keywords— Fuzzy Querying, Fuzzy Cube, OLAP Data Warehouse Methodology, Management Indicators, Linguistic Labels

1. INTRODUCCIÓN

Las bases de datos se han convertido en un producto estratégico de primer orden, al constituir el fundamento de los sistemas de información, y soportar la gestión y toma de decisiones gerenciales. Los grandes almacenes de datos, conocidos como Data Warehouses (DW) [6] contienen información histórica que permiten a los gerentes tomar decisiones en base al comportamiento de los datos relevantes de sus organizaciones. Este tipo de información es representada conceptualmente mediante el modelo de datos multidimensional, el cual puede ser implementado con una base de datos relacional.

Las bases de datos relacionales multidimensionales son consultadas mediante un tipo de operadores en SQL conocidas como OLAP (Online Analytical Process) [8]. Estas consultas permiten explorar los datos multidimensionales a diversos niveles de agregación, produciendo sumarios de los datos. Sin embargo estos sumarios son valores numéricos cuyo análisis requiere aún de un esfuerzo considerable por el usuario. Por ejemplo si decimos que "el volumen de ingresos de una tienda en el mes de enero para el ramo de los productos conservados fue de 3000 unidades monetarias", esta afirmación es precisa, pudo haber sido calculada mediante una consulta OLAP, pero al usuario le sería más útil si el sistema fuera capaz de decirle "el volumen de ingresos de una tienda en el mes de enero para el ramo de los productos conservados fue bajísimo". La palabra *bajísimo* es un término lingüístico vago. Este tipo de términos puede manejarse computacionalmente a través de la teoría de conjuntos difusos.

Algunas propuestas de extensión de SQL como SQLf [1] y SoftSQL [2] incorporan la teoría de conjuntos difusos para el manejo de preferencias de usuarios en consultas y otras propuestas como FSQL [4] incorporan esta teoría para el manejo de datos imperfectos. Sin embargo, estas propuestas no han considerado la posibilidad de usar términos lingüísticos en lugar de valores clásicos como resultados a las consultas OLAP.

En este artículo proponemos una metodología para Data Warehouse que permite incorporar términos lingüísticos vagos adaptables al contexto de los datos y a las preferencias del usuario para producir cubos OLAP calificados con tales términos. El modelo de conjuntos difusos que permite dar una semántica adecuada a este tipo de términos lingüísticos es un aporte novedoso del presente trabajo. Una virtud de

TEKHNE.indb 13 1/2/10 09:11:47

nuestra propuesta es que es realizable en un sistema gestor de bases de datos relacionales clásico que tenga soporte para operaciones OLAP de acuerdo con el estándar SQL.

El resto de este documento se organiza de la siguiente manera: — en la sección II se resumen los Fundamentos Teóricos de nuestra investigación, aquí se incluye el nuevo concepto de conjuntos difusos auto-ajustables concebidos en este trabajo; — la sección III está dedicada a presentar la Definición de la Metodología para el desarrollo de Data Warehouse que proponemos para incorporar términos lingüísticos vagos e implementar el cubo OLAP calificado; — en la sección IV mostramos la aplicación de nuestra propuesta a través de el Desarrollo de un Caso de Estudio real; finalmente, — la sección V sintetiza las Conclusiones del trabajo realizado.

2. Fundamentos Teóricos

A. Modelo Multidimensional

Un Data Warehouse (DW) [6] es una colección de datos, de ciertas áreas de importancia de una empresa, la cual se organiza, integra y se pone a disposición de los usuarios con el fin de facilitar la toma de decisiones gerenciales. Para diseñar un DW, en general se emplea un modelo multidimensional. Este modelo contiene tres conceptos importantes: cubo, medida y dimensión. Los cuales se ilustran en la Fig. 1.

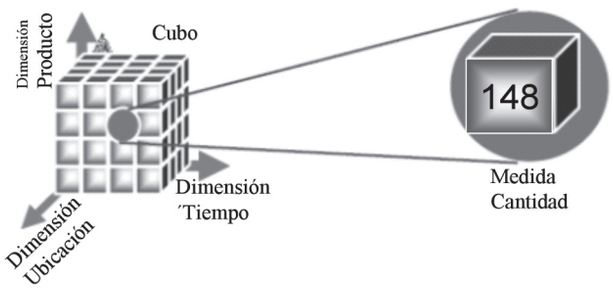

Fig. 1 Data Warehouse con tres dimensiones (Ubicación, Producto, Fecha) y una medida (Cantidad). Se muestra el valor de la medida para un registro.

Para el diseño de Data Warehouse, Carpani [3] propuso un modelo conceptual multidimensional, llamado CMDM por sus siglas en inglés (Conceptual Multi-Dimensional Model). Este modelo se compone de tres vistas: Vista de Niveles (figura 2), Vista de Dimensiones (figura 3) y Vista de Relación Dimensional (figura 4).

En la vista de Niveles se conceptualiza la definición de los niveles dimensionales. Cada nivel representa un conjunto de datos del mismo tipo similar a las clases del modelo objeto. Su representación es mediante cajas que contienen el nombre del nivel y sus atributos. La figura 2 muestra un ejemplo.

La vista de Dimensiones consiste en un diagrama de jerarquías para cada una de las dimensiones del DW. La notación gráfica de una jerarquía utiliza rectángulos para los niveles y flechas que conectan dichos niveles. Los elementos de un nivel de la jerarquía agrupan varios elementos del nivel anterior de la jerarquía. También puede decirse que cada elemento de un nivel tiene un representante en el nivel superior de la jerarquía. Una Vista de Dimensiones compuesta de sólo una dimensión, llamada *Tiempo*, se muestra en la figura 3. Por ejemplo, el dato <*24/12/2007*> sería un elemento del nivel *fecha*, este dato tiene como representante en el nivel *mes* el <*Diciembre, 2007*> el cual a la vez es representado en el nivel *año* por el *2007*. Asimismo, <*24/12/2007*> del nivel "*fecha*", tiene como representante en el nivel *temporada* el <*Invierno, 2007*> el cual a la vez es representado en el nivel *estación* por el <*Invierno*>.

Fig. 2 CMMD: Vista de Niveles de un Data Warehouse.

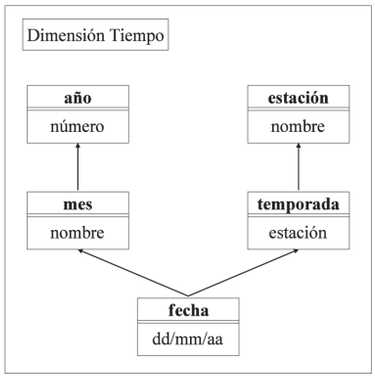

Fig. 3 CMMD: Vista de Dimensiones de un Data Warehouse.

TEKHNE.indb 14 1/2/10 09:11:47

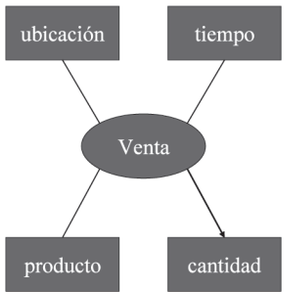

Fig. 4 CMMD: Vista de Relación Dimensional de un DW.

La vista de Relación Dimensional es un diagrama en forma de estrella compuesto por un óvulo central que identifica al cubo al cual se unen cajas periféricas que identifican las dimensiones y medidas, éstas últimas señaladas con una flecha (figura 4). Una relación dimensional representa todos los cubos que se pueden construir a partir de los niveles de un conjunto dado de dimensiones. En cada uno de los cubos que pertenecen a la relación dimensional, cada dimensión debe estar en un nivel determinado.

B. Diseño lógico del DataWarehouse

Usualmente un Data Warehouse se implementa a partir de datos existentes en una o varias bases de datos fuentes provenientes de sistemas transaccionales. En el caso de ser varias bases de datos, éstas son primero integradas para hacer luego el proceso de construcción del DW. A partir de un diseño conceptual del DW en CMDM y una base de datos relacional fuente es posible generar un modelo lógico del DW aplicando un proceso propuesto por Peralta [9]. Este proceso se ilustra en la figura 5. Primero se especifica una correspondencia entre el CMDM y el modelo relacional de la base de datos fuente, la cual consiste en indicar para cada elemento del modelo, dónde se encuentra representado en la base de datos.

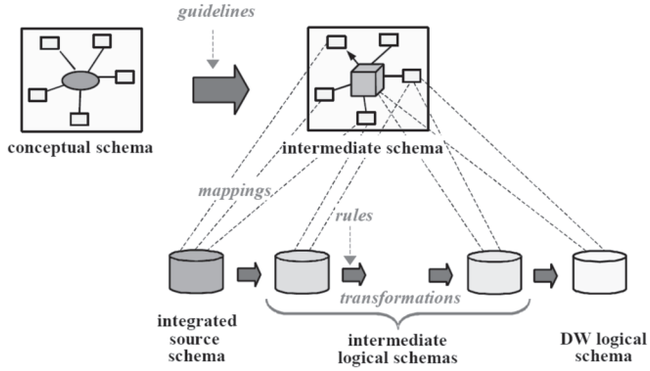

Fig. 5 Proceso de diseño lógico de un DW (Fuente Peralta 2001).

Todos los atributos requeridos se integran en un esquema intermedio. Luego se realiza una serie de transformaciones sobre el esquema integrado para producir el esquema lógico final del DW. Entre estas transformaciones se encuentran: la extracción e integración de atributos de la base de datos fuente del sistema y de otras fuentes externas; el filtrado de los datos para eliminar información innecesaria; la modificación de formatos y valores; la estandarización de las tablas de códigos de los sistemas operacionales y simplificación esquemas de codificación; y la agregación de nuevos campos, a través de cálculos, consolidaciones y derivaciones de datos.

La correspondencia o mapeo (del inglés *mapping*) entre el modelo multidimensional y el diagrama de la base de datos fuente puede hacerse mediante un paralelo de los diagramas en el cuales con flechas continuas se indica cada elemento del modelo multidimensional donde se encuentra en la base de datos fuente. En el caso que el valor para el dato del modelo multidimenional se consiga mediante operaciones sobre el dato de la base de datos, se coloca una línea segmentada para indicar que es calculado y se especifica aparte la forma del cálculo. Algunas medidas podrían ser calculadas directamente a partir de otras en el mismo modelo multidimensional, en este caso se coloca la fórmula en el diagrama de correspondencia usando para ello una plantilla en forma de pentágono. Para ver un ejemplo de esto, refiérase al apartado E de la sección IV (más adelante) donde se muestra el Desarrollo de un Caso de Estudio, allí se muestra y se explica este tipo de diagrama de correspondencia.

El modelo lógico resultante de este proceso está compuesto por tablas de hechos y tablas de dimensiones. Los hechos son los indicadores de gestión. La forma general del modelo sería como la mostrada en la figura 6. Se distinguen dos tipos de diseños lógicos: — Esquema Estrella, en éste cada tabla de dimensión estará relacionada con la tabla de hechos. Las tablas de dimensiones estarán enlazadas a la tabla de hechos mediante una clave foránea. La clave primaria en la tabla de hechos se compone de una relación de las diferentes claves primarias de las tablas de dimensiones. El esquema en estrella consiste de una tabla central de hechos y varias tablas de dimensión no normalizadas. — Esquema Copo de Nieve, en éste las tablas de dimensiones son normalizadas, obteniendo una jerarquía de datos. Con esto se simplifican las operaciones de selección de datos para lograr una

TEKHNE.indb 15 1/2/10 09:11:48

representación de la información sin redundancia. Este esquema representa mejor la semántica de las dimensiones del ambiente de los negocios, ya que tiene un acceso mas directo a los datos, lo cual se traduce en una eficiente recuperación de la información contenida en las tablas. La versión normalizada del esquema en estrella es el esquema copo de nieve.

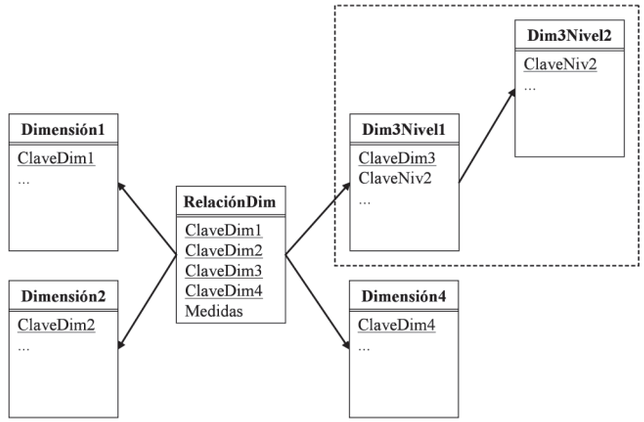

Fig. 6 Diagrama relacional de un Esquema Estrella. La esquina superior derecha muestra en un cómo sería un Esquema Copo de Nieve.

C. El Cubo OLAP de SQL

Una buena definición del término OLAP fue dada por el Concilio OLAP [8] "'*On-line Analytical Processing' es una categoría de tecnología de software que permite a los analistas, gerentes y ejecutivos obtener ventaja de la data a través de acceso interactivo rápido y consistente a una amplia variedad de vistas posibles de información que ha sido transformada a partir de data cruda para reflejar la dimensionalidad real de la empresa ala manera como es entendida por el usuario*".

El SQL estándar [5] provee un operador OLAP conocido como el cubo OLAP cuya forma más sencilla de uso es la en una consulta de la forma **SELECT** $d_1,\ldots,d_k$, $f_1(m_n),\ldots, f_n(m_n)$ **FROM** *r* **GROUP BY CUBE** $(d_1, \ldots, d_k)$, donde: — *r* es una relación en el sentido más amplio del concepto (puede ser una relación básica o una vista o cualquier combinación de éstas incluso con cláusula **WHERE**); — $d_1,\ldots,d_k$ son atributos de *r*, usualmente representando claves dimensionales o atributos de niveles dimensionales; — $f_1(m_n),\ldots,f_n(m_n)$ son funciones de agregación $f_1,\ldots, f_n$ (**COUNT**, **AVG**, **MIN**, **MAX**) aplicadas respectivamente a $m_n,\ldots, m_n$ que son atributos de *r*, usualmente conteniendo las medidas de la relación multidimensional. La semántica de esta consulta es producir una relación con los cálculos de las funciones de agregación $f_1(m_n),\ldots,f_n(m_n)$ para los grupos de filas resultantes de particionar la relación *r* por los valores de los atributos $d_1,\ldots,d_k$ y todas las posibles combinaciones de subconjuntos de ellos. Estas distintas agrupaciones forman un cubo abstracto como el de la figura 7, a ello se debe el nombre del operador.

De acuerdo al estándar SQL pueden hacerse consultas más complejas con el operador cubo OLAP, sin embargo, sin pérdida de generalidad, a efectos de este artículo, nos concentraremos en esta su forma más sencilla del uso del operador, de manera tal que la presentación del aporte sea lo más clara posible.

Vamos a ilustrar esto con un ejemplo sencillo. Consideremos la relación de ventas en la Tabla I. En la figura 7 se presenta un cubo OLAP abstracto que muestra gráficamente la semántica del operador CUBE de SQL con los valores de la Tabla II que es el resultado de la consulta **SELECT** *producto, tienda* **SUM**(*venta*) **FROM** *ventas* **GROUP BY CUBE** (*producto, tienda*).

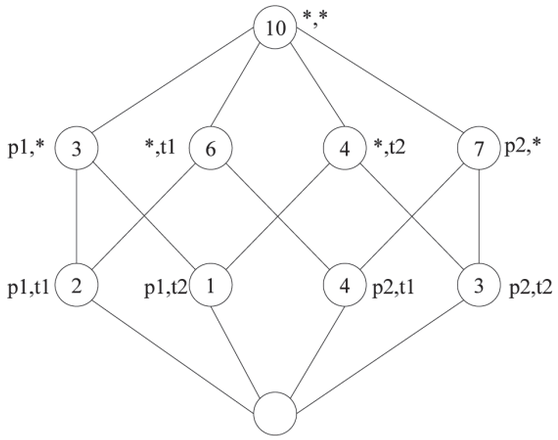

Fig. 7 Cubo abstracto OLAP.

TABLA I

Relación de ventas (en millones) por producto y tienda

Producto	tienda	venta
p1	t1	2
p1	t2	1
p2	t1	4
p2	t2	3

TEKHNE.indb 16 1/2/10 09:11:48

TABLA II
Resultado de consulta con cubo OLAP

producto	tienda	venta
p1	t1	2
p1	t2	1
p1		3
p2	t1	4
p2	t2	3
p2		7
	t1	6
	t2	4
		10

D. Etiquetas Lingüísticas en Particiones Difusas

Los conjuntos difusos fueron propuestos por Zadeh [11] como una forma de representar la imprecisión y la incertidumbre, y su motivación inicial eran las aplicaciones de sistemas de control, pero con el tiempo se comenzaron a usar en predicción y optimización, reconocimiento de patrones y sistemas expertos. En los conjuntos clásicos la pertenencia de un elemento a un conjunto es rígida, definida por una función indicatriz cuyo rango es {0,1}, el 0 representa la exclusión, mientras que el 1 la inclusión. Un conjunto difuso *F* en un universo *X* admite pertenencia gradual definida por una función de membresía μ_F cuyo rango es el intervalo real [0,1], permitiendo no sólo elementos incluidos y excluidos, sino también elementos parcialmente incluidos, aquéllos cuyo grado de membresía está en (0,1). Al conjunto formado por estos elementos de les conoce como el *borde* del conjunto difuso. El conjunto de los elementos completamente incluidos se les llama el *núcleo*. Los elementos que no están completamente excluidos conforman el *soporte*.

La función de membresía de un conjunto difuso puede definirse de distintas formas. En caso que el conjunto difuso sea definido sobre un universo numérico ordenado, la representación más sencilla y usual de la función de membresía es la forma trapezoidal (figura 8), la cual se especifica simplemente con una cuatrupla (x_1,x_2,x_3,x_4) de elementos ordenados del dominio $(x_1 \leq x_2 \leq x_3 \leq x_4)$que definen los vértices del trapecio $\{(x_1,0),(1,x_2),(x_3,1),(x_4,0)\}$.

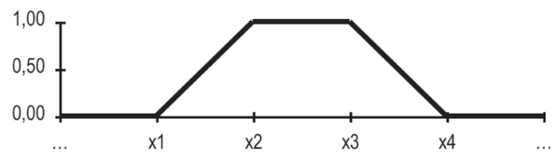

Fig. 8 Modelo de función de membresía trapezoidal

Los conjuntos difusos permiten dar una representación computacional a términos lingüísticos vagos, tales como los adjetivos calificativos *bueno, regular, malo, excelente, deficiente, paupérrimo, bajo, alto*, entre otros.

Un atributo o concepto que puede ser descrito por un conjunto de términos cualitativos se conoce como una variable lingüística [12]. Este tipo de variables son caracterizadas por un conjunto de etiquetas lingüísticas las cuales son representadas mediante conjuntos difusos en el universo del tipo base de la variable.

Un conjunto de etiquetas lingüísticas dotado de un orden total conforma un marco de cognición (Tudorie 2008). Adicionalmente, se dice que este conjunto forma una partición difusa cuando se cumplen dos condiciones: no vacuidad de los conjuntos difusos que definen las etiquetas; y que la unión de todos los conjuntos produce el universo. La unión de conjuntos difusos tiene distintas interpretaciones, en este contexto se interpreta así:

$$\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x).$$

La definición de partición difusa generaliza el concepto matemático de partición, relajando la condición de exclusividad. Esto es, la intersección de los conjuntos no tiene que ser vacía, pues en los bordes habrá intersección.

Algunas restricciones se imponen a las particiones difusas para que sean manejables y con una semántica clara [6]: — El número de elementos en un marco de cognición no puede ser muy grande pues dificultaría al usuario su comprensión y manipulación. — El orden total definido entre las etiquetas del marco de cognición debe ser consistente, esto es: si un elemento *a* está en el núcleo del conjunto *A* y otro *b* está en el núcleo del conjunto *B* y $a \leq b$ entonces debe cumplirse también $A \leq B$. — Todos los conjuntos de la partición deben ser normales, esto es, su núcleo no es vacío. — Todo los conjuntos difusos de la partición deben ser convexos, esto es si tenemos tres elementos *a*, *b* y *c* tales que $a \leq b \leq c$ entonces debe cumplirse $\mu_A(c) \geq min(\mu_A(a), \mu_A(b))$.

TEKHNE.indb 17 1/2/10 09:11:49

Una consecuencia directa de estas restricciones y de la definición de una partición difusa la condición de *complementaridad*, que un elemento cualquiera tiene sólo dos opciones: — estar exactamente en un conjunto, en este caso, estaría en el núcleo; o — estar exactamente en dos conjuntos de la partición, en este caso estará en el borde de ambos y el grado de membresía de uno es el complemento a uno del grado de membresía en el otro. Esto se evidencia claramente en el ejemplo de la figura 9.

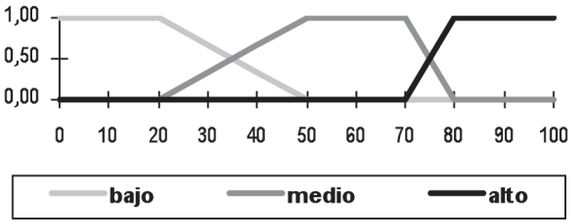

Fig. 9 Partición difusa (consistente) en [0,100]

Una partición difusa que cumpla con las precedentes restricciones diremos que es una *partición consistente* (como la de la figura 9), en caso de no cumplirla, diremos que es *inconsistente*. La figura 10, muestra una partición inconsistente. Por razones de sencillez, de aquí en adelante usaremos el término *partición difusa* para referirnos a una partición difusa consistente.

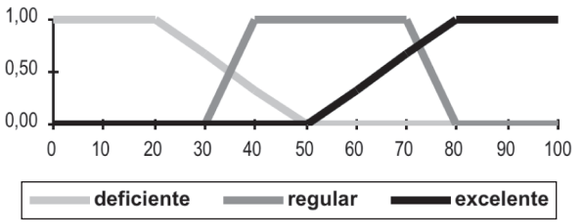

Fig. 10 Conjunto de etiquetas lingüísticas en el universo [0,100] que son inconsistentes como partición difusa. En los intervalos (20,50) y (50,80) no se cumple la condición de complementariedad.

Un conjunto difuso se define sobre un universo o dominio base. Sin embargo, un término lingüístico vago podría tener una interpretación adecuada en distintos dominios o rangos de valores. Por ejemplo, un total alto de ventas diarias de una tienda no es de la misma magnitud que un total de ventas altas de toda la cadena de tiendas para todo un año. Es por esto que Bordogna [2] ha propuesto la noción de predicados lingüísticos adaptables. La idea es que el predicado lingüístico se defina en base a un modelo de conjunto difuso en el universo del intervalo real [0,1], a este modelo se le añade una cota que define el rango del universo del predicado, además un operador de modificación. Luego, al usar el predicado se puede hacer un acercamiento o alejamiento del valor actual (zoom in/out) mediante el operador de modificación.

Veamos un ejemplo de este tipo de predicados. La figura 11 muestra el predicado *costoso* definido con la cota 300 y el modelo trapezoidal (0,25, 1, 1, 1) y el modificador *. El precio 225 tendría 0,66 como grado de membresía al predicado *costoso*. El precio 600 tendría 0,33 como grado de membresía al predicado *costoso* si este se aplica con un *zoom* de *0,25*, pues se toma el valor modificado 600*0,25 y se le aplica la función de membresía.

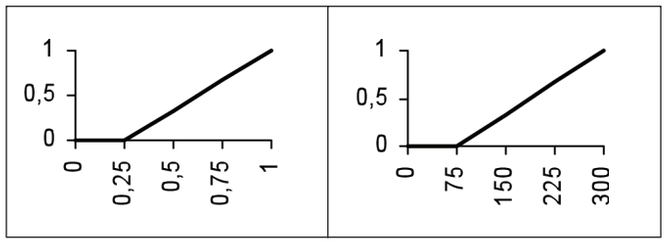

Fig. 11 Predicado adaptable costoso en el intervalo [0,300]. A la izquierda, modelo en el intervalo [0,1], a la derecha, predicado real en [0,300].

La noción de predicados ajustables es deseable para el problema que nos ocupa en este trabajo. Sin embargo, la propuesta de Bordogna [2] tiene una limitante, que el usuario tendría que especificar el parámetro del zoom por cada contexto. Si estamos haciendo una consulta de tipo cubo OLAP con k dimensiones tenemos 2^k contextos, pues cada combinación de dimensiones da un contexto diferente, así el número de contextos es igual al número de subconjuntos del conjunto de k dimensiones. Sin embargo, si los predicados no fueran ajustables el problema sería peor, pues en lugar de tener que dar 2^k parámetros de zoom, serían 2^k particiones difusas por cada medida en el cubo. Esto de cualquier manera requiere mucho de la intervención del usuario experto que tendría que saber a priori cuáles van a ser los rangos de los datos para cada contexto. Sería conveniente hallar una forma en que los predicados puedan ser auto-ajustables al contexto.

Existe una propuesta de definir particiones en las que el sistema puede inferir automáticamente las definiciones de los predicados según el contexto de la consulta [6]. El usuario sólo especifica el marco de cognición y luego al momento de la consulta el sistema calcula percentiles estadísticos de los datos en el contexto y define los predicados. El inconveniente de

TEKHNE.indb 18 1/2/10 09:11:50

esta propuesta es el volumen de trabajo que implica el calcular los percentiles para los 2^k contextos de un cubo OLAP.

Aquí nosotros proponemos una nueva forma de predicados auto-ajustables al contexto. El usuario debe especificar el conjunto de etiquetas que conforman una partición y definir el modelo de cada etiqueta en el intervalo [0,100], así como la partición difusa de la figura 9. El sistema debe ser capaz de ajustarlos automáticamente a cada contexto. Para ello tomaría el valor máximo del contexto como cota del rango que define el universo y el predicado se ajustará proporcionalmente en el contexto en que se usa. De esta manera la adaptación de los predicados resulta computacionalmente aceptable, pues sólo hay que calcular máximos. Así, por ejemplo, con la partición de la figura 9, si en un contexto el máximo valor es 5000, el valor 3750 sería *medio* y *alto* con grado 0,5, mientras que 2000 sería *medio* con grado 0,66 y *bajo* con 0,33.

3. Definición de la Metodología

A partir de una base de datos fuente cómo se puede llegar a hacer consultas de tipo cubo OLAP que den los resultados en términos lingüísticos. Para ello definimos acá una metodología en dos etapas: Construcción de diseño lógico multidimensional para datos precisos y Extensión del diseño lógico relacional multidimensional con etiquetas lingüísticas.

La primera etapa esta compuesta por siete actividades que permiten conocer el caso de estudio, definir indicadores de gestión para el área involucrada, modelar el diseño conceptual de los indicadores, establecer la fuente de información, realizar un mapeo entre los indicadores y la base de datos fuente, y finalmente construir el cubo OLAP.

La segunda etapa desarrolla tres actividades principales: definición de las etiquetas lingüísticas que extenderán el cubo OLAP, el diseño de las consultas imprecisas, y finalmente una comparación y análisis de resultados entre consultas precisas y consultas imprecisas. En la figura 12 se presentan gráficamente esta secuencia de pasos, los cuales se explican en detalle en los apartados siguientes.

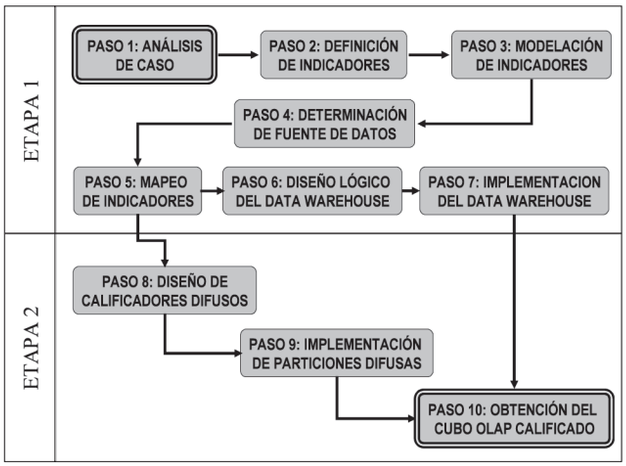

Fig. 12 Diagrama de la Metodología Propuesta

A. Paso 1: Análisis del Caso de Aplicación

Lo primero es tener un conocimiento adecuado de la organización para la cual se desean hacer las consultas OLAP. Hay que buscar documentos que definen a la organización, sus objetivos, misión y visión. Se necesita contactar con el personal gerencial de la organización y conocer sus necesidades de información para los procesos de toma de decisión. También hay que conocer cuáles son los sistemas que tienen o han tenido en operación de los cuales se podrán obtener las fuentes de datos. El resultado de este análisis sería la definición de los requerimientos de información.

B. Paso 2: Definición de Indicadores de Gestión

En esta actividad se definen los indicadores de gestión del caso de aplicación. Estos indicadores nos servirán para medir el cumplimiento de las metas y alcances de la corporación. Corresponden a instrumentos de medición del comportamiento y/o tendencias de las metas para una determinada área estratégica de la empresa. Cada indicador se representa con medidas tangibles que pueden ser de tres tipos: completamente aditivas, pues representan un valor asociado a un período de tiempo y se pueden sumar (por ejemplo, las ventas de tres meses consecutivos pueden sumarse para obtener las ventas totales del trimestre); semi-aditivas, medidas que representan un momento particular en el tiempo y su sumatoria no representaría un valor con sentido (por ejemplo, los inventarios de tres meses consecutivos no se pueden sumar para obtener el inventario "total" del trimestre), sin embargo, la sumatoria puede tener sentido en otra dimensión (por ejemplo, es posible sumar inventarios de ubicaciones diferentes en un momento determinado

TEKHNE.indb 19 1/2/10 09:11:50

para obtener el inventario total); y no aditivas, cuando son medidas que no implican ninguna sumatoria, pero resultan útiles cuando se desea contabilizar ítems (por ejemplo, contar el número de códigos postales diferentes a los que se envío un producto). Al finalizar este paso se tiene la lista de indicadores con sus medidas asociadas.

C. Paso 3: Modelación de Indicadores de Gestión

Una vez identificados los indicadores de gestión, para cada uno de ellos es necesario diseñar un modelo multidimensional. Especificar las medidas, niveles, dimensiones y relación dimensional. Para ello adoptamos el modelo CMDM propuesto por Carpani [3].

D. Paso 4: Determinación de Fuentes de Datos

Esta actividad consiste en determinar la base de datos operacional que será usada como fuente de datos primaria del Data Warehouse. Puede tomarse una o varias de las bases de datos existentes en la organización y otras fuentes de información disponibles que sean relevantes, de acuerdo con los elementos de cada modelo multidimensional realizado en el paso anterior. Si se usa más de una base de datos, éstas deben integrarse al menos a nivel de diseño. Se requiere tener un diagrama lógico de la base de datos integrada, dotado de un diccionario de datos donde se explique bien la semántica y se coloquen restricciones no representadas en el diagrama.

E. Paso 5: Mapeo Fragmentado de Indicadores

Para cada uno de los indicadores de gestión hay que establecer una correspondencia entre su modelo multidimensional y su base de datos fuente determinada. El propósito de este paso es establecer una correspondencia entre la base de datos fuente y los indicadores de gestión determinados en los pasos previos. Esto permitirá definir el futuro cubo de datos. Para cada indicador de gestión, se define un mapeo entre las medidas y dimensiones con uno o más atributos de la base de datos según propone Peralta [9].

F. Paso 6: Diseño Lógico del Data Warehouse

En este paso de nuestra metodología, se completa el proceso propuesto por Peralta [9], para obtener el esquema relacional multidimensional que modela el cubo de datos. Este esquema puede resultar en un modelo estrella y/o copo de nieve según las exigencias de cada indicador. Los requerimientos de usuario, las restricciones de plataforma y de más rasgos del sistema deben ser tomados en cuenta.

G. Paso 7: Implementación del Data Warehouse

En este paso se selecciona un sistema manejador de bases de datos adecuado. Debe tomarse en cuenta las dimensiones de la data con que se va a trabajar. También debe ser un manejador cuya versión de SQL soporte los operadores OLAP. El DW debe ser creado de acuerdo con su modelo lógico y poblado a partir de la base de datos fuente. Hasta este punto lo que hemos hecho es preparar los datos para que puedan ser analizados, aún no hemos integrado los términos lingüísticos, lo cual se hará en los pasos siguientes.

H. Paso 8: Diseño de Calificadores Difusos

Para cada uno de los indicadores debe definirse un conjunto de etiquetas lingüísticas que formarán la partición difusa apropiada al indicador. Se debe hacer el modelo de cada uno de los predicados mediante funciones trapezoidales en el intervalo [0,100]. Los modelos deben cumplir todas las restricciones para ser una partición difusa consistente. Esos modelos se hacen una sola vez por cada indicador, pues haremos que el sistema sea capaz de adaptarlos al contexto de acuerdo con nuestra propuesta presentada más atrás en este artículo, en la sección titulada “Etiquetas Lingüísticas en Particiones Difusas”.

I. Paso 9: Implementación de Particiones Difusos

Las particiones difusas deben guardarse en una tabla en la base de datos. En realidad el contenido de esta tabla forma parte de la meta-data del esquema, pues en la tabla se encontrarán las definiciones de predicados que describen a las medidas. El esquema relacional de esta tabla es *fuzzyPartition*(*measure, label, x1, x2, x3, x4*), donde la columna *measure* contendrá los nombres de las medidas de los indicadores, la columna *label* contendrá por cada medida las etiquetas que conforman la partición y para cada medida y etiqueta *x1, x2, x3, x4* serán los valores que especifican el conjunto difuso que modela a la etiqueta como un predicado auto-ajustable en el intervalo [0,100].

Adicionalmente, hay que almacenar en el manejador de base de datos la función de cálculo del grado de membresía para los predicados auto-ajustables. Tabla III contiene el código en SQL estándar para la implementación de esta función, la cual recibe como parámetros el valor *x* al cual calcular el grado de membresía, límite superior *u* del rango del universo según el contexto y los valores *x1*, *x2*, *x3* y *x4* que definen la función de membresía trapezoidal que modela el predicado difusos en el intervalo [0,100].

TEKHNE.indb 20 1/2/10 09:11:50

TABLA III

Código en sql para el cálculo del grado de membresía en un predicado autoajustable

```
CREATE FUNCTION membership(
        x,u,x1,x2,x3,x4 NUMERIC
) RETURNS NUMERIC AS
BEGIN
        x = (x/u)*100;
        IF x1<x AND x<x2 THEN RETURN (x-x1)/(x2-x1)
        ELSE IF x2<=x AND x<=x3 THEN RETURN (1)
        ELSE IF x3<x AND x<x4 THEN RETURN (x4-x)/(x4-x3)
        ELSE RETURN (0);
END;
```

J. Paso 10: Obtención del Cubo OLAP Calificado

El último paso es generar para cada indicador el cubo OLAP calificado con etiquetas lingüísticas. Esto se hace con el procedimiento definido en la Tabla IV. El bloque de código lo hemos especificado para la forma más sencilla y general del cubo OLAP en el estándar SQL, su adecuación a formas más complejas es directa a partir de estas especificaciones. Preferimos mantenerlo lo más simple posible para facilitar su comprensión.

Primero se obtienen el cubo OLAP tradicional anotando en cada fila a qué contexto pertenece. Este resultado queda en la tabla temporal cubeView. Las anotaciones del contexto en los atributos g1,..,gk de esta tabla que se calculan mediante la operación GROUPING que es una función predefinida en el estándar SQL que retorna 1 cuando se está agrupando por el atributo que se especifica como parámetro y 0 cuando no.

TABLA IV

Pseudo código en sql para la obtención del el cubo olap calificado difuso

```
BEGIN

        CREATE TEMPORAL TABLE cubeView AS
        SELECT keyDim1,..,keyDimk, aggFun(mesure) AS aggMes,
        GROUPING(keyDim1) AS g1,.., GROUPING(keyDimk) AS
gk
        FROM dataWarehouse
        GROUP BY CUBE (keyDim1,..,keyDimk);

        CREATE TEMPORAL TABLE rangeBound AS
        SELECT g1,..,gk, MAX(aggMes) AS upperBound
        FROM cubeView
        GROUP BY g1,..,gk;

        SELECT keyDim1,..,keyDimk, aggMes, label,
        membership(v.aggMes,r.upperBound,p.x1,p.x2,p.x3,p.x4) AS
degree
        FROM cubeView AS v,rangeBound AS r,fuzzyPartiction AS
p
        WHERE p.measure = measureName
        AND v.g1 = r.g1 AND .. AND b.gk = r.gk
        AND (v.aggMes/r.upperBound)*100 BETWEEN p.x1 AND
p.x4

END;
```

Veamos en qué consiste la primera instrucción en el bloque de la Tabla IV: Se obtiene el cubo OLAP tradicional anotando en cada fila a qué contexto pertenece. Este resultado queda en la tabla temporal cubeView. Las anotaciones del contexto se colocan en los atributos g1,..,gk de esta tabla que se calculan mediante la función GROUPING predefinida en el estándar SQL que retorna 1 cuando se está agrupando por el atributo que se especifica como parámetro y 0 cuando no. Las variables: — *keyDim1,..,keyDimk* representan los nombres de los atributos que contienen las claves de las dimensiones; — *aggFun* representa la función de agregación que se usa sobre la medida del indicador; — *mesure* representa el nombre del atributo en que se encuentra la medida que se está analizando; — *aggMes* representa el nombre de atributo con que se visualizará el resultado de la medida calculada; — *dataWarehouse* representa la tabla dimensional o el JOIN de ella con las tablas dimensionales al cual se le puede además haber aplicado una operación de cambio de nivel, conocida como ROLL UP.

La segunda instrucción en el procedimiento de la Tabla IV tiene el propósito de obtener el límite superior de los datos en cada uno de los 2^k contextos del cubo OLAP almacenado en la tabla temporal cubeView. Estos límites se registran en una nueva tabla temporal llamada rangeBound que tiene la clave del contexto g1,..gk y el el atributo upperBound que es el límite determinado.

Finalmente, la tercera instrucción en el bloque de la Tabla IV combina el cubo, el contexto y las particiones difusas para producir el cubo OLAP calificado difuso. Éste tendrá la clave dimensional *keyDim1,..,keyDimk*, la medida calculada *aggMes*, la etiqueta difusa correspondiente label y el atributo calculado degree que es el grado de membresía de la medida a al predicado auto-ajustable de la etiqueta, según el contexto. Se verifica en la condición de la consulta que la etiqueta sea definida para la medida, para ello se usa la variable *measureName* que tendrá el nombre de la medida que se está analizando. También se verifica que el valor calculado de la medida caiga en el soporte del predicado ajustado al contexto.

Nótese que una combinación *keyDim1,..,keyDimk*, *aggMes* podría aparecer dos veces en el cubo OLAP calificado difuso. Esto ocurre cuando el valor de *aggMes* cae en el borde de dos conjuntos difusos contiguos de la partición, en este caso los grados de membresía de las dos filas serán complementarios, es decir sumarían 1. De acuerdo a las restricciones impuestas no puede

TEKHNE.indb 21 1/2/10 09:11:51

aparecer más de dos veces y si aparece una vez es porque está en el núcleo de un conjunto difuso, luego su grado de membresía sería 1.

El cubo OLAP es una tabla que frecuentemente resulta muy grande para ser visualizada directamente por el usuario, de manera que puede preferirse aplicar algunos filtros. Si los filtros son precisos, éstos podrían colocarse en la primera instrucción de la Tabla IV simplemente añadiendo una cláusula **WHERE** o una cláusula **HAVING** con su modo de empleo y semántica habitual del estándar SQL.

Algo novedoso es que también podemos establecer filtros difusos basados en los términos lingüísticos. Por ejemplo un usuario podría estar interesado sólo en conocer cuáles son las agrupaciones dimensionales en que el indicador de gestión analizado está arrojando una medida *baja*, siendo *baja* una etiqueta lingüística en la partición definida para esa medida. En casos como estos el filtro se colocaría en la cláusula **WHERE** de la tercera instrucción de la Tabla IV y se haría sobre el atributo *label*.

También al usuario podrían interesarle ciertos niveles de satisfacción aceptable (*threshold*). Por ejemplo, si en una agrupación, la medida se califica con alguna etiqueta con un grado de satisfacción 0.3 y se ha establecido que el *threshold* es estrictamente por encima de 0.5, entonces esa etiqueta no aparecería en el resultado para esta agrupación dimensiona, sino que aparecería la que aplica de forma complementaria cuyo grado sería 0.7. Para este tipo de restricciones el filtro se establece sobre el atributo *degree* en la cláusula WHERE de de la tercera instrucción de la Tabla IV.

4. Desarrollo de un Caso de Estudio

A fin de ilustrar los aportes presentados en este artículo, a continuación desarrollamos un caso de estudio con la metodología propuesta.

A. Paso 1: Análisis del Caso de Aplicación

Se tomó el caso real de una empresa chilena de ventas al mayor y detal, por razones de confidencialidad no damos su identidad y algunos nombres fueron cambiados en los ejemplos a mostrar. Esta empresa tiene una importante pero no masiva participación en el mercado. Sus objetivos actuales están relacionados con porcentajes de crecimiento, expansión demográfica y utilidades netas, por lo que potenciar el uso de nuevas herramientas de gestión y control, representa un desafío importante para la compañía.

La empresa comercializa sus productos a nivel nacional, teniendo mayor fuerza y presencia en la región metropolitana, razón por lo cual se definió este alcance geográfico en la muestra de los datos a trabajar. La empresa posee dos plantas productivas, un centro de distribución y una cadena de cincuenta tiendas, agrupadas en sectores dentro de la región metropolitana. La empresa vende tanto sus productos como los de terceros a supermercados, restaurantes, clientes minoristas, entre otros.

Se quiere apoyar la visión de la gerencia de ventas al detal, la cual depende de la gerencia comercial, cuyo objetivo es administrar y velar por el cumplimiento de las metas de venta de toda la cadena de tienda de la compañía. Esta gerencia es la responsable de la definición, control y seguimiento de los indicadores de gestión que garanticen sus objetivos.

Cada año la gerencia de ventas al detal en conjunto con la gerencia comercial y la gerencia general, debe definir los objetivos, metas, desafíos, proyectos y planes de acción para facilitar la gestión y resultados exitosos de la cadena de tiendas de la compañía. Estas metas deberán determinar la evolución de las ventas acumuladas y el análisis de mercado en diferentes dimensiones (por tienda, por sector, por familia o subfamilia de productos).

Se quiere desarrollar un sistema de apoyo a la gestión que esté orientado a resumir la información sobre las ventas en la región metropolitana, de manera cualitativa y cercana al usuario. La información sobre las ventas puede recuperarse a partir del sistema de puntos de ventas. Cuando decimos que se quiere resultados cualitativos nos referimos, por ejemplo a decir si el volumen de ventas fue “bajo” o “medio” o “alto”.

B. Paso 2: Definición de Indicadores de Gestión

Los indicadores fueron obtenidos a partir de las necesidades definidas por los ejecutivos de la empresa. Posteriormente se seleccionará uno sobre el cual se desarrollarán el resto de los pasos de la metodología, ya que considerarlos todos puede ser repetitivo y muy extenso. Todas las medidas obtenidas son aditivas.

1. Ventas y participación porcentual

Este indicador mide para cada punto de venta de la región metropolita, su total de ventas y su participación porcentual, dentro del total de ventas de todas las tiendas o de todo un sector. Las dimensiones para agrupar la información son la ubicación (tienda o sector) y el periodo (mes o año). Se considerarán dos

TEKHNE.indb 22 1/2/10 09:11:52

medidas: venta acumulada y participación porcentual. La participación se calcula como la sumatoria de ventas de una tienda entre el total de ventas de todas las tiendas (o todo un sector), para un período dado.

2. Análisis de Rentabilidad

Este indicador mide la rentabilidad de una tienda y/o sector, para esto se requiere conocer las ventas y costos acumulados. Las dimensiones para agrupar la información son la ubicación (tienda o sector) y el período (mes o año). Se considerarán tres medidas: venta acumulada, total de costos y rentabilidad. La rentabilidad se calcula como el cociente entre la venta acumulada y el costo de venta.

3. Participación de las formas de pago

Aquí se quiere calcular el porcentaje de participación en las diferentes formas de pago registradas en los puntos de venta de una tienda y/o un sector. Estas formas de pago pueden ser efectivo, tarjetas de débito, tarjetas de crédito, cheques y ticket. Las dimensiones para agrupar la información son la ubicación (tienda o sector) y el período (mes o año). Se considerarán tres medidas: ventas acumuladas, ventas por forma de pago y participación en la forma de pago. La participación se calcula como el cociente entre el total de una forma de pago dada, y el total de ventas.

4. Venta y participación de proveedores

El indicador mide las ventas y porcentaje de participación de cada familia de proveedores. En este caso de estudio existen tres familias de proveedores: Frutale, Ensenada y Tercero. Las dimensiones para agrupar la información son la ubicación (tienda o sector) y el período (mes o año). Se considerarán dos medidas: ventas acumuladas y participación del proveedor en las ventas. Las ventas acumuladas se calcularán por tienda, por sector, por proveedor de tienda y por proveedor de sector. La participación se calcula como el cociente entre el total de ventas para una familia de proveedores de una tienda (o sector) entre el total de venta de dicha tienda (o sector).

5. Análisis ventas/metas

Aquí se quiere medir las metas de ventas contra el porcentaje de cumplimiento de ventas. Las dimensiones para agrupar la información son la ubicación (tienda o sector), el período (mes o año), la categoría de productos y los proveedores. Se considerarán tres medidas: ventas acumuladas, metas previstas y cumplimiento de ventas. El cumplimiento de ventas se calcula como el cociente entre la venta acumulada y la meta para ese rubro.

C. Paso 3: Modelación de Indicadores de Gestión

En este paso se obtendrá el diseño conceptual multidimensional que representa cada indicador de gestión. Para efectos de este artículo sólo presentaremos el indicador "Análisis ventas/metas". La figura 13 se muestra el diseño de los niveles correspondientes a las dimensiones: Producto (producto y categoría), Ubicación (tienda y sector), Tiempo (mes y año), Proveedor (proveedor). Los niveles determinados se jerarquizan en cada dimensión de acuerdo al diagrama de la figura 14. Finalmente, el diagrama de la relación multidimensional se muestra en la figura 15. Puede observarse que hay un globo que representa la relación el cual tiene un satélite por cada dimensión y uno por cada medida, éstos últimos identificados por puntas de flecha.

D. Paso 4: Determinación de Fuentes de Datos

La fuente de datos proviene principalmente del sistema de puntos de venta (POS) instalado en la cadena de tiendas, desarrollado con tecnología JAVA, el cual se caracteriza por su portabilidad, parametrización, flexibilidad, y escalabilidad. Esta aplicación, se encarga de registrar localmente cada venta junto con su forma de pago. Cuando el procesador del POS detecta la ausencia de transacciones de ventas envía al servidor central la información acumulada hasta el momento. En este servidor se localiza el repositorio principal desde donde se tomarán los datos necesarios para realizar el análisis de gestión.

Fig. 13 CMDM: Niveles para el indicador "Análisis ventas/metas"

TEKHNE.indb 23 1/2/10 09:11:52

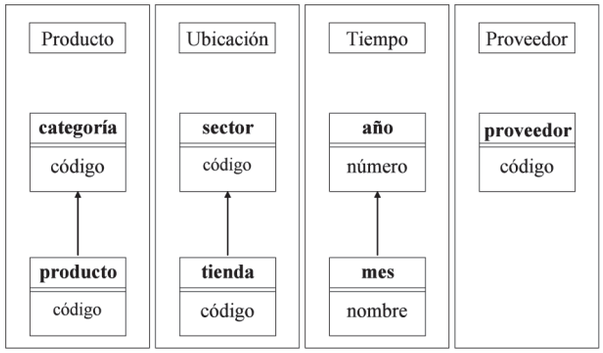

Fig. 14 CMDM: Dimensiones para el indicador "Análisis ventas/metas"

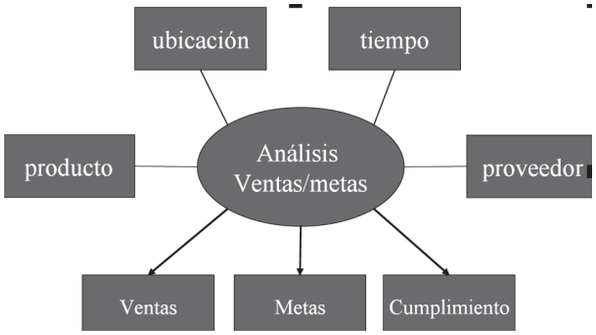

Fig. 15 CMDM: Relación dimensional para "Análisis ventas/metas"

Entre las funcionalidades que realiza el sistema POS se destacan la descripción y parametrización del POS; el manejo de usuarios y privilegios; el manejo de formas de pago, monedas e impuestos; actualización de listas de precios; clasificación de departamentos, clasificación de artículos; códigos de barra; visualización de videos; manejo de tarjetas de débito y crédito, entre otros.

La base de datos relacional del sistema POS es bastante completa y algo compleja. Sin embargo ésta no tiene información sobre algunos datos relevantes para nuestro análisis, como son los proveedores y las metas. Hay un sistema de control de suministro e inventario que sí tiene información de los proveedores, tenemos que tomar esta información de allí. Las metas no se encuentran en un sistema en línea, sino que los gerentes las manejaban como hojas de cálculo personales en su puesto de trabajo, estas requieren ser llevadas a una tabla relacional.

Se hizo la integración de las tres fuentes de datos disponibles. La figura 16 muestra un fragmento representativo del diagrama relacional de la base de datos resultante. Para simplificar su visualización, se han omitido muchos de los atributos y varias de las tablas que conforman la base de datos, sólo se muestran los más relevantes para nuestro propósito.

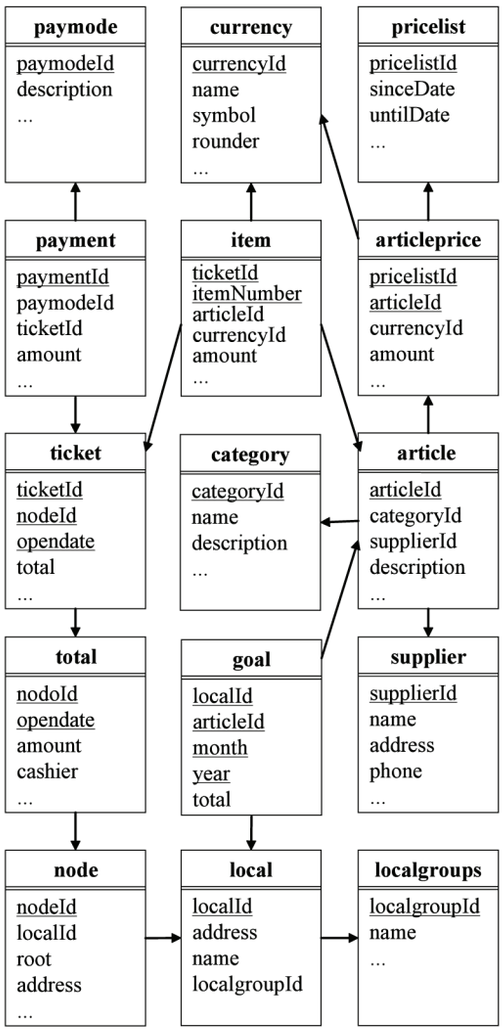

Fig. 16 Esquema relacional integrado de las fuentes de datos

E. Paso 5: Mapeo Fragmentado de Indicadores

En este paso se observa como corresponden los objetos del modelo conceptual definido a través de CMDM para el indicador de gestión seleccionado "Análisis ventas/metas" con la base de datos fuente. Esta correspondencia se muestra en la figura 16 para los niveles dimensionales y para las medidas de la relación dimensional obtenidos en los pasos anteriores.

Para el nivel dimensional *categoría* se buscaron en la base de datos cuáles eran los campos que correspondían con los atributos *código* y *nombre*. Se encontró dicha correspondencia en las columnas *categoryId* y *name* de la tabla *category*. Para el nivel *producto* los atributos *código* y *nombre* corresponden a las columnas *articleId* y *description* de la tabla *article*. Para el caso de los atributos *número* del nivel *año* y *nombre* del nivel *mes*, no existe una correspondencia

TEKHNE.indb 24 1/2/10 09:11:53

directa, sino que se extraen del campo *opendate* de la tabla *ticket*, por esta razón en el modelo aparecen con líneas punteadas. Los atributos *código*, *nombre* y *dirección* del nivel *tienda* corresponden a las columnas *localId*, *name* y *address* de la tabla *local*. Los atributos *código* y *nombre* del nivel *sector* corresponden a las columnas *localgroupId* y *name* de la tabla *localgroups*. Finalmente, los atributos *código* y *nombre* del nivel *proveedor* corresponden a los campos *supplierId* y *name* de la tabla *supplier*. Para todas las correspondencias directas se utilizan líneas continuas. En el caso de las medidas que aparecen en la relación dimensional se realizó una búsqueda similar y se llegó a lo siguiente: — En cuanto a la medida *Cumplimiento*, ésta es una medida no aditiva, es decir, para un nivel jerárquico su valor no se obtiene como la agregación de los valores en el nivel anterior. Ella se obtiene como el cociente de las otras dos medidas, de manera que no se encuentra en la base de datos fuente. Para ella se especifica en el diagrama su fórmula (*Ventas*/*Metas*). — Para la medida *Ventas*, la especificación es más compleja. En el diagrama se coloca una flecha segmentada hacia campo *amount* de la tabla *item*, indicando así que se calcula en base a este campo. Fijada una tienda, un producto, un proveedor, un mes y un año, la medida *ventas* se calcula mediante la sumatoria de los *amount* de la tabla *item* que correspondan a ese *producto* y que se encuentren en *tickets* de la *tienda* en ese *año* y *mes*. La especificación formal de este cálculo sería un poco compleja, por lo que preferimos dar en la Tabla V una especificación en SQL. — El caso de la medida *Metas* es más sencillo, pues se tiene una correspondencia directa con el campo *total* de la tabla *goal*.

TABLA V

Código en sql para el cálculo de la columnas ventas

```
SELECT SUM(amount) AS Ventas FROM
   item AS i, ticket AS t, node AS n, local AS c,
WHERE
   i.ticketId=t.ticketId AND t.nodeId=n.nodeId AND n.localId=c.localId
GROUP BY
   c.localId, i.articleId, MONTHS(t.opendate), YEAR(t.opendate)
```

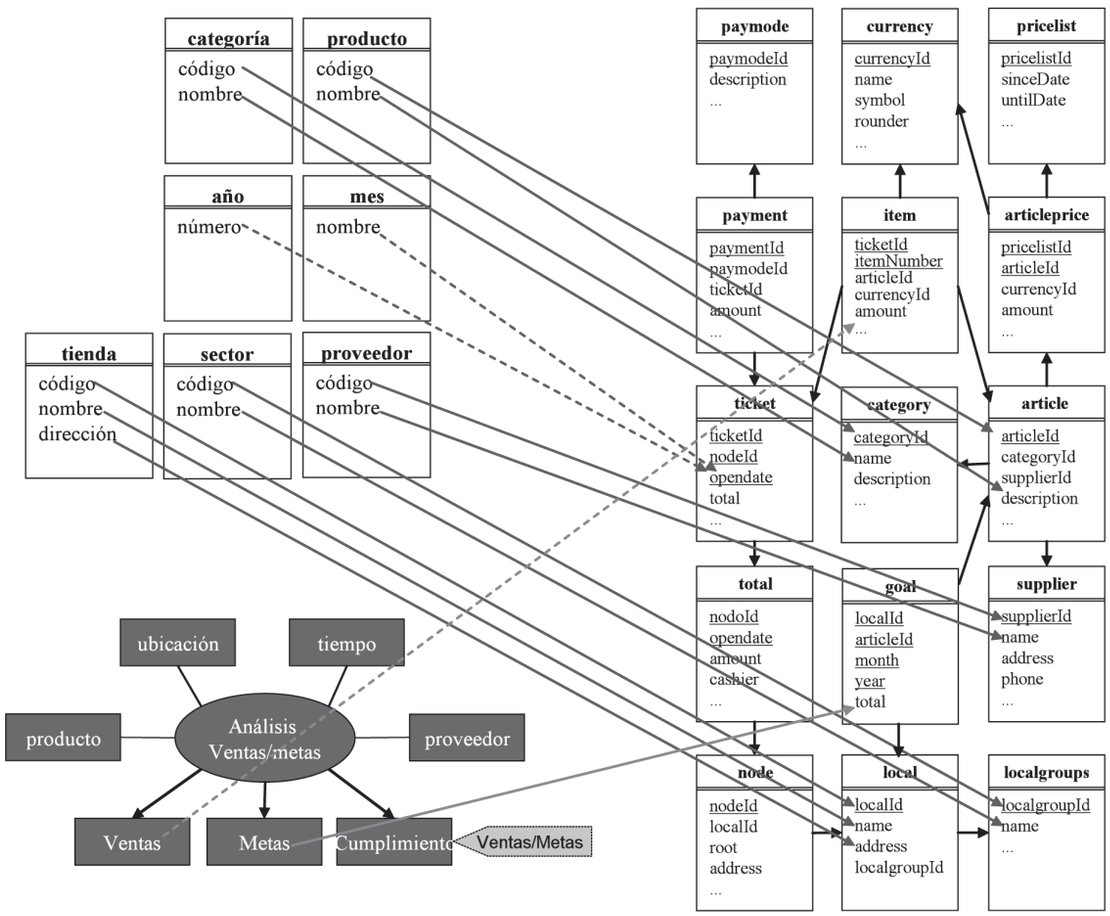

Fig. 17 Mapeo fragmentado del indicador "Análisis ventas/metas"

TEKHNE.indb 25 1/2/10 09:11:54

F. Paso 6: Diseño Lógico del Data Warehouse

A partir del mapeo fragmentado, mediante el proceso de diseño propuesto por Peralta [9], se obtiene el esquema relacional del Data Warehouse, el cual se muestra en la figura 17. Se puede observar que el esquema es una estrella con dos copos de nieve. La dimensión *Proveedor* tiene un solo nivel, por lo que no tiene sentido hacer un copo de nieve para ella. Para la dimensión *Tiempo* es preferible no hacer el copo de nieve pues la tabla para el *mes* necesariamente tiene que incluir el *número* del *año* que es el único atributo de ese nivel. Las dimensiones *Ubicación* y *Producto* se implementan mediante el esquema copo de nieve a fin de evitar redundancia. El esquema estrella representa las cuatro dimensiones *Período*, *Ubicación*, *Producto* y *Proveedor*. La tabla de hechos en el centro de la estrella se construye con las claves primarias cada una de las dimensiones, además de los atributos *ventas* y *metas*, que representan las dos medidas principales del indicador. La medida *cumplimiento* no se incluye por ser un atributo calculado.

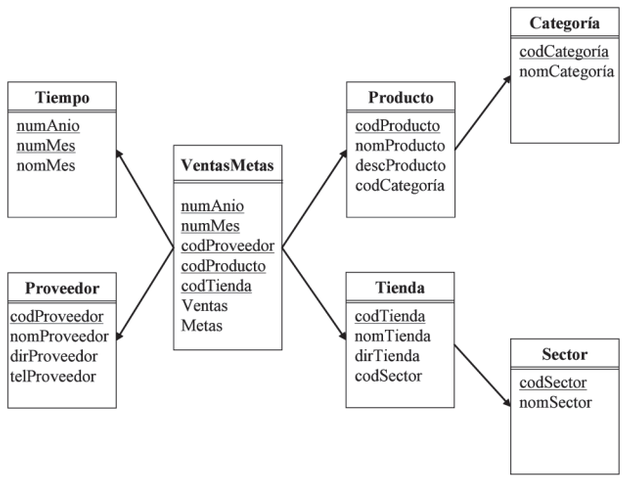

Fig. 18 Modelo de datos del repositorio (DataWarehouse)

G. Paso 7: Implementación del Data Warehouse

Se trabajó en la implementación con SQL Server 2000 por ser el gestor de bases de datos disponible en la organización por la cual se hizo la aplicación. Este manejador implementa los operadores de OLAP de SQL estándar con ciertas variaciones sintácticas. Una virtud de SQL Server es que tiene herramientas que asisten en el análisis de Data Warehouse. Sin embargo, para efectos del artículo preferimos presentar el código en SQL estándar. De acuerdo al esquema y al mapeo, se migró la data de la base de datos fuente al Data Warehouse.

H. Paso 8: Diseño de Calificadores Difusos

Se realizó el diseño de las etiquetas a utilizar en el cubo OLAP calificado. El hecho de especificar los modelos en el intervalo [0,100] resulta muy natural para el usuario habituado a manejar porcentajes. Se determinaron tres etiquetas para cada una de las tres medidas en el análisis ventas/metas. Estas etiquetas expresan la idea de datos que están cercanos a los extremos superior e inferior del rango y datos que están cerca del centro. Sin embargo, los modelos dados para las tres medidas fueron diferentes de acuerdo a las preferencias del usuario. Las etiquetas lingüísticas para *Ventas* serán *bajas*, *medias* y *altas*, sus modelos se muestran en la figura 19. Las etiquetas lingüísticas para *Metas* serán *pesimista*, *conservadora* y *optimista*, sus modelos se muestran en la figura 20. Las etiquetas lingüísticas para *Cumplimiento* serán *deficiente*, *regular* y *excelente*, sus modelos se muestran en la figura 21.

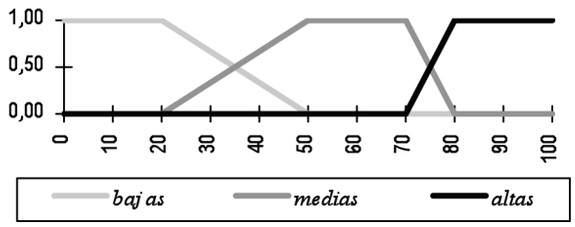

Fig. 19 Modelo de la partición difusa para la medida Ventas

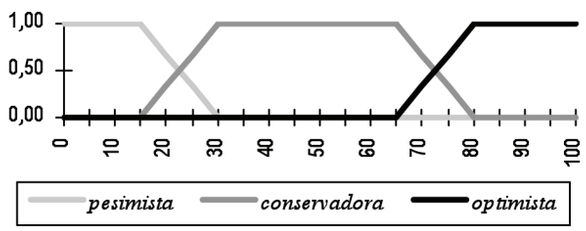

Fig. 20 Modelo de la partición difusa para la medida Metas

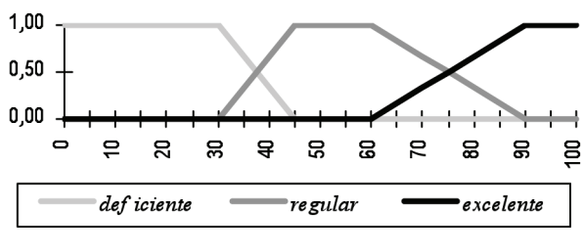

Fig. 21 Modelo de la partición difusa para la medida Cumplimiento

TEKHNE.indb 26 1/2/10 09:11:55

I. Paso 9: Implementación de Particiones Difusas

Creamos la función *membership* según la Tabla III y la tabla *fuzzyPartition* con los datos de la Tabla VI.

TABLA VI

Modelo de las particiones difusas

measure	label	x1	x2	x3	x4
Ventas	bajas	0	0	20	50
Ventas	medias	20	50	70	80
Ventas	altas	70	80	100	100
Metas	pesimista	0	0	15	30
Metas	conservadora	15	30	65	80
Metas	optimista	65	80	100	100
Cumplimiento	deficiente	0	0	30	45
Cumplimiento	regular	30	45	60	90
Cumplimiento	excelente	60	90	100	100

J. Paso 10: Obtención del Cubo OLAP Calificado

Como es habitual para un Data Warehouse, el volumen de datos en el caso de estudio es muy grande, de manera que la combinatoria que daría el hacer el cubo OLAP es imposible de mostrar en este artículo. Se hace necesario restringir los datos sobre los cuales hacer las consultas OLAP.

A efectos del ejemplo a mostrar, nos restringimos a un solo proveedor cuyo nombre es *FRUTALE*, dos tiendas de nombres *ALG* y *MEGA*, dos categorías de productos de nombres *GRANEL* y *PANPITA*, un año, el 2007 y tres meses *julio, agosto* y *septiembre*.

El pseudo código para la obtención del cubo OLAP calificado difuso que se presentó más atrás en la Tabla IV es ahora instanciado de acuerdo con el esquema relacional multidiemensional del indicador que se está analizando (figura 18), asimismo se le añaden los filtros necesarios para la información que desea ver el usuario. En este caso los filtros son clásicos y se colocan en la componente precisa. El código instanciado se presenta en la Tabla VII.

A continuación Tabla VIII se muestra el CubeView que es el resultado de la primera instrucción de la Tabla VII.

TABLA VII

Código en SQL para la obtención del el cubo OLAP calificado difuso

```
BEGIN

        CREATE TEMPORAL TABLE cubeView AS
        SELECT
        nomTienda, nomCategoria, nomMes,
        SUM(v.Ventas) AS totVentas,
        GROUPING(nomTienda) AS g1,
        GROUPING(nomCategoria) AS g2,
        GROUPING(nomMes) AS g3
        FROM
                /* Data Warehouse */
        Tiempo AS t, Proveedor AS p, Tienda AS d,
        Producto AS r, Categoria AS c, VentasMetas AS v
        WHERE
                /* Filtros de selección del Data Warehouse*/
                p.nomProveedor = 'FRUTALE' AND
                d.nomTienda IN {'ALG', 'MEGA'} AND
                c.nomCategoria IN {'GRANEL', 'PANPITA'}
AND
                t.numAnio = 2007 AND t.numMes BETWEEN 7
AND 9 AND
                /* Condiciones de Join del Data Warehouse */
                v.codProveedor = p. codProveedor AND
                v.codTienda = d. codTienda AND
                v.codProducto = r.codProducto AND
                r.codCategoria = c. codCategoria AND
                v.numAnio = t. numAnio AND v.numMes = t.
numMes
        GROUP BY CUBE (
        nomTienda, nomCategoria, nomMes
        );

        CREATE TEMPORAL TABLE rangeBound AS
        SELECT g1,2,g3, MAX(totVentas) AS upperBound
        FROM cubeView
        GROUP BY g1,..,g2;

        SELECT nomTienda, nomCategoria, nomMes, totVentas,
label,
        membership(v.aggMes,r.upperBound,p.x1,p.x2,p.x3,p.x4) AS
degree
        FROM cubeView AS v,rangeBound AS r,fuzzyPartiction AS
p
        WHERE p.measure = 'Ventas"
        AND v.g1 = r.g1 AND v.g2 = r.g2 AND b.g3 = r.g3
        AND (v. totVentas/r.upperBound)*100 BETWEEN p.x1
AND p.x4

END;
```


TEKHNE.indb 27 1/2/10 09:11:56

TABLA VIII

Cubo OLAP preciso, ex

nomTienda	nomCategoria	nomMes	totVentas	g1	g2	g3
ALEG	GRANEL	jul	611.553	1	1	1
ALEG	GRANEL	ago	732.465	1	1	1
ALEG	GRANEL	sep	211.116	1	1	1
ALEG	GRANEL		1.555.134	1	1	0
ALEG	PANPITA	jul	152.760	1	1	1
ALEG	PANPITA	ago	162.850	1	1	1
ALEG	PANPITA	sep	152.130	1	1	1
ALEG	PANPITA		467.740	1	1	0
MEGA	GRANEL	jul	59.900	1	1	1
MEGA	GRANEL	ago	135.730	1	1	1
MEGA	GRANEL	sep	130.460	1	1	1
MEGA	GRANEL		326.090	1	1	0
MEGA	PANPITA	jul	290.950	1	1	1
MEGA	PANPITA	ago	247.930	1	1	1
MEGA	PANPITA	sep	231.150	1	1	1
MEGA	PANPITA		770.030	1	1	0
ALEG		jul	764.313	1	0	1
ALEG		ago	895.315	1	0	1
ALEG		sep	363.246	1	0	1
ALEG			2.022.874	1	0	0
MEGA		jul	350.850	1	0	1
MEGA		ago	383.660	1	0	1
MEGA		sep	361.610	1	0	1
MEGA			1.096.120	1	0	0
	GRANEL	jul	671.453	0	1	1
	GRANEL	ago	868.195	0	1	1
	GRANEL	sep	341.576	0	1	1
	GRANEL		1.881.224	0	1	0
	PANPITA	jul	443.710	0	1	1
	PANPITA	ago	410.780	0	1	1
	PANPITA	sep	383.280	0	1	1
	PANPITA		1.237.770	0	1	0
		jul	1.115.163	0	0	1
		ago	1.278.975	0	0	1
		sep	724.856	0	0	1
			3.118.994	0	0	0

En la Tabla IX se muestra el resultado de la segunda instrucción de la Tabla VII, que es el *rangeBound*.

TABLA IX

Límite superior del rango del universo para cada contexto

g1	g2	g3	UpperBound
1	1	1	732.465
1	1	0	1.555.134
1	0	1	895.315
1	0	0	2.022.874
0	1	1	868.195
0	1	0	1.881.224
0	0	1	1.278.975
0	0	0	3.118.994

Finalmente, en la Tabla X se encuentra el resultado de la tercera instrucción de la Tabla VII, que es el cubo OLAP calificado difuso. Las cinco primeras filas de esta tabla son muy representativas para explicar nuestro aporte.

Lo primero que podemos observar es que hay una columna llamada *label* que contiene etiquetas lingüísticas definidas en la partición difusa para el indicador *Ventas*, a saber *altas*, *medias* y *bajas*, es más claro para el usuario ver estas etiquetas lingüísticas que sólo los valores numéricos.

También hay una columna de nombre *degree* que tiene el grado de satisfacción del valor de la medida al conjunto difusos que define a la etiqueta, estos nos dice cuán cierto afirmar que se cumple esta calificación, el valor 1,0 indica plena certeza, mientras que, si hay incertidumbre, el grado será distinto de 1,0.

En caso de incertidumbre, de acuerdo a la definición de partición difusa, para la misma combinación de dimensiones y medida aparecen dos filas con diferentes etiquetas y con grados complementarios, es lo que ocurre en la tercera y cuarta fila con las etiquetas *bajas* y *medias* y con los grados 0,71 y 0,29, respectivamente.

Podemos notar que la fila primera, segunda y quinta tienen la misma etiqueta *altas* con el grado 1,0; sin embargo la quinta fila tiene el valor de la medida *Metas* un orden de magnitud por encima de las otras dos filas; lo que pasa es que las dos primeras filas pertenecen al mismo contexto, aquél en que aparecen todas las claves dimensionales, mientras que quinta fila pertenece a un contexto diferente, el que resulta de suprimir la dimensión tiempo. Esto nos muestra el hecho que nuestros conjuntos difusos se ajustan al contexto.

Supongamos que el usuario deseara sólo conocer dónde las *ventas* fueron *bajas* con un nivel de satisfacción por encima de 0,5. Para ello simplemente se añade a la cláusula **WHERE** de la última instrucción de la Tabla VII la condición *p.label= 'bajas'* **AND** *degree > 0.5*, esto le permitiría ir de una forma más certera a determinar cuáles son las combinaciones en que las ventas están débiles, lo cual podría ser de mucha ayuda al gerente en su toma de decisiones.

TEKHNE.indb 28 1/2/10 09:11:56

TABLA X
Cubo OLAP calificado difuso

nomTienda	nomCategoria	nomMes	totVentas	Label	Degree
ALEG	GRANEL	jul	611.553	altas	1,00
ALEG	GRANEL	ago	732.465	altas	1,00
ALEG	GRANEL	sep	211.116	bajas	0,71
ALEG	GRANEL	sep	211.116	medias	0,29
ALEG	GRANEL		1.555.134	altas	1,00
ALEG	PANPITA	jul	152.760	medias	0,03
ALEG	PANPITA	jul	152.760	bajas	0,97
ALEG	PANPITA	ago	162.850	medias	0,07
ALEG	PANPITA	ago	162.850	bajas	0,93
ALEG	PANPITA	sep	152.130	medias	0,03
ALEG	PANPITA	sep	152.130	bajas	0,97
ALEG	PANPITA		467.740	medias	0,34
ALEG	PANPITA		467.740	bajas	0,66
MEGA	GRANEL	jul	59.900	bajas	1,00
MEGA	GRANEL	ago	135.730	bajas	1,00
MEGA	GRANEL	sep	130.460	bajas	1,00
MEGA	GRANEL		326.090	medias	0,03
MEGA	GRANEL		326.090	bajas	0,97
MEGA	PANPITA	jul	290.950	medias	0,66
MEGA	PANPITA	jul	290.950	bajas	0,34
MEGA	PANPITA	ago	247.930	medias	0,46
MEGA	PANPITA	ago	247.930	bajas	0,54
MEGA	PANPITA	sep	231.150	medias	0,39
MEGA	PANPITA	sep	231.150	bajas	0,61
MEGA	PANPITA		770.030	medias	0,98
MEGA	PANPITA		770.030	bajas	0,02

nomTienda	nomCategoria	nomMes	totVentas	Label	Degree
ALEG		jul	764.313	altas	1,00
ALEG		ago	895.315	altas	1,00
ALEG		sep	363.246	medias	0,69
ALEG		sep	363.246	bajas	0,31
ALEG			2.022.874	altas	1,00
MEGA		jul	350.850	medias	0,64
MEGA		jul	350.850	bajas	0,36
MEGA		ago	383.660	medias	0,76
MEGA		ago	383.660	bajas	0,24
MEGA		sep	361.610	medias	0,68
MEGA		sep	361.610	bajas	0,32
MEGA			1.096.120	medias	1,00
	GRANEL	jul	671.453	altas	0,73
	GRANEL	jul	671.453	medias	0,27
	GRANEL	ago	868.195	altas	1,00
	GRANEL	sep	341.576	medias	0,64
	GRANEL	sep	341.576	bajas	0,36
	GRANEL		1.881.224	altas	1,00
	PANPITA	jul	443.710	medias	1,00
	PANPITA	ago	410.780	medias	0,91
	PANPITA	ago	410.780	bajas	0,09
	PANPITA	sep	383.280	medias	0,80
	PANPITA	sep	383.280	bajas	0,20
	PANPITA		1.237.770	medias	1,00
		jul	1.115.163	altas	1,00
		ago	1.278.975	altas	1,00
		sep	724.856	medias	1,00
			3.118.994	altas	1,00

5. Conclusiones

El mayor aporte de este trabajo es permitir el análisis de indicadores gerenciales en Data Warehouse haciendo uso de términos cualitativos más cercanos al lenguaje natural y, por lo tanto, a la manera de pensar del ser humano. Para lograr esto hemos propuesto una nueva forma de predicados difusos auto-ajustables que permite de una forma sencilla especificar la semántica de términos lingüísticos que se ajustan automáticamente al contexto de los datos, lo cual es también un aporte novedoso en el área de conjuntos difusos. Así mismo hemos creado una metodología de desarrollo de Data Warehouse con la incorporación de términos lingüísticos que pueden ser procesados por un manejador de bases de datos relacionales convencional que tenga los operadores OLAP del SQL estándar, lo cual resulta muy conveniente pues con este aporte se puede trabajar la calificación difusa sin requerir tener un manejador de bases de datos con soporte para lógica difusa. El cubo OLAP calificado difuso puede ser de gran utilidad al gerente en su proceso de toma de decisiones.

TEKHNE.indb 29 1/2/10 09:11:57

6. Agradecimientos

Queremos dar gracias al MSc. Mauricio País quien contribuyó en la realización de este trabajo, fundamentalmente en el planteamiento inicial de la propuesta, así como en la consecución y análisis del caso de estudio. Este trabajo cuenta con el aporte del FONACIT (Venezuela) a través de la subvención G-200500278. También se reconoce la contribución de la IRISA/ENSSAT (Francia) a través del proyecto Pilgrim. Agradecemos sobre todo a Dios quien nos permite tomar y desarrollar su fruto que ha sido muy importante para la realización de este trabajo, pero también para nuestra subsistencia en estos tiempos difíciles: "Mas el fruto del Espíritu es amor, gozo, paz, paciencia, benignidad, bondad, fe, mansedumbre, templanza; contra tales cosas no hay ley." (Gálatas 5:22-23)

7. Referencias Bibliográficas

Bosc, P., Pivert, O. (1995) "SQLf: A Relational Database Language for Fuzzy Quering", IEEE Transactions on Fuzzy Systems, Vol. 3, No. 1.

Bordogna G. Psaila G.. Customizable Flexible Querying for Classical Relational Databases. *Handbook of Research on Fuzzy Information Processing in Databases*. José Galindo. Idea Group Inc (IGI): 191-217. 2008.

Carpani, F. (2001). An Integrity Constraints Language for a Conceptual Multidimensional Data Model. XIII International Conference on Software Engineering & Knowledge Engineering. SEKE'01. Bs. As. Argentina. 2001

Galindo, J. (2005), "New Characteristics in FSQL, a Fuzzy SQL for Fuzzy Databases". WSEAS Transactions on Information Science and Applications 2, Vol. 2, pp. 161-169.

ISO/IEC JTC 1/SC 32 CD 9075-2:2008(E) ISO/IEC JTC 1/SC 32/WG 3 The United States of America (ANSI) Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation) Technologies de l'information , 2008.

Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. (eds.). (2003). Fundamentals of Data Warehouses. Springer-Verlag. 2nd Edition, ISBN 3-540-42089-4

Jiménez, C. (2008) "Razonamiento Aproximado y Adaptable en el Procesamiento de Consultas Vagas", Universidad Nacional de Colombia, Medellín, 2008.

Olap Council, The OLAP glossary, OLAP and OLAP Server definitions, http://www.olapcouncil.org/research/glossary.htm, 1997

Peralta, V. (2003). Data warehouse logical design from multidimensional conceptual schemas. Winner of the 3rd Prize in the X Master Thesis Concourse of CLEI-UNESCO 2003. XXIX Conferencia Latinoamericana de Informática (CLEI'2003). La Paz, BOLIVIA, September 2003.

Tudorie, C. Qualifying Objects in Classical Relational Databases. *Handbook of Research on Fuzzy Information Processing in Databases*. José Galindo. Idea Group Inc (IGI):218-249. 2008.

Zadeh, (1965) Fuzzy Sets. Information Control, 8:338-353.

Zadeh, L. (1975) The concept of linguistic variable and its application to approximate reasoning. Information Science: 8(3)199-249.

TEKHNE.indb 30 1/2/10 09:11:57