

Implementación Indicadores de Gestión Cuantitativos y Cualitativos en Almacenes de Datos

Angélica Urrutia (1), Marcela Varas (2)

1Universidad Católica del Maule, Chile,
aurrutia@ucm.cl

2Universidad de Concepción, Chile.
mvaras@udec.cl

Resumen. Con el fin de flexibilizar consultas de gestión de información estratégica de la organización, es que en este trabajo se presenta un conjunto de pasos para implementar cubos y consultas para datos precisos e imprecisos. Las consultas se desarrollan utilizando las instrucciones *Cube* y *Ventana* del estándar SQL 2003 y una extensión de indicadores de gestión con conjuntos difusos.

Introducción

Los sistemas de inteligencia de negocios tienen como elemento clave de su arquitectura a los almacenes de datos o datawarehouses (DW). Un DW es una base de datos histórica, orientada a temas y satisfacer indicadores de gestión, diseñada para ser eficiente en las consultas, no en las actualizaciones (como es el caso de las bases de datos relacionales que soportan los sistemas de información transaccionales). Un DW considera un proceso analítico en línea (OLAP), el cual se implementa sobre la base de alguna de las siguientes alternativas: MOLAP (OLAP sobre una base de datos multidimensional), ROLAP (OLAP sobre una base de datos relacional) o HOLAP (combinando ambos enfoques) [4, 5].

Los DW se diseñan, entre otros, con el objetivo de ser eficientes en los requerimientos de análisis para niveles estratégicos en las organizaciones. Una forma reduccionista de verlo, pero útil para nuestro artículo, es considerar que los DW se diseñan para poder calcular y analizar un conjunto de indicadores de gestión.

Con este enfoque, los indicadores de gestión dirigirán el diseño, y se convertirán en las “medidas”, y las variables a considerar en su análisis, en las “dimensiones”, componentes básicas del modelo Multidimensional.

Avanzando en el proceso de diseño, a nivel lógico, cuando se implanta una base de datos multidimensional sobre ROLAP, se utilizan esquemas estrella y copos de nieve [11]. Estos esquemas se componen de dos tipos de tablas: de hechos o *Fact* que contiene los indicadores de gestión y de dimensión que contienen el detalle de los valores que están asociados a la tabla de hechos. La estructura básica de un esquema estrella es conformada por una tabla central y un conjunto de tablas que la atienden radialmente. El centro de la estrella consiste de una o más tablas de hechos y las puntas de la estrella son las tablas dimensionales. El esquema copo de nieve es obtenido de la normalización de las dimensiones que componen un modelo estrella y tiene como objetivo reducir la información redundante de los modelos dimensionales [3].

Los indicadores de gestión actualmente manejados son medidas numéricas (cuantitativas), que son sumariadas o desumariadas por alguna dimensión. Considerando que una forma de facilitar el análisis es adicionar información cualitativa a estos indicadores, es que se les añaden etiquetas lingüísticas que les son asignadas a cada indicador según su valor, de acuerdo a una función de pertenencia que responde a los principios de la lógica difusa, de modo de acercarlo aún más a los usuarios finales de los sistemas.

Se presenta en este artículo un conjunto de pasos que permiten implementar en un DW ROLAP indicadores de gestión con información cuantitativa y realizar consultas con los operadores CUBE y VENTANA del estándar SQL 2003 [8, 9, 10] y una extensión de indicadores de gestión cualitativos con lógica difusa finalmente la conclusiones y referencias.

Antecedentes

En el Estándar SQL 2003 se incorporan la instrucción *CUBE*, definido como:

```
v1, v2,..., vn, f()
v1, v2,..., 'Todos', f()
....
v1, 'Todos',..., 'Todos', f()
'Todos', 'Todos',..., 'Todos', f()....., donde f() es una función de
agregación.
```

El operador *CUBE* [8, 9, 10] es la generalización en N-dimensiones (N-D) de funciones de agregación simples. El cubo de datos de cero dimensiones (0-D) es un punto, el de 1-D es una línea con un punto, el de 2-D es una tabla cruzada, un plano, dos líneas y un punto y el de 3-D es un cubo con tres intersecciones de tablas cruzadas en 2-D. En la siguiente sentencia SQL se ve un ejemplo básico de sintaxis del operador *CUBE*.

```
SELECT   día, nación, MAX(Temp)
FROM      tiempo
GROUP BY   CUBE ( Día (Time) AS dia, Pais(Latitud,
Longitud) AS Nacion);
```

La sentencia expuesta, agrupa los atributos del *SELECT* como en un *GROUP BY* estándar. Estos atributos serán las dimensiones del cubo y están relacionados con un indicador de toma de decisiones. Luego, sobre los grupos resultantes se aplican las funciones de agregación a las medidas. Estas columnas serán llamadas columnas de agregación.

Otro operador incluido en SQL utilizado en la expresión de consultas que soportan la toma de decisiones es la Ventana: *Windows* [8, 9, 10]. El uso de este operador produce una tabla “aventanada” (*windowed table*) definida como una tabla que posee una o más ventanas (*windows*). Una ventana es una estructura de datos transitorios asociados con una expresión de la tabla, cada ventana permite la especificación de particiones y marcos.

Una ventana define un ordenamiento de los registros dentro de cada partición definida por la Ventana. Opcionalmente, una ventana puede definir un marco (*frame*) para cada registro. El marco se define de forma relativa al registro. Las opciones para especificar serían un rango o algunos registros.

La selección de un rango (RANGE), para indicar la definición del marco por medio de una distancia hacia atrás y adelante del registro actual, definida por un valor. La selección de un número de registros (ROWS) ubicados en una posición hacia atrás o adelante menor o igual a un valor dado, con respecto al registro actual. En ambos casos los registros incluidos dependen del orden definido para la ventana.

Un ejemplo que clarifica el uso de estos marcos y ventanas se presenta a continuación. Esta instrucción permitiría mostrar para cada vendedor, promedios de ventas de 5 ventas consecutivas.

```
SELECT avg(venta) over ( partition by vendedor
                        order by time
                        rows between 2 preceding and 2 following)
FROM Ventas;
```

La teoría de conjuntos difusos (también llamada borrosos) parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia al conjunto, definida ésta como un número real entre 0 y 1 propuesta por Zadeh en 1965 [13]. Así, se introduce el concepto de conjunto o subconjunto difuso asociado a un determinado valor lingüístico, definido por una palabra, adjetivo o etiqueta lingüística A . Para cada conjunto o subconjunto difuso se define una *función de pertenencia o inclusión* $\mu_A(u)$, que indica el grado en que la variable u está incluida en el concepto representado por la etiqueta. Un conjunto difuso A sobre un universo de discurso U (dominio ordenado) es un conjunto de pares dado por:

$$A = \{\mu_A(u) / u : u \in U, \mu_A(u) \in [0,1]\}$$

Donde, μ es la llamada función de pertenencia y $\mu_A(u)$ es el *grado de pertenencia* del elemento u al conjunto difuso A . Este grado oscila entre los extremos 0 y 1, definido como:

- $\mu_A(u) = 0$, indica que u no pertenece en absoluto al conjunto difuso A .
- $\mu_A(u) = 1$, indica que u pertenece totalmente al conjunto difuso A .

Por ejemplo si consideramos el valor lingüístico *estatura_de_una_persona* podría definirse tres subconjuntos difusos, cada uno identificado por una etiqueta {bajo, medio, alto}, y con una función de pertenencia $\mu_{bajo}(u)$, $\mu_{medio}(u)$, $\mu_{alto}(u)$, respectivamente.

Como un estudio de estas instrucciones, en los siguientes apartados se presenta el resultado de este trabajo el cual esta distribuido en: Caso Práctico Cubos y Ventanas, Consultas con extensión del Cubo con lógica difusa (datos imprecisos) para finalizar con las conclusiones y referencias bibliográficas.

Caso Práctico Cubos y Ventanas

Para mostrar nuestro trabajo, a continuación se presenta un caso es el de un sistema de ventas de una cadena de tiendas de productos con presencia en EE.UU, México y Chile (caso de muestra del SQL Server 2000). Los pasos que utilizamos son:

PASO 1: Definición del Problema y Análisis de Requisitos.

El departamento de mercadotecnia, de la cadena de tiendas, desea analizar por productos y clientes todas las ventas realizadas durante el año específico, mediante la consulta de indicadores de gestión que soporten la toma de decisiones o gestión de información.

Los indicadores que deben satisfacer al usuario en la gestión de información son los siguientes:

1. Ventas por producto: esto quiere decir que se necesita reflejar el total de ventas por productos en cualquier mes de un año determinado.

2. Ventas por cliente: se reflejan todos los totales de ventas por clientes que se hacen por día, por un mes específico.
3. Ventas por almacén: son los totales de las ventas que pueden realizar todos los almacenes existentes por un período.
4. Ventas por país: son todos los totales de las ventas por país que se hallan realizado en todos los almacenes en un período.

Para obtener los indicadores de gestión mencionados anteriormente, se requiere de una base de datos fuente, por tanto el esquema relacional de Fig. 1, refleja la base de datos actual de la empresa, la cual es la fuente de información, inicial de nuestro proyecto. En este esquema, la tabla *almacén* asocia las tablas *empleado*, *bodega* y *país_almacén* entrega información detallada de los almacenes, así como también información de las ventas que se pueden realizar en los almacenes con su respectiva bodega y empleados asociados a una venta. Estos datos serán fundamentales como variables para indicadores y análisis de gestión en un momento determinado.

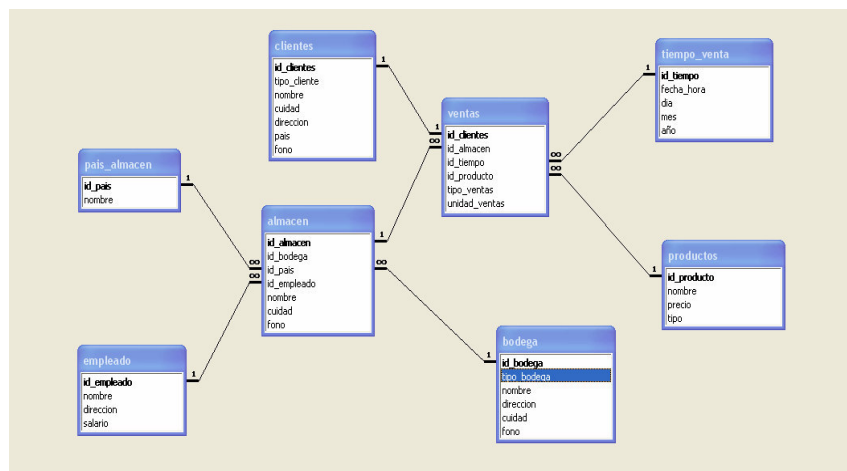


Fig. 1: Modelo relacional del sistema Operacional (BD fuente).

Se debe considerar que en la construcción de un cubo multidimensional o *hipercubo*, se extraen los datos más importantes que satisfacen los indicadores que el usuario especificó inicialmente. A continuación sólo se mostrará el trabajo realizado para un indicador, considerando que el resto se desarrollan de manera análoga.

PASO 2: Creación y Carga del Cubo Ventas.

El *Cubo Ventas* se construye para representar el siguiente Indicador: “*Total de ventas por tipo de productos para un determinado almacén, agrupadas por tipos de clientes en un determinado tiempo*”.

Identificación de tablas de hechos y dimensiones

Comprender el proceso de negocio es importante para el diseño implementación del cubo multidimensional, principalmente para que éste último satisfaga apropiadamente los indicadores. Se creó la tabla de hechos *Fact_Ventas* que representa el centro del cubo. Los atributos de la tabla de hechos, fueron extraídos de la base de datos operacional (Fig. 1) y su propósito es satisfacer el indicador mostrado en un esquema estrella para las *Ventas* (véase Fig. 2), también denominado *el Cubo Venta*. La medida en esta tabla de hechos es el total del valor de las ventas en un período determinado.

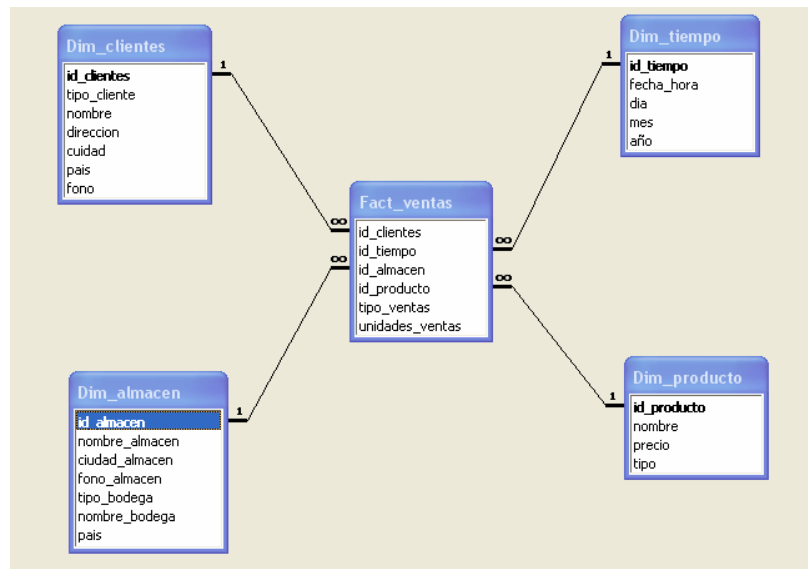


Fig. 2: Esquema Estrella para el *Cubo Venta*.

En la Fig. 2, las dimensiones son: clientes, productos, tiempo y almacén. Éstas son las aristas del cubo multidimensional, también denominadas variables del cubo, en el esquema estrella fueron denominadas:

Dim_clientes, Dim_almacen, Dim_tiempo y Dim_productos. La tabla *Dim_tiempo* es una variable del cubo que engloba todas las variables, en este caso estamos hablando de un *hipercubo ventas*.

Una de las actividades relevantes es la extracción de datos desde la base de datos operacional hacia el DW (cubo ventas), con frecuencia trimestral o semestral, para el posterior uso de funciones analíticas que van a responder las inquietudes frente al negocio de los usuarios para quienes se construye el sistema.

PASO 3: Consultas al Cubo Ventas Utilizando la Instrucción CUBE.

En esta parte se realizan operaciones al cubo ventas (Fig.2), utilizando la instrucción *CUBE* del estándar SQL 2003 [1, 2] para generar diversos reportes.

Resultado de ventas por productos.

Como se dijo anteriormente, las soluciones que presta el cubo desarrollado sirven para dar una mejor visión de las variables productos y clientes, en un tiempo determinado. La empresa requiere consultar todas las ventas que son de tipo entrante y saliente de cada producto en un tiempo determinado. Véase Fig. 3, a su izquierda la instrucción en SQL usando la instrucción *CUBE*, a la derecha la estructura de la consulta usando ANY Where.

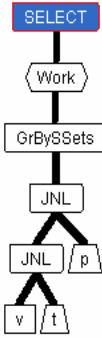
CONSULTA EN SQL	ESTRUCTURA
<pre>SELECT case when grouping (v.tipo_ventas)=1 then 'TODAS' when grouping (v.tipo_ventas)=0 then v.tipo_ventas end as "tipo de ventas", case when grouping(p.tipo)=1 then 'TODAS' when grouping (p.tipo)=0 then p.tipo end as "tipo de productos", case when grouping (t.mes)=1 then 'TODAS' when grouping (t.mes)=0 then t.mes end as "mes de venta", sum(p.precio) as "Precio" FROM fact_ventas v , dim_tiempo t, dim_productos p WHERE v.id_tiempo=t.id_tiempo AND v.id_producto=p.id_producto GROUP BY CUBE (v.tipo_ventas,p.tipo,t.mes);</pre>	 <pre> graph TD SELECT[SELECT] --> Work{Work} Work --> GrBySSets[GrBySSets] GrBySSets --> JNL1[JNL] JNL1 --> JNL_p[JNL / p] JNL_p --> v[v] JNL_p --> t[t] </pre>

Fig. 3: Sentencia *CUBE* de SQL para tipo de ventas por productos.

La visualización de la estructura de la consulta del software SQL ANY Where 10 entrega todo lo necesario para entender en detalle todo lo relacionado con la consulta en cuestión para ello, utiliza la siguiente nomenclatura:

Work: Declaración de variables a utilizar en la consulta. Para el caso de estudio se conformaría de las variables v.tipo_ventas, p.tipo, t.mes, p.precio.

GrBySSets: realiza la agrupación de todas las variables de la instrucción *CUBE*, es decir las agrupaciones entre los conjuntos:

1. v.tipo_ventas, p.tipo, t.mes
2. v.tipo_ventas, p.tipo
3. v.tipo_ventas, t.mes
4. p.tipo, t.mes
5. v.tipo_ventas
6. p.tipo
7. t.mes
8. () significa que todas las ventas o todas los tipos.

JNL: Unión de 2 o mas tablas.

La ejecución de la consulta de la Fig. 3 genera un reporte que entrega todas las ventas ya sean de tipo entrante o saliente de todos los meses del una año específico, como también subtotales de meses por ventas.

PASO 4: Implementación de Funciones de VENTANA utilizando *Cubo Ventas*.

En esta sección se utilizan la función de ventana, que reflejen la manipulación del *Cubo Ventas*.

Resultado funciones RANK de VENTANAS aplicadas a las ventas por país.

Las funciones RANK calculan la posición ordinal de un registro en la partición generada a través de la ventana. Las funciones de RANK (), DENSE_RANK () y ROW_NUMBER () son bastantes similares. La función RANK retorna la posición relativa de un valor en un grupo ordenado, los valores iguales compartirán la misma posición. DENSE_RANK elimina la discontinuidad en la numeración de RANK causada por el conjunto de registros que comparten los mismos valores, es decir los que se encuentran en empate. ROW_NUMBER enumera de forma continua a los registros y asocia de forma no determinística una posición distinta a cada ocurrencia de un valor, inclusive si hay más de ocurrencia.

En el caso de estudio se desea realizar un consulta utilizando funciones de ranking, RANK () DENSE_RANK () Y ROW_NUMBER () para entregar las ventas por almacén clasificadas por país. Esta consulta se encuentra expresada en la Fig. 4, para la cual se observarán diferencias en las numeraciones de los registros cuando existan distintos registros que compartan el mismo monto de venta. El resto de las instrucciones ventanas se utilizan de la misma forma [7].

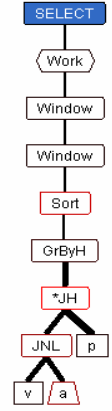
CONSULTA EN SQL	ESTRUCTURA
<pre> SELECT a.id_almacen,a.pais, sum(p.precio) as ventas, RANK() OVER(PARTITION BY a.pais order by sum(p.precio) desc) as ranking, DENSE_RANK() OVER(PARTITION BY a.pais order by sum(p.precio) desc) as ranking_des, ROW_NUMBER() OVER(PARTITION BY a.pais order by sum(p.precio) desc) as ranking_row FROM fact_ventas v, dim_almacen a, dim_productos p WHERE v.id_almacen=a.id_almacen AND v.id_producto=p.id_producto GROUP BY a.id_almacen,a.pais; </pre>	 <pre> graph TD SELECT[SELECT] --> Work{Work} Work --> Window1[Window] Window1 --> Window2[Window] Window2 --> Sort[Sort] Sort --> GrByH[GrByH] GrByH --> JH[*JH] JH --> JNL[JNL] JH --> p[p] JNL --> v[v] JNL --> a[a] </pre>

Fig. 4: Utilización de función RANK (), DENSE_RANK (), ROW_NUMBER en el Cubo Ventas.

PASO 5: Comparación en la Facilidad de Entendimientos de Resultados.

En esta parte, la comparación se puede elegir la óptima consulta con respecto a los requerimientos del usuario, pero la facilidad de entendimiento depende exclusivamente del usuario con respecto al contexto que esté. A modo de ejemplo la Fig. 5 a) fue construida con instrucciones *CUBE* y el reporte mostrado en la Fig. 5 b) fue construido utilizando la instrucción *VENTANA*.

a.

Tipo De Ventas	Tipo De Productos	Mes De Venta	Precio
TODAS	construccion	octubre	7000
Entrante	construccion	octubre	7000
Entrante	TODAS	octubre	7000
TODAS	TODAS	octubre	7000
Saliente	construccion	julio	20000
Saliente	TODAS	julio	20000
Entrante	TODAS	julio	24900
Entrante	construccion	julio	24900
Saliente	estructura	abril	30000
Entrante	tecnologia	marzo	30000
Saliente	TODAS	abril	30000
Entrante	TODAS	marzo	30000
TODAS	construccion	julio	44900
TODAS	TODAS	julio	44900
Saliente	TODAS	diciembre	50000
Saliente	tecnologia	diciembre	50000
Saliente	tecnologia	marzo	55000

b.

TIPO_VENTAS	TIPO	MES	Precio
Entrante	construccion	enero	80000
Entrante	construccion	enero	95300
Entrante	construccion	julio	24900
Entrante	construccion	octubre	5000
Entrante	construccion	octubre	7000
Entrante	construccion	septiembre	70000
Entrante	construccion	septiembre	93000
Entrante	construccion	septiembre	101000
Entrante	estructura	abril	25000
Entrante	estructura	abril	43000
Entrante	estructura	abril	58000
Entrante	tecnologia	diciembre	300000
Entrante	tecnologia	diciembre	380000
Entrante	tecnologia	enero	300000
Entrante	tecnologia	marzo	30000

Fig. 5: a) Resultado Utilizando Cube, b) Resultado utilizando Ventanas.

Se realizan otras consultas utilizando las instrucciones CUBE y VENTANA las cuales se pueden encontrar en [7]. En algunos casos, no se pudo hallar una consulta usando el operador CUBE, que fuera equivalente a la que incluía el uso de ventanas.

Consultas con Extensión del Cubo con Lógica Difusa (datos imprecisos, Fuzzy Cube)

El caso de estudio tomado de una situación real de una empresa, considera las ventas generadas en una cadena de tiendas de una *Empresa de Retail*. Para una mejor y más acabada comprensión analizaremos el origen de los datos y los requerimientos generales de la empresa.

PASO 1: Descripción del Caso.

En la descripción del origen de los datos, la fuente de datos pertenece al registro de las ventas generadas en una cadena de puntos de venta (POS) de una empresa del sector Retail, sector en el cual la empresa tiene una importante pero no masiva participación de mercado. Actualmente, se están realizando grandes esfuerzos en potenciar las fortalezas de la empresa y optimizar al máximo el uso de la plataforma tecnológica con la que se cuenta, con lo cual se espera enfrentar con éxito los importantes objetivos y desafíos definidos para el presente año.

Los indicadores de gestión están relacionados principalmente, 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 el análisis, logro de los objetivos y metas de la compañía. Actualmente, la empresa posee dos plantas productivas, un centro de distribución y una cadena de 50 tiendas, las cuales se agrupan en sectores en la región metropolitana (Santiago de Chile), lugares en donde se comercializan principalmente sus productos y otros productos de proveedores a terceros, razón por lo cual se definió este alcance geográfico en la muestra de los datos a trabajar. La comercialización de sus productos y los de terceros, se realiza en su cadena de tiendas, más otro canal de distribución, entre los cuales destacan principalmente supermercados, restaurantes, clientes minoristas y otros.

Bajo el escenario anteriormente señalado, nuestros esfuerzos estarán orientados al apoyo del cumplimiento de las metas de la gerencia de *Retail*, la cual depende de la gerencia comercial y cuyo principal objetivo es el de administrar y velar por el cumplimiento de las metas de ventas de todas las cadenas de tiendas (POS), de nuestra compañía. Esta gerencia es la responsable de la definición, control y cumplimiento de los indicadores de gestión que garanticen los objetivos del área Retail están siendo cumplidos.

A lo anterior el aporte de nuestro proyecto tiene relación, con que a la información numérica actualmente en uso para realizar gestión, le falta aportar un análisis cualitativo de la información, que permita una mejor representación. Por ejemplo, si se vendieron 1000 unidades de un producto, quisiéramos conocer si las ventas corresponden a una venta “buena”, “regular” o “mala” de un período y región.

Requerimientos: Cada año la gerencia de *Retail* debe definir en conjunto con la gerencia comercial y gerencia general los objetivos, desafíos, proyectos y planes de acción en lo referente a la gestión y resultados que deberán tener toda la cadena de tiendas de la compañía. Estos compromisos y definiciones de tareas deberán estar centrados principalmente en una proyección, compromiso y control de las ventas, generadas en todas las tiendas. Estas metas deberán ser definidas en forma mensual y anual, a partir de las cuales con un buen análisis y seguimiento realizado en forma periódica e histórica de las ventas generadas, se podrá determinar algunos indicadores como:

1. Evolución de las ventas: Por tienda, Por sector, Por familia o subfamilia de productos.
2. Análisis de Mercado y Distribución geográfica: Medir las ventas para los diferentes sectores de la ciudad.

PASO 2: Definición de Indicadores de Gestión.

Corresponden a instrumentos de medición del comportamiento y/o tendencias de las variables asociadas a las metas para una determinada

área estratégica de la empresa. Estos indicadores fueron rescatados a partir de las necesidades definidas por los ejecutivos y directores de la empresa. A continuación, se definen un conjunto de indicadores de gestión, para posteriormente seleccionar uno sobre el cual desarrollaremos nuestro trabajo de consultas precisas y consultas imprecisas.

El análisis de medidas definidas en indicadores de gestión se desarrollan en una de las tres categorías de medidas (aditivas, semi aditivas y no aditiva). En este caso todas las medidas generadas por los indicadores son aditivas. Los indicadores propuestos son los siguientes:

1. Ventas y participación porcentual.
2. Análisis de Rentabilidad tiendas.
3. Participación de las formas de pago por tiendas.
4. Venta y Participación proveedores en tiendas.
5. Análisis ventas/metras de tiendas.

Para el indicador 5, que es el que se mostrará en este trabajo se considera la medición de ventas, metas y porcentaje de cumplimiento de ventas. Esta información será analizada por los proveedores, categoría de productos, fecha y ubicación. Los criterios para analizar la información; Ubicación tiendas, Fechas de ventas (mes, año), Familia de productos y Familia de proveedores. Los indicadores a considerar son el total de; Venta acumulada, Metras y Cumplimiento de ventas.

Para el análisis de las ventas, se consideraron las Venta acumulada: Mide las Ventas acumuladas de una tienda, Ventas acumuladas de un sector, las Ventas acumuladas de un proveedor (tienda o sector), las Ventas acumuladas categoría de productos (tienda o sector) y las Ventas acumuladas (mes, año).

Para el análisis de las Metras: Mide las metras de una tienda, Metras de un sector, Metras de un proveedor (tienda o sector), Metras categoría de productos (tienda o sector) y Metras por mes/año.

Cumplimiento de Ventas: Mide las *Ventas* y *Metras* acumuladas de una tienda, de un sector, de un proveedor (tienda o sector), de categoría de productos (tienda o sector) y Ventas acumuladas por mes y año.

PASO 3: Diagrama actual puntos de venta tiendas.

La aplicación actual para puntos de venta, se encuentra en un sistema POS de puntos de venta instalado en cada cadena de tiendas, desarrollado en JAVA, asegurando portabilidad. Además posee un elevado nivel de parametrización, flexibilidad, y escalabilidad.

Esta aplicación, es OLPT con un servidor central, que almacena el repositorio principal desde donde se rescata la data para realizar gestión de *ventas-tiendas*.

La Fig. 6, muestra el Modelo de Datos de solución puntos de venta (POS). Corresponde al diseño de datos de las tablas y sus relaciones, las cuales contienen el registro de todas las ventas, formas de pago, artículos, categorías, etc, generadas en toda la cadena de tiendas.

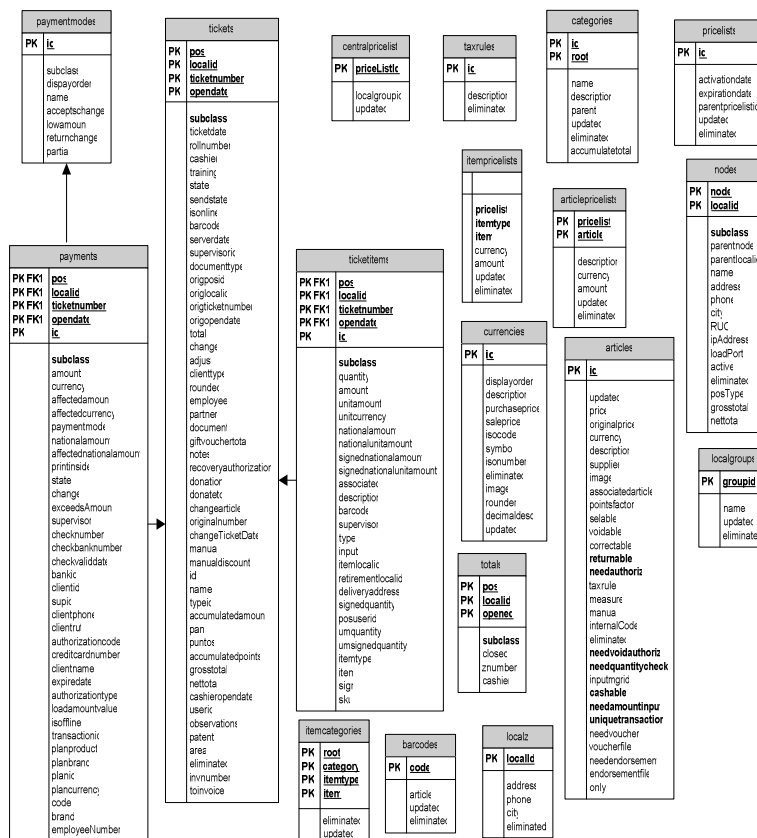


Fig. 6: Modelo de datos de la base de datos fuente de la solución puntos de venta.

PASO 4: Diseño Lógico.

La construcción de un indicador comienza a partir de la base de datos fuente, con se respectivo mapeo que genera un cubo que es sus medidas tiene los indicadores requeridos por la empresa a partir de las tablas de la base de datos fuente. Se ha tomado la decisión de continuar nuestro trabajo de investigación con la implementación del indicador 5 definido: “Análisis ventas/metras tiendas”. Esto se inicia con la creación del repositorio DW y herramientas ETL considerando las siguientes etapas:

1. Extracción de datos de la fuente (sistema de venta POS y otras fuentes externas).
2. Filtrado de datos, eliminando datos innecesarios.
3. Modificar formatos y valores cuando se requiere (filtrado).
4. Uniformar las tablas de códigos de los sistemas operacionales y simplificar esquemas de codificación.
5. Agregaciones de nuevos campos, a través de cálculos y consolidaciones de indicadores a partir de datos de la fuente.
6. Transformación de datos (Calcular sumalizaciones, variables derivadas, etc).

Estas actividades solo son señaladas, ya que si profundizamos en cada una de ellas, podríamos extendernos en demasía y nos desviaremos un poco del objetivo principal del proyecto.

El resultado de este proceso es el diseño lógico del repositorio (DW), siendo este diseño obtenido por el proceso ETL, y forman la base de datos que implementará el cubo de gestión OLAP para el indicador “Análisis ventas/metras tiendas”. El esquema resultante del indicador de gestión seleccionado se representa a través de un esquema estrella y dos esquemas copo de nieve (véase Fig. 8 parte superior).

PASO 5 : Implementación cubo OLAP.

Una vez seleccionado el indicador de gestión *Análisis ventas/metras tiendas*, fue construido el modelo CMDM [11] y mapeo fragmentado desde la base de datos fuente considerando la especificación de

requerimientos, además el diseño lógico del repositorio, se trabajo con la herramienta de gestión OLAP de Análisis Manager de Microsoft SQL Server 2000, para desarrollar e implementar las consultas precisas. El resultado de este proceso se encuentra en [12].

PASO 6: Análisis de Indicadores difusos (Etiquetas).

Una vez implementada la herramienta para visualizar las consultas OLAP precisas, es el momento de implementar los componentes necesarios para implementar las consultas OLAP imprecisas (difusas) usando; conjuntos difusos, lógica difusa y definición de etiquetas lingüísticas con los siguientes valores:

1. Venta Mala {1000, 2000, 3000, 99999}.
2. Venta Buena {5000, 6000, 7000, 9999}.
3. Venta Excelente {9000, 10000, 12000, 099999}.

Cada una de estas etiquetas fue definidas en la especificación de requerimientos del Paso 1 y 2. La siguiente actividad es describir las componentes necesarias para la implementación de este paso y la definición de ciertas estructuras, procedimientos, funciones y carga de datos. Se utiliza el lenguaje de consultas flexibles SQLf [14] para extender el diseño estrella. El cómo se implementa, lo muestra la Fig. 7 extendiendo el diseño lógico de un cubo a tres tablas: tabla multidimensional, tabla de matriz de clasificación y de etiquetas lingüísticas.

En la creación de la Tabla Multidimensional se toma como tabla de entrada del *Cubo Ventas Retail* (Fig. 8 parte superior), y su particularidad será la utilización del operador *CUBE*, entregando tuplas con indicadores de totalización para la intersección de las dimensión definida. Las dimensiones serán valores, pertenecientes al dominio asociado, ejemplo: Dimensión proveedores, Dominio valores: Frutales, Ensenada y Terceros. A continuación se muestra el llenado de las tablas 1 y 2 para este caso.

Tabla 1: Matriz de Clasificación, sus atributos dimensiones deben tener la misma estructura de las dimensiones que en tabla Multidimensional Difusa.

Código	Clasificación		Proveedor	Tienda	Mes
C1	YES	YES	YES		
C2	YES	YES	ALL		
C3	YES	ALL	ALL		

Tabla 2: Contiene las Etiquetas Lingüísticas que están asociadas a códigos de clasificación.

Código	Clasificación	Etiqueta	Valor1	Valor2	Valor3	Valor4
C1		MALA	1000	2000	3000	99999
C1		BUENA	5000	6000	7000	99999
C1		EXCELENTE	9000	10000	12000	99999

Creación de código para implementación. Aquí se definen los procedimientos, vistas y funciones necesarias para implementar las consultas multidimensionales imprecisas. Algunos de estos procedimientos serán complementarios a SQLf y otros harán uso de los componentes y/o definiciones ya desarrolladas, para conseguir el resultado de las consultas multidimensionales difusas.

PASO 7: Implementación de Indicadores Extendidos.

En este paso se tendrá como objetivo el implementar las componentes definidas en el paso anterior, el cual nos permitirá visualizar con datos el resultado de las consultas OLAP Difusas. El Diseño de la solución a implementar se muestra en la Fig. 8, que representa una extensión de cubo para consultar en forma imprecisa los datos sobre una base de datos multidimensional relacional precisa, y obtener como resultado las medidas convertidas ahora en etiquetas lingüísticas, para lo cual es necesario extender, o más bien generar nuevas piezas y procedimientos, los que se pueden complementar con las componentes actuales del lenguaje SQLf. La Fig. 7 muestra el ciclo de implementación del modelo propuesto para consultas en cubos imprecisos.

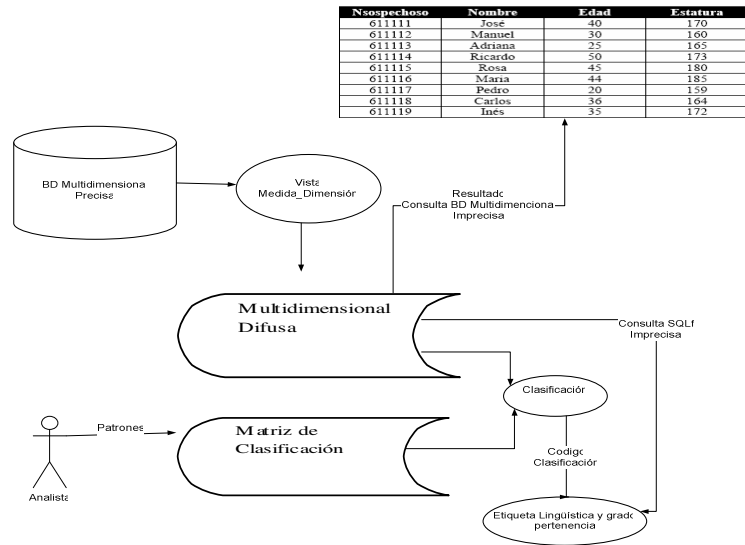


Fig. 7. Etapas del OLAP Difuso.

Extensión del Diseño de Datos: En este modelo lógico de estrella en copo de nieve, podemos observar la incorporación de las Tablas *Multidimensional*, *Matriz Clasificación* y *Etiquetas*, mostrada en la parte inferior de la Fig. 8. Dichas tablas constituyen la extensión difusa al diseño Estrella y Copo de nieve de la Figura 8 parte superior.

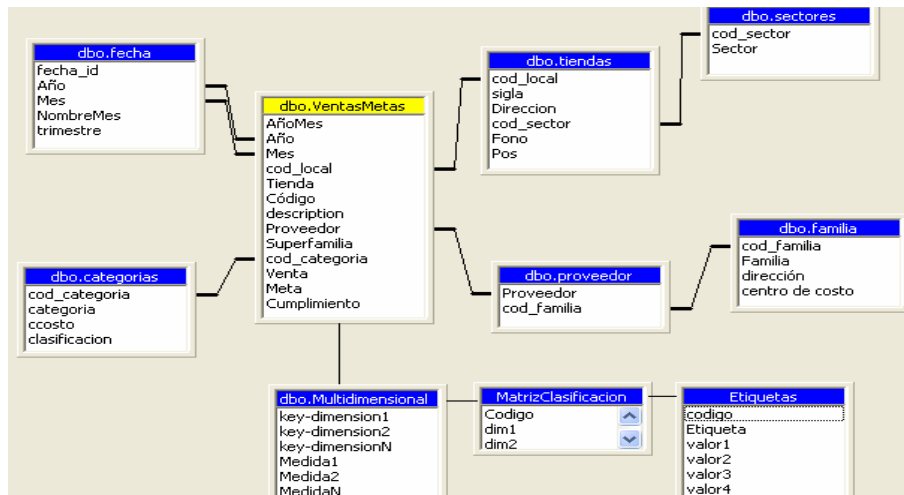


Fig. 8. Extensión del diseño estrella a diseño difuso.

El set de datos de prueba, está constituido por los datos del Cubo obtenido de la base de datos fuente de *ventas retail*, los cuales son rescatados y formateados como totalizadores de dimensiones, con la utilización de vista *ventas_tiendas*, el cual utiliza el operador *CUBE*. A continuación mostramos algunos ejemplos.

A modo de ejemplo, se puede apreciar de mejor manera el resultado de la implementación, aplicando algunos criterios de selección a las dimensiones existentes en una consulta precisa y una consulta imprecisa.

Requerimiento de consulta:

Proveedor : Frutale
 Tienda : ALEG
 Año : 2007
 Período Acumulado: Agosto/07 – Septiembre/07 – Octubre/07

Consulta Precisa:

*SELECT * FROM ventas_tiendas WHERE (tienda = 'ALEG') and (proveedor = Frutale) and (Ano = 2007)*

Resultado obtenido:

Proveedor	Tienda	Categoría	Mes	Venta
FRUTALE	ALEG	GRANEL	10	3611553
FRUTALE	ALEG	GRANEL	8	3732465
FRUTALE	ALEG	GRANEL	9	3211116
FRUTALE	ALEG	GRANEL	ALL	10555134
FRUTALE	ALEG	HOT DOG	10	52760

””

La aplicación de función Etiqueta y de acuerdo con los parámetros de entrada, clasificación y monto de la medida, será posible establecer la etiqueta lingüística correspondiente para cada patrón de totalización de dimensiones (medida). Una consulta y respectivo resultado se muestra a continuación:

Consulta Imprecisa:

```
SELECT proveedor,tienda,categoria,mes,venta,  
dbo.etiqueta(dbo.clasificacion(proveedor,tienda,categoria,mes),VE  
NTA), dbo.clasificacion(proveedor,tienda,categoria,mes)  
CLASIFICACION  
FROM VENTAS_TIENDAS
```

Resultado obtenido:

Proveedor	Tienda	Categoría	Mes	Venta	Etiqueta	Clasificación
FRUTALE	ALEG	GRANEL	10	3611553	EXCELENTE	C1
FRUTALE	ALEG	GRANEL	8	3732465	EXCELENTE	C1
FRUTALE	ALEG	GRANEL	9	3211116	EXCELENTE	C1
FRUTALE	ALEG	GRANEL	ALL	10555134	EXCELENTE	C2
FRUTALE	ALEG	HOT DOG	10	52760	MALA	C1

Obsérvese, que el resultado obtenido por la función etiqueta aplicada sobre el cubo Venta, representa la salida de consultas para cubos difusos con datos imprecisos tratados con etiquetas lingüísticas (OLAP Difusas) sobre una base de datos multidimensional (estrella). Esta información, ahora cualitativa, ofrece una nueva alternativa de visión de la información a los ejecutivos de las empresas.

En nuestras investigaciones existen otros ejemplos de este tipo con casos reales, que una vez implementados, nuestros usuarios han quedado satisfechos.

Conclusiones y Trabajos Futuros.

Se ha presentado la implementación de indicadores de gestión con etiquetas lingüísticas y la implementación de operaciones CUBE y VENTANA que permiten consultarlos, según su definición en el Standard SQL2003 en Oracle 10G, además de una propuesta para su utilización en implantaciones de Cubos en ROLAP que creemos que es de utilidad para los desarrolladores de sistemas de gestión de información y usuarios finales.

También, se ha comparado los planes de ejecución de las consultas que contienen estas instrucciones, permitiendo concluir que la operación

CUBE es menos eficiente que VENTANA, teniendo claro que son instrucciones distintas pero pueden cumplir un rol similar a la hora de hacer consultas, esto se encuentra más detallado en [7].

Para las consultas de gestión nuestra propuesta de extender los cubos precisos a cubos imprecisos con el uso de la lógica difusa, lo cual probamos con un caso real, es una interesante herramienta de gestión para los usuarios finales, sobre todo cuando solicitan la venta buena o excelente para saber la mejor Venta y ¿que es la venta buena? lo respondemos con nuestra investigación [12].

Como trabajo futuro, se tiene pensado complementar este trabajo con más casos de la vida real, y revisar otras instrucciones que puedan cumplir un rol similar en apoyo a la gestión.

Referencias

1. Melton Jim, “(ISO-ANSI Working Draft) Foundation (SQL/Foundation)”, ISO/IEC 9075-2:2003 (E), United States of America (ANSI), 2003.
2. Microsoft, <http://technet.microsoft.com/es-es/library/ms175939.aspx>, artículo sobre el uso del operador CUBE, 2007.
3. Peralta Verónica, “Diseño Lógico de Data Warehouses a partir de Esquemas Conceptuales Multidimensionales”, Tesis de Maestría, Universidad de la Republica Uruguay, Uruguay, 2001.
4. Inmon, W., “Building the Data Warehouse”. John Wiley & Sons, Inc. 1996.
5. Kimball R., “The DataWarehouse ToolKit”. Jhon Wiley & Son, Inc, 1996.
6. Galindo Jose, Urrutia Angélica, Piattini Mario, Fuzzy Databases: Modeling, Design and Implementation, Editorial Group Idea Publishing, USA, 2006.
7. Carlos Guzman (2008): Implementación de CUBE Ventana y Merge del Standar SQL 2003 orientado a la toma de decisions”, tesis de licenciatura, Universidad Católica de Maule, Chile, julio 2008.
8. ANSI, “American National Standard for Information Systems: Database Language SQL”, FDT, ANSI X3, 135-1986, American National Standards Institute, Nueva York.

9. ANSI, "Database Language SQL With Integrity Enhancement", ANSI X3, 135-1989, American National Standards Institute, Nueva York.
10. ANSI, "Database Language SQL", ANSI X3, 135-1992, American National Standards Institute, Nueva York.
11. Carpani Fernando, "CMDM: Un Modelo Conceptual para la Especificación de Bases Multidimensionales", Tesis de Maestría, Universidad de la República, Uruguay, 2000.
12. País Mauricio (2008): Extension de Cubos Precisos a Consultas Imprecisas con Teoría de Conjuntos Difusos, Tesis de Grado de Magíster en Ingeniería Informática. Universidad Santiago de Chile, Octubre 2008.
13. Zadeh, Lord A. Fuzzy Sets. Information Control, 8, 338-353, 1965.
14. Tineo Leonid, "Interrogaciones Flexibles en Bases de Datos Relacionales". Universidad Simon Bolivar, Enero 1998