UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN - TACNA

ESCUELA DE POSGRADO

MAESTRÍA EN COMPUTACIÓN E INFORMÁTICA

MODELO DE DATOS MULTIDIMENSIONALES PARA EL DISEÑO ÓPTIMO DE BASE DE DATOS

TESIS

PRESENTADA POR:

LIC. VICTOR YAPUCHURA PLATERO

Para optar el Grado Académico de:

MAESTRO EN CIENCIAS (*MAGÍSTER SCIENTIAE*) CON MENCIÓN EN COMPUTACIÓN E INFORMÁTICA

TACNA - PERÚ

2011

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN - TACNA
ESCUELA DE POSTGRADO

MAESTRÍA EN COMPUTACIÓN E INFORMÁTICA

MODELO DE DATOS MULTIDIMENSIONALES PARA EL DISEÑO ÓPTIMO DE BASE DE DATOS

Tesis sustentada y aprobada el 29 de abril del 2011; estando el jurado calificador integrado por:

PRESIDENTE:

Dr. Lorenzo Walter Ibárcena Fernández

SECRETARIO:

Mgr. Alberto Enrique Cohaila Barrios

MIEMBRO:

Mgr. Hugo Euler Tito Chura

ASESOR:

Mgr. Artidoro Velapatiño Castilla

DEDICATORIA

- A mi padre Eleuterio Yapuchura Huarahuara por su ejemplo perenne de honestidad y superación.
- A mi madre Francisca Platero Quispe por su aliento y cooperación.
- A mis hijos Jairo Arturo y Mayra Kiara, por ser ellos mi fuente de inspiración.

AGRADECIMIENTOS

- Al **Mgr. Artidoro Velapatiño Castilla,** mi asesor, por su valiosa orientación;
- Al **Dr. Ernesto Cuadros,** por sus orientaciones en Doctorado de ciencias de computación, UNSA;
- Al **Dr. Agustín Gutierrez,** docente de Universidad de México por sus recomendaciones sobre mi trabajo;
- A todos los que motivaron y cooperaron a dar este paso importante, de invalorable costo por el constante aprender.

Muchas gracias

CONTENIDO

CAPÍTULO III

CAPÍTULO IV

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El presente trabajo de tesis de magister describe el modelo multidimensional para la implementación de un Data Warehouse (Almacén de Datos).

El objetivo principal es determinar la eficiencias de la aplicación del modelo multidimensional en el proceso de diseño óptimo de Base de Datos, que dé soporte a las necesidades de información de gestión de los usuarios que definen la estrategia a seguir en una Institución.

Como segundo objetivo, este trabajo es caracterizar el uso de Base de datos con visión multidimensional. Los Data Warehouses deben ser diseñados estructurando los datos de una manera que puedan ser manejados por el sistema OLAP. Generalmente, se encuentra dos técnicas para modelar los Data Warehouse: el modelo multidimensional y el modelo relacional.

La estrategia relacional es muy escalable y puede manejar Data Warehouses muy grandes; sin embargo este crecimiento afecta la performance de las consultas. El enfoque multidimensional, por otra parte, tiene mucho mejor desempeño en el procesamiento de consultas, pero no es muy escalable. Ésas son las razones de la existencia de estas dos estrategias y lo que fundamenta su investigación.

En los últimos tiempos, con el advenimiento de la web, han surgido otras técnicas que utilizan tipos de datos semiestructurados tanto como fuente de datos, repositorio información multidimensional y mecanismos de intercambio. Es interesante ver la capacidad de estos tipos de datos para la implementación de Data warehouses y los trabajos que existen al respecto.

Finalmente se pretende formalizar el modelo multidimensional. La tecnología Datawarehousing debido a su orientación analítica, impone un procesamiento y pensamiento distinto, la cual se sustenta por un modelamiento de Bases de Datos propio, conocido como Modelamiento Multidimensional, el cual busca ofrecer al usuario su visión respecto de la operación del negocio.

ABSTRACT

The multidimensional fashion model for a Data Warehouse's implementation describes magister's present work of thesis (Datos's Store).

The principal objective is to determine her efficiencies give it Base's application of the multidimensional model in the process of optimal design, that give technical support to the users' needs of information of step that define the strategy to follow at an Institution of Datos.

Like second objective, this work is to characterize Base's use of data with multidimensional vision. Warehouses dates them they must be designed structuring data in a way that they may be driven by the system OLAP. Generally, you find two techniques to model Warehouse dates them: The multidimensional model and the model relacional.

Strategy relacional is very climbable and Data Warehouses can drive very large; However the performance of the consultations affects this growth. The multidimensional focus, on the other hand, has much better performance in the processing of consultations, but it is not very climbable. Those are the reasons of the existence of these two strategies and that bases its investigation.

In recent times, with the arrival of the Web, have happened another techniques that types of semi-structured data as much as source of data, repositorio utilize multidimensional information and mechanisms of interchange. It is interesting to see the capability of these types of data for Data's implementation warehouses and the works that exist with regard to this matter.

Finally it is intended to formalize the multidimensional model. Technology Datawarehousing due to his analytical orientation, imposes a processing and different thought, which holds itself for Bases's modelamiento of own Datos, known as Modelamiento Multidimensional, which attempts to offer his vision in respect of the operation of business to the user.

INTRODUCCIÓN

El aumento del volumen y variedad de información que se encuentra informatizada en bases de datos digitales y otras fuentes ha crecido espectacularmente en las últimas décadas. Gran parte de esta información es histórica, es decir, representa transacciones o situaciones que se han producido. Aparte de su función de "memoria de la organización", la información histórica es útil para explicar el pasado, entender el presente y predecir la información futura. La mayoría de las de las decisiones de empresas, organizaciones e instituciones se basan también en información sobre experiencias pasadas extraídas de fuentes muy diversas. Además, ya que los datos pueden proceder de fuentes diversas y pertenecer a diferentes dominios, parece clara la inminente necesidad de analizar los mismos para la obtención de información útil para la organización.

En muchas situaciones, el método tradicional de convertir los datos en conocimiento consiste en un análisis e interpretación realizada de forma manual.

Por ejemplo, supongamos que una cadena de supermercados quiere ampliar su local de actuación abriendo nuevos locales. Para ello, la

empresa analiza la información disponible en sus bases de datos de clientes para determinar el perfil de los mismos y hace uso de diferentes indicadores demográficos que le permiten determinar los lugares más idóneos para los nuevos emplazamientos. La clave para resolver este problema es analizar los datos para identificar el patrón que define las características de los clientes más fieles y que se usa posteriormente para identificar el número de futuros buenos clientes de cada local.

Uno de las metas es, por tanto, que nosotros seamos capaz de utilizar apropiadamente las diversas técnicas existentes en cada una de las fases de la extracción de conocimiento a partir de datos: la recopilación de datos, mediante almacenes de datos o de manera directa, la representación de datos, mediante visualización, agregación, limpieza o transformación.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

A finales de los años noventa asistimos a una nueva "ola" respecto a la utilización de la informática por parte de las organizaciones: del proceso de "automatización" de trabajo rutinario, se pasó a un proceso de verdadera "informatización", por el que la informática se convierte en un aliado estratégico de la capital importancia para la supervivencia de la organización, soportando nuevos tipos de aplicaciones más "creativas" en áreas como marketing, ventas, atención al cliente, etc. En este cambio han desempeñado un papel fundamental las bases de datos multidimensionales.

Los sistemas de gestión de base de datos se utilizan en todo tipo de sectores, siendo los sistemas de gestión de bases de datos relacionales el tipo de sistemas dominante. Estos sistemas han sido diseñados para

gestionar una alta tasa de transacciones, realizando cada transacción normalmente pequeños cambios en los datos operacionales de la organización, es decir, en los datos que la organización requiere para gestionar sus operaciones cotidianas. Este tipo de sistemas se denominan sistemas de procesamiento de transacciones en línea (OLTP, Online Transaction processing). El tamaño de las bases de datos OLTP puede ir desde base de datos de pequeño tamaño, tan sólo unos pocos megabytes (MB), a bases de datos de tamaño medio con varios gigabytes (GB), y base de datos de gran tamaño que requieren terabytes (TB) o incluso petabytes (PB) de capacidad de almacenamiento.

Se pretende mostrar modelo de datos multidimensionales con el fin de responder a las necesidades de procesamiento analítico y toma de decisiones de las organizaciones. Se analiza el concepto, junto con las ventajas e inconvenientes, de un almacén de datos y sus principales componentes. En cuanto al diseño de los almacenes de datos se presenta el conocido como diseño en estrella, que permite su implementación en SGBDR, así como los enfoques basados en el modelado conceptual del almacén de datos. Todo esto para poder diseñar óptimamente la base de datos.

Las diferencias de los Data warehouses con las bases de datos tradicionales, sobre todo al tipo de consultas y performance esperada en

las mismas, hacen que las estrategias de diseño y los modelos de datos utilizados para Data warehouse (DW) sean diferentes. En el diseño de base de datos existen escasas técnicas para la construcción de un esquema lógico relacional de DW a partir del modelo de datos multidimensionales.

Los modelos, dieron lugar a uno nuevo, el relacional, cuya simplicidad y potencia revolucionaron el mercado de las bases de datos. A pesar de ello, la evolución siguió otra rama alternativa dando lugar a las denominadas bases de datos orientados a objetos. Aunque fueron muchos los que apostaron por éstas, la realidad es que en la actualidad su uso es de menor porcentaje con respecto al modelo relacional.

1.1.1 DESCRIPCIÓN DEL PROBLEMA

Los usuarios que toman decisiones y planifican día a día, a mediano plazo o a largo plazo, la calidad, disponibilidad y presentación de la información juegan un papel categórico. Este tipo de usuarios necesitan disponer de información tanto consolidada como detallada de cómo marchan las actividades ya cumplidas, predecir tendencias y comportamientos para tomar decisiones proactivas.

Con los sistemas tradicionales se preparan reportes ad-hoc para encontrar las respuestas a algunas de las preguntas, pero se necesita dedicar aproximadamente un 60% del tiempo asignado al análisis de localización y presentación de los datos, como también asignación de recursos humanos y de procesamiento del departamento de sistemas para poder responderlas, sin tener en cuenta la degradación de los sistemas transaccionales. Esta problemática se debe a que dichos sistemas transaccionales no fueron construidos con el fin de brindar síntesis, análisis, consolidación, búsquedas y proyecciones. Por tal motivo necesita un buen diseño de base de datos.

Cada año en el mundo se multiplica la cantidad de datos almacenados en diferentes medios magnéticos. Existen referencias que ofrecen estadísticas sobre ese enorme crecimiento. Un gran porcentaje de los datos generados representan "hechos" que diariamente se están registrando tales como: transacciones financieras, operaciones de compra y venta, préstamos y devoluciones o movimientos de almacén. Si se cuenta con las herramientas adecuadas, esos datos pueden ser utilizados para detectar áreas de oportunidad o crear nuevas estrategias para las organizaciones que han estado colectándolos durante años.

Convertir los datos en información limpia y completa, útil para el análisis y el apoyo en la toma de decisiones, es una tarea compleja. Se requieren nuevas y mejores herramientas que permitan, por un lado, acceder e integrar los datos sin importar los diversos formatos y fuentes heterogéneas de las que provienen y, por otro lado, herramientas para su manipulación y visualización interactiva. Con herramientas como estas, es factible detonar acciones inmediatas de manera proactiva o reactiva que conduzcan a mejorar la operación y estrategia de las organizaciones que las usen y por lo tanto su rentabilidad.

Para desarrollar una herramienta de análisis de información, se requiere de una infraestructura de hardware poderosa y software basado en algoritmos que optimicen el uso de los recursos con los que se cuente para poder manipular grandes volúmenes de datos. Se requieren dispositivos de almacenamiento que soporten grandes volúmenes de datos y tengan excelente velocidad de respuesta, uno o más procesadores para realizar cálculos y agregados así como suficiente memoria. Por otro lado, se necesitan redes de alta velocidad que permitan copiar o mover datos de un lado a otro sin tener que esperar por horas. Se requiere de tiempos de respuesta

de minutos o segundos para analizar gigabytes o cantidades mayores de información.

Hay herramientas para análisis de información eficientes y con buen desempeño. Dichas herramientas se han enfocado en la integración de datos heterogéneos y la visualización y manipulación de los mismos de manera interactiva. Desafortunadamente las soluciones se ofrecen por separado y hasta recientemente se está trabajando en el desarrollo de paquetes de software que ofrezcan todos los servicios de manera integrada.

En este trabajo de tesis se analizan las arquitecturas de sistemas para el análisis de información y sus componentes, los modelos multidimensionales y las herramientas para navegación de datos usando procesamiento analítico en línea (OLAP, por sus siglas en inglés).

Actualmente se está investigando acerca de la integración de todas esas tecnologías para producir soluciones completas de análisis de información, se está viendo que las diferentes tecnologías pueden apoyarse unas a otras para mejorar y enriquecer el proceso de análisis de información.

1.1.2 ANTECEDENTES DEL PROBLEMA

Los modelos de los datos presentados hasta ahora difieren en el poder expresivo, complejidad y formalismo. En lo siguiente, alguna investigación trabaja en el campo de sistemas de almacenaje de datos y herramientas de OLAP sólo se resume.

Según Li, c., Wang (1996) un modelo del datos multidimensional se introduce basado en los elementos correlativos. Se planean las dimensiones como las "relaciones de la dimensión". Los cubos se modelan como las funciones del producto Cartesiano de las dimensiones a la medida y se trazan como "grupos de relaciones" a través de una definición de la pertinencia.

Según Gyssens, M., Lakshmanan (1997), se definen las tablas n-dimensionales y una cartografía correlativa se proporciona a través de la anotación de relación. Se considera que la base de datos multidimensional es compuesta del juego de tablas. Se diseñan las jerarquías del atributo a través de la introducción de dependencias funcionales en los atributos de tablas de la dimensión.

Trujillo, J. y Palomar, M. (2001), tomaron los conceptos y las ideas básicas del modelo multidimensional clásico basadas en el paradigma Orientado a Objeto-orientado. Los elementos básicos de

su O-O al Modelo Multidimensional son la dimensión clases y clases de hecho. Ellos también presentaron que el cubo clasifica como la estructura básica para permitir un análisis subsecuente de los datos guardado en el sistema.

Los aportes principales son: (a) la introducción de un modelo de los datos multidimensional formal; (b) definiciones de tres operadores del cubo, jumping, el rollingUp y drillingDown,; (c) los datos multidimensionales conceptuales se diseñan en término de clases usando UML.

Otra fuente importante de información para los procesos de toma de decisiones vendría de expertos. Sería interesante poder modelar los datos aportados por estos al proceso de forma que se integre con el resto de los datos. Los usuarios tienen una capacidad muy alta para trabajar con información imprecisa y, en la mayoría de los casos, la información que nos aporten los expertos vendría sustentada en conceptos ambiguos que para nosotros es fácil de entender pero complicado de modelar utilizando modelos matemáticos clásicos. Por esto, su integración con el resto de las fuentes implicaría la incorporación de conceptos vagamente definidos que de forma intuitiva se entienden pero cuya relación con el resto de los datos considerados sería imprecisa.

Las técnicas de extracción de reglas de asociación suelen tener el problema de obtener un conjunto muy numeroso de reglas. Esto hace que sea complicado por parte del usuario el interpretarlas. Las técnicas de inducción orientada por atributos utilizan taxonomías para reducir el número de reglas obtenidas. En el caso de basarlas en DataCubos, las jerarquías definidas en las dimensiones podrían cumplir la misma función. Además, utilizar conceptos mas cercanos al usuario en la reglas ayudaría a su interpretabilidad.

1.1.3 FORMULACIÓN DEL PROBLEMA

¿El uso de modelo de datos multidimensionales elevará significativamente en el proceso de diseño de base de datos?

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Determinar el grado de eficiencia de la aplicación de modelo de datos multidimensionales en el proceso de diseño óptimo de Base de Datos.

1.2.2 OBJETIVOS ESPECÍFICOS

- Caracterizar el uso de Base de Datos con visión multidimensional.
- Formalizar el modelo multidimensional.

1.3 JUSTIFICACIÓN E IMPORTANCIA

Las universidades nacionales, desde hace varios años, vienen trabajando en la implementación de sistemas de gestión o transaccionales que integran procesos y áreas. Estos sistemas producen datos que se almacenan en bases de datos, con dimensiones considerablemente grandes y diversidad de temas. En muchas oportunidades estos datos no están estandarizados entre áreas de una misma institución o entre distintas instituciones. Por ejemplo, la forma en que se definen los centros de costos varías entre un área de liquidaciones y un área presupuestaria. Otro ejemplo: distintas unidades académicas difieren al definir que es un alumno.

En otras oportunidades los datos no están completos, por ejemplo se quiere analizar el rendimiento académico de los ingresantes haciendo un seguimiento por tipo de escuela de proveniencia y este dato con organismo que no pertenecen al sistema universitario. Por ejemplo

supongamos que necesitamos analizar la oferta cruzada con índices de desempleo por zonas. Cada organismo puede tener su propia codificación esto hace que el trabajo a realizar sea pesado, que requiera tiempo extra de elaboración o que sea imposible de resolver según sea el caso.

O sea se dispone de los datos. La pregunta que surge en forma inmediata es si esto es suficiente para las necesidades actuales. Se puede afirmar sin duda que disponer de los datos no alcanza para cubrir las necesidades de los directivos de las universidades de analizar estos datos y cruzarlos con datos externos. Por otra parte se requiere que estos datos se presenten en forma clara y simple para el que realice la consulta y con cierta independencia del personal técnico. Sin duda necesitamos nuevas herramientas que permitan estar a la altura de las circunstancias diarias de las operaciones de soporte a las decisiones de una organización.

Data Warehouse ayuda a los directivos y tomadores de decisiones a convertir datos crudos en información valiosa. A través de este tipo de información se puede lograr una visión mas completa e integral de la organización, entender los eventos en forma sistemática permitiendo así un redefinición de estrategias.

Los beneficios de trabajar con Data Warehouse es que simplifica los procesos de toma de decisiones porque ofrece imágenes integradas de los datos. Facilita el proceso de comparación, proyección a futuro, relación con otros datos, muestra de indicadores, información consolidada, etc.

Una de las misiones más importantes en la construcción de Data Warehouse es la construcción de su esquema lógico. El Esquema lógico es una especificación más detallada que el esquema conceptual donde se incorporan nociones de almacenamiento, performance y estructuración de los datos. En el caso de diseño de Data Warehouses se debe tener en cuenta un componente adicional: las bases de datos fuente. Un DW se construye con información extraída de un cierto conjunto de bases de datos fuentes.

CAPÍTULO II

MARCO TEÓRICO

2.1 TECNOLOGÍA DE ALMACEN DE DATOS

2.1.1 INTRODUCCIÓN

Prácticamente, no existe hoy en día una faceta de la realidad de la cual no se disponga información de manera electrónica, ya sea estructurada, en forma de base de datos, o no estructurada, en forma textual o hipertextual. Desgraciadamente, gran parte de esta información se genera con un fin concreto y posteriormente no se analiza ni integra con el resto de información o conocimiento del dominio de actuación. Un ejemplo claro podemos encontrar muchas empresas y organizaciones, donde existe una base de datos transaccional que sirve para el funcionamiento de las aplicaciones del día a día. Pero que raramente se utiliza con fines analíticos. Esto se debe,

fundamentalmente, a que no se sabe cómo hacerlo, es decir, no se dispone de las personas y de las herramientas indicadas para ello.

Afortunadamente, la situación ha cambiado de manera significativa respecto a unos años atrás, donde el análisis de datos se realizaba exclusivamente en las grandes corporaciones, gobiernos y entidades bancarias, por departamentos especializados con nombre diversos: planificación y prospectiva, estadística, logística, investigación operativa, etc. Tanto la tecnología informática actual, la madurez de las técnicas de aprendizaje automático y las nuevas herramientas de minería de datos de manejo sencillo, permiten a una pequeña o mediana organización (o incluso un particular) tratar los grandes volúmenes de datos almacenados en las bases de datos.

En un entorno de competencia global, sólo aquellas organizaciones capaces de detectar, evaluar y responder rápidamente y acertadamente a cambios y tendencias en el mercado, tendrán éxito; usando tecnología punta,

utilizando de forma eficiente sus recursos y teniendo a la información como ventaja competitiva.

Los beneficios que la arquitectura DW brinda son:

- Incrementar la productividad del negocio.
- Proveer los cimientos de las decisiones ejecutivas.
- Originar en la empresa nuevas formas de hacer negocios.

Tendencia en el tiempo: En los años de la posguerra, la economía estaba orientada al producto. La prioridad estaba en la producción. En los años 70 la prioridad estaba en mejorar la calidad de los productos. En los años 80 se toma conciencia del factor tiempo. La prioridad estaba en reducir plazos (diseño y entrega). En los años 90 la prioridad estaba en la mejora de los servicios asociados al producto (servicios a los clientes, garantía). *Actualmente la prioridad esta en la personalización. Dar a cada cliente la impresión de ser único.*

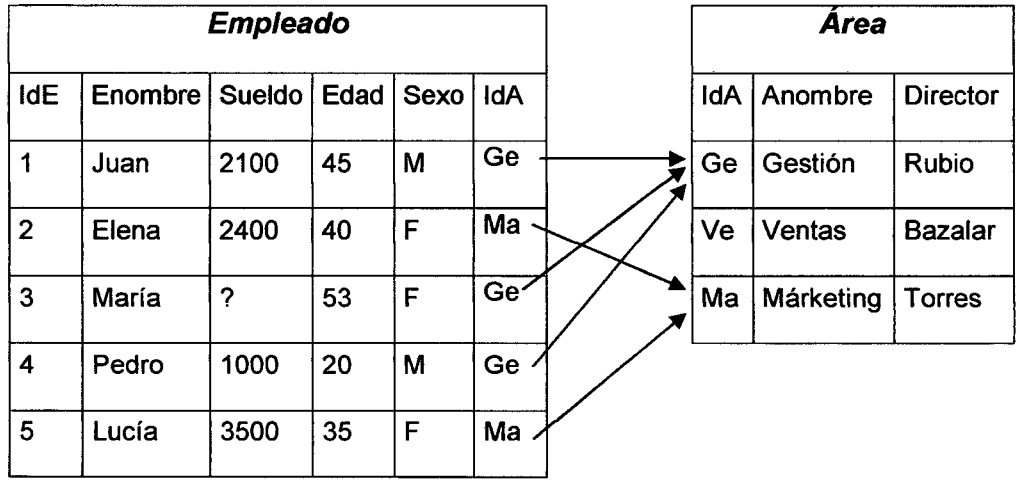
2.1.2 TIPOS DE DATOS

¿Qué tipo de dato usar? Vamos a diferenciar entre datos estructurados provenientes de bases de datos relacionales, otros tipos de datos estructurados en bases de datos (espaciales, temporales, textuales y multimedia) y datos no estructurados provenientes de la web de otros tipos de repositorios de documentos.

A) BASES DE DATOS RELACIONALES

Una base de datos relacional es una colección de relaciones (tablas). Cada tabla consta de un conjunto de atributos (columnas o campos) y puede contener un gran número de tuplas (registros o filas). Cada tupla representa un objeto, el cual se escribe a través de los valores de sus atributos y se caracteriza por poseer una clave única o primaria que lo identifica. Por ejemplo, la Figura 2.1 ilustra una base de datos con dos relaciones: *empleado* y *área*. La relación *empleado* tiene seis atributos: el identificador o clave primaria (IdE), el nombre del empleado (Enombre),

su sueldo (Sueldo), su edad (Edad), su sexo (Sexo) y el área en el que trabaja (IdA), y la relación Área tiene tres atributos: su identificador o clave primaria (IdA), el nombre (Anombre) y su director (Director). Una relación puede además tener claves ajenas, es decir, atributos que hagan referencias a otra relación, como por ejemplo el sexto atributo de la relación *empleado*, IdA, que hace referencia (por valor) al IdA de Área.



Empleado

IdE	Enombre	Sueldo	Edad	Sexo	IdA
1	Juan	2100	45	M	Ge
2	Elena	2400	40	F	Ma
3	María	?	53	F	Ge
4	Pedro	1000	20	M	Ge
5	Lucía	3500	35	F	Ma

Área

IdA	Anombre	Director
Ge	Gestión	Rubio
Ve	Ventas	Bazalar
Ma	Márketing	Torres

Figura 2.1. Base de datos relacional.

Fuente: Elaboración propia.

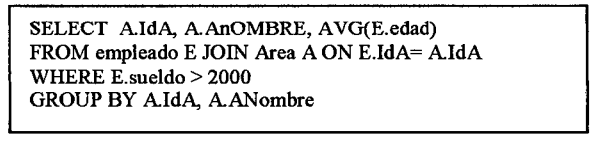
Una de las principales características de las bases de datos relacionales es la existencia de un esquema

asociado, es decir, los datos deben seguir una estructura y son, por tanto, estructurado. Así, el esquema de la base de datos del ejemplo indica que las tuplas de la relación *empleado* tienen un valor para cada uno de sus seis atributos y las de la relación Área constan de tres valores, además de indicar los tipos de datos (numérico, cadena de caracteres, etc.).

La integridad de los datos se expresa a través de las restricciones de integridad. Éstas pueden ser de dominio (restringen el valor que puede tomar un atributo respecto a su dominio y si puede tomar valores nulos o no), de identidad (por ejemplo la clave primaria tiene que ser única) y referencial (los valores de la claves ajenas se deben corresponder con uno sólo un valor de la tabla referenciada).

La obtención de información desde una base de datos relacional se ha resuelto tradicionalmente a través de lenguajes de consulta especialmente diseñados para ello, como SQL. La Figura 2.2 muestra una consulta típica SQL

sobre una relación empleada que lista la media de edad de todos los empleados de una empresa cuyo sueldo es mayor de 2000 soles, agrupada por Área.



Resultado de la consulta

IdA	Dnombre	AVG_edad
Ge	Gestión	45
Ma	Marketing	37

Figura 2.2. Consulta SQL

Fuente. Elaboración propia.

Aunque las bases de datos relacionales (recogidas o no en un almacén de datos, normalizadas o estructuradas de una manera multidimensional) son la fuente de datos para la mayoría de aplicaciones de minería de datos. Mediante una consulta (por ejemplo en SQL, en una base de datos relacionales tradicionales, o con herramientas y operadores más potentes, en los almacenes de datos) podemos combinar en una sola tabla o *vista minable* aquella información de varias tablas que requiramos para cada tarea concreta de minería de datos.

En esta presentación tabular, es importante conocer los atributos y, aunque en base de datos existen muchos tipos de datos (enteros, reales, fechas, cadenas de texto, etc.).

- Los atributos *numéricos* contienen valores enteros o reales. Por ejemplo, atributos como el *salario* o la *edad* son numéricos.
- Los atributos categóricos o nominales toman valores en un conjunto finito y preestablecido de categorías. Por ejemplo, atributos como el sexo (M, F), el nombre del Área (Gestión, Marketing, Ventas) son categóricos.

B) OTROS TIPOS DE BASES DE DATOS

Otros tipos de bases de datos que contienen datos complejos son:

- Las bases de datos espaciales contienen información relacionada con el espacio físico en un sentido amplio (una ciudad). Estas bases de datos incluyen datos geográficos, imágenes médicas, redes de transportes o

información de tráfico, etc., donde las relaciones espaciales son muy relevantes.

- Las bases de datos temporales almacenan datos que incluyen muchos atributos relacionados con el tiempo o en el que éste es muy relevante. Estos atributos pueden referirse a distintos instantes o intervalos temporales. (pueden utilizarse características de evolución)

- Las bases de datos documentales contienen descripciones para los objetos (documentos de texto) que pueden ir desde las simples palabras clave a los resúmenes. Estas bases de datos pueden contener documentos no estructurados (como una biblioteca digital de novelas), semi-estructurados (si se puede extraer la información por partes, con índices, etc.) o estructurados (como una base de datos de fichas bibliográficas). (pueden utilizarse para obtener asociaciones entre los contenidos, agrupar o clasificar)

- Las bases de datos multimedia almacenan imágenes, audio y video. Soportan objetos de gran tamaño ya que, por ejemplo, los vídeos pueden necesitar varios gigabytes de capacidad para su almacenamiento. (integrar con búsqueda y almacenamiento).

Las bases de datos objetuales y los objetos-relacionales son aproximaciones generales a la gestión de la información y, por tanto, pueden utilizarse para los mismos usos que las relaciones.

C) La World Wide Web

La World Wide Web es el repositorio de información más grande y diversa de los existentes en la actualidad. Por ello, hay gran cantidad de datos en la web de los que se pueden extraer conocimiento relevante y útil. En la Web muchos de los datos son no estructurados o semi-estructurados, a que muchas páginas web contienen datos multimedia (texto, imágenes, vídeo y/o audio), y a que

estos datos pueden residir en diversos servidores o en archivos (como los que contienen los *logs*)

2.1.3 BUSINESS INTELIGENCE

Según Vitt[1], el término de BI (Business Inteligence) es usado por diferentes expertos y fabricantes de software para distinguir un amplio rango de tecnologías, plataformas de software, aplicaciones específicas y procesos. Se utiliza este término desde tres diferentes perspectivas:

- Tomar mejores decisiones rápidamente
- Convertir los datos en información
- Utilizar un método razonable para la gestión empresarial.

El objetivo primario de la Inteligencia de Negocios es ayudar a las personas a tomar decisiones que mejoren el rendimiento de la compañía e impulsen su ventaja competitiva en el mercado. Es decir, faculta a las organizaciones a tomar las mejores decisiones rápidamente. Para tomar mejores decisiones más

[1] Vitt Elizabeth, Luckevich Michael, Misner Stacia. Business Intelligence

rápidamente, los directivos y gerentes necesitan de información relevante y útil al alcance de la mano. Pero es común una larga brecha entre la información que los responsables en la toma de decisiones requieren, y las grandes cantidades de datos que las organizaciones recopilan cada día. Para saltar esta brecha, las organizaciones hacen significativas inversiones en desarrollar sistemas de BI para convertir los datos originales en información de utilidad. Los sistemas de BI más efectivos tienen acceso a inmensas cantidades de datos para posteriormente entregar a los responsables en la toma de decisiones, información expresada de una forma que ellos pueden asimilar fácilmente. La inteligencia de Negocios puede ser definida como un método para la gestión empresarial, una forma de pensamiento organizacional y una filosofía de gestión. Tanto las personas como las organizaciones se interesan en la Inteligencia de Negocios, porque creen que el uso de un enfoque racional y basado en hechos a la hora de tomar decisiones resulta positivo en la medida que sea posible.

El interés por adoptar el BI tiene las siguientes características:

- Buscar hechos (datos) que se puedan medir cuantitativamente acerca del negocio.
- Usar métodos organizados y tecnologías para analizar los hechos.
- Inventar o compartir modelos que expliquen las relaciones de causa y efecto entre las decisiones operativas y los efectos que éstas tienen en alcanzar los objetivos de negocio.
- Experimentar con métodos alternos y supervisar con retroalimentación sobre los resultados.
- Gestión de la empresa (decisiones e iniciativas) basadas en todas estas características.

2.1.4 DSS

Según IBM (1999), un Sistema de Soporte de Decisiones (*DSS- Decision Support System*) contiene todos los servicios y procesos, para seleccionar, manipular, y analizar información y presentar resultados. Debe de

permitir acceso transparente a la data en varias partes del Data Warehouse y proveer una interfaz común para los diferentes grupos de usuarios. Un DSS también puede ser definido como un sistema computacional diseñado para apoyar en los procesos de la toma de decisiones en una organización. Un DSS es la ventana del usuario a los datos almacenados en el ambiente del Data Warehouse.

2.1.5 ALMACÉN DE DATOS

La definición más extendida es la que ha dado W. Inmon[2]: "Un almacén datos (data warehouse) es una colección de datos orientado a temas, integrados, variante en el tiempo y no volátil que soporta el proceso de toma de decisiones de la dirección". Si analizamos esta definición, encontramos que:

- Orientado a temas significa que se centra en entidades de alto nivel (como, por ejemplo, cliente, producto) no en los procesos.

[2] William Inmon, se le conoce como el "padre" del almacén de datos

- Integrados, implica que los datos se almacenan en un formato consistente. Hay que tener en cuenta la gran variedad existente en las aplicaciones de una empresa respecto a la denominación de los elementos de datos, restricciones, unidades de medida, etc. Así, por ejemplo, nos podemos encontrar con que en distintas aplicaciones, el sexo se especifica como M o F, 0 o 1, X o Y, etc., en el almacén de datos se deberá depurar ("limpiar") estos datos para poder integrarlos. Otro gran problema es el de las fechas, especificadas por medio de múltiples formatos.
- Variantes en el tiempo, los datos están asociados a un instante del tiempo (semestre, año, . . .).
- No volátiles, significa que los datos no cambian una vez que se encuentran en el almacén, de hecho este tipo de base de datos es mayoritariamente de sólo lectura.

Es conveniente resaltar el hecho de que los datos se obtienen de las fuentes de datos heterogéneas periódicamente y una vez que se almacenan en el

DW, no se modifican ya que son utilizados fundamentalmente para realizar consultas.

En la Figura 2.3 podemos observar un ejemplo de arquitectura de almacenes de datos que nos sirve para introducir las áreas de investigación que han despertado el interés de la comunidad científica de las bases de datos y que sirve como introducción para centrar el ámbito de aplicación del presente trabajo. En primer lugar, existe una serie de procesos que recogen los datos de las bases de datos operacionales o transaccionales (también denominados sistemas de procesamiento en línea[3]) y fuentes externas, los transforman, integran y realizan una carga de los datos unificados en el almacén de datos (Figura 2.3. (a)). Además, estos procesos se encargan de mantener la meta información[4] que contiene información actualizada sobre el almacén.

[3] On Line Transactional Processing, OLTP
[4] Metadata

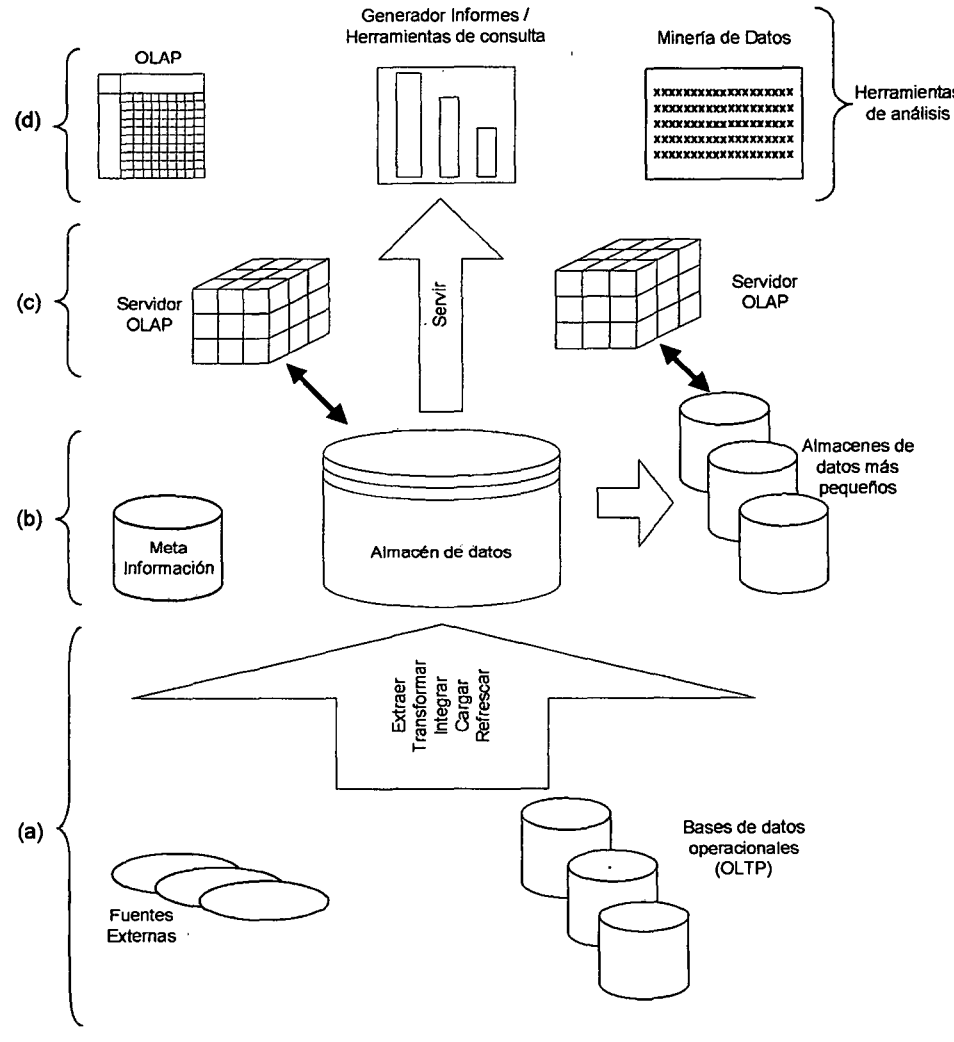


Figura 2.3. Arquitectura de un almacén de datos.

Fuente. Elaboración propia.

La información contenida en el almacén de datos se puede extraer parcialmente para formar almacenes de

datos departamentales[5], que son almacenes de datos dedicados a áreas más pequeñas que los primeros (Figura 2.3. (b)).

Sin embargo, el análisis de los datos llevado a cabo por las herramientas de análisis de los usuarios no se suele realizar directamente sobre la información guardada en el almacén de datos. Los servidores de procesamiento analítico en línea[6] se encargan de ofrecer los datos de interés para el análisis desde una perspectiva o vista multidimensional (Figura 2.3 (c)). Esto es debido a que en las decisiones estratégicas adoptadas en entornos decisionales, los datos objeto de estudio (hechos) se analizan de acuerdo a una serie de perspectivas o **dimensiones**, que en la mayoría de los casos son independientes. Para facilitar este análisis, los datos se muestran normalmente en estructuras que ofrezcan dicha vista

[5] Data marts
[6] On-Line Analytical Processing, OLAP

multidimensional, tales como cubos, hipercubos (combinación de varios cubos), tablas de hechos multidimensionales o vectores multidimensionales. Tal y como se introducirá en la siguiente sección, el **modelo multidimensional** estructura la información en función de hechos y dimensiones de una formal natural y próxima a cómo la percibe el usuario que llevará a cabo el análisis.

Estos servidores OLAP se clasifican principalmente en relacionales (ROLAP[7]) o multidimensionales (MOLAP[8]). Un servidor ROLAP es un sistema relacional extendido en el que las operaciones llevadas a cabo sobre la representación multidimensional de los datos se traducen a operaciones relacionales estándar (SQL). Por otro lado, un servidor MOLAP es un servidor que

[7] Relational OLAP
[8] Multidimensional OLAP

representa y manipula datos directamente en vectores multidimensionales.

Las herramientas de análisis (Figura 2.3. (d)), clasificadas generalmente en herramientas de generación de informes, OLAP y minería de datos, llevan a cabo una petición de datos a estos servidores OLAP. Las herramientas más clásicas son las de generación de informes, las cuales emiten un informe por cada petición de datos del usuario. Por otro lado, las herramientas más modernas son las de minería de datos, que aplican técnicas de inteligencia artificial (árboles, técnicas del vecino más próximo, redes neuronales, etc.) para intentar descubrir tendencias "escondidas" en los datos y ayudar a la toma final de decisiones. Finalmente, las herramientas OLAP proporcionan al usuario un entorno gráfico sencillo en el que los datos se presentan en una vista multidimensional. Dentro de este entorno, el usuario lleva a cabo un proceso interactivo sobre estas representaciones multidimensionales para realizar un análisis más profundo de los datos a través de ciertas

operaciones, comúnmente denominadas operaciones OLAP (Chaudhuri & Dayal, 1997).

Los objetivos más importantes de un almacén de datos son:

- Proveer una única visión de los clientes a través de toda la compañía.
- Proveer la mayor cantidad de información a la mayor cantidad de personas dentro de la organización.
- Mejorar el tiempo de emisión de algunos informes.
- Monitorear el comportamiento de los clientes.
- Mejorar la capacidad de respuesta a las cuestiones del negocio.
- Mejorar la productividad

Hay metodologías y tecnologías para realizar esta recopilación e integración. En particular introducimos la tecnología de los almacenes de datos y algunos conceptos relacionados, como las herramientas OLAP (On-Line Analytical Processing). Es importante destacar las

diferencias entre el análisis que se puede realizar con técnicas OLAP y con minería de datos (aunque exista un cierto solapamiento entre ambas), así como comprender que ambas tecnologías son complementarias.

A) DOCE REGLAS QUE DEFINEN UN ALMACÉN DE DATOS

En 1994 William y Check Kelley crearon una lista de 12 reglas que definen un almacén de datos, las cuales resumen muchos de los puntos que se han tocado con relación a los almacenes de datos.

1. El almacén de datos y los ambientes operativos están separados.
2. Los datos guardados en el almacén de datos están integrados.
3. El almacén de datos contiene datos históricos que abarcan un amplio horizonte de tiempo
4. Los datos en el almacén de datos son capturados instantáneamente en un punto dado del tiempo.

5. Los datos en el almacén de datos están orientados a sujetos.
6. Los datos en el almacén de datos principalmente son de sólo lectura con actualizaciones por lotes periódicas a partir de datos operativos. No se permiten actualizaciones en línea.
7. El ciclo de vida del desarrollo del almacén de datos difiere del desarrollo de sistemas clásicos. Los datos motivan el desarrollo del almacén de datos; los procesos motivan el método clásico.
8. El almacén de datos contiene datos con varios niveles de detalle: datos detallados actuales, datos detallados viejos, datos ligeramente resumidos y datos altamente resumidos.
9. El ambiente del almacén de datos se caracteriza por transacciones de sólo lectura de conjuntos de datos muy grandes. El ambiente operativo se caracteriza por numerosas transacciones de actualización de unas cuantas entidades de datos a la vez.

10. El ambiente del almacén de datos dispone de un sistema que rastrea fuentes, transformaciones y almacenamiento de datos.
11. Los metadatos del almacén de datos son un componente crítico de este ambiente. Los metadatos identifican y definen todos los elementos de datos; proporcionan la fuente, transformación, integración, almacenamiento, uso, relaciones e historial de cada elemento de datos.
12. El almacén de datos contiene un mecanismo de retrocarga para el uso de los recursos que exige la utilización óptima de los datos por parte de los usuarios.

Estas 12 reglas capturan el ciclo de vida del almacén de datos completo, desde su introducción, hasta sus componentes, funcionalidad y procesos de administración. La generación OLAP proporciona una amplia infraestructura para diseñar, desarrollar, ejecutar y utilizar sistemas de soporte de decisiones dentro de una organización.

B) CARACTERÍSTICAS DE LOS ALMACENES DE DATOS

La Figura 2.4 presenta una perspectiva general de la estructura conceptual de un almacén de datos. Muestra todo el proceso de almacenamiento de datos. Dicho proceso incluye la posible limpieza y reformateado de datos antes de su almacenamiento. En la parte no visible del proceso, OLAP, la minería de datos y DSS pueden generar nueva información relevante como por ejemplo, reglas. Esta información está representada en la Figura 2.4 retornando al almacén. La Figura también muestra que las fuentes de datos pueden incluir ficheros.

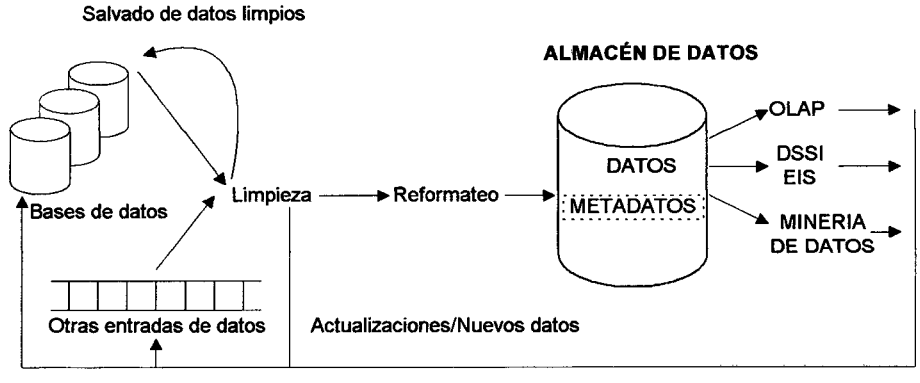


Figura 2.4. Proceso completo de almacenamiento utilizando almacenes de datos.

Fuente: RAMEZ A. ELMASRI, SHAMKANT B. NAVATHE.

Los almacenes de datos tienen las siguientes características distintivas.

- Visión conceptual multidimensional
- Dimensionalidad genérica
- Dimensiones ilimitadas y niveles de agregación
- Operaciones de dimensiones cruzadas sin restricciones.
- Tratamiento de matriz *sparse* y dinámica.
- Arquitectura cliente-servidor.
- Soporte multiusuario.
- Accesibilidad.
- Transparencia.
- Manipulación de datos intuitiva.
- Buen rendimiento al crear informes consistentes.
- Creación de informes flexibles.

Debido a que abarcan gran cantidad de datos, los almacenes de datos tienen un orden de magnitud (a veces dos) superior al de las bases de datos fuentes. El simple volumen de datos (que probablemente sea en terabytes)

- Los **almacenes de datos en grandes empresa** son proyectos de gran tamaño que requieren una enorme inversión de tiempo y recursos.
- Los **almacenes de datos virtuales** proporcionan vistas de bases de datos operacionales que se materializan para un acceso eficiente.
- Los *data marts* tienen generalmente como objetivo un subconjunto de la organización como, por ejemplo, un departamento y tienen un enfoque más riguroso.

C) ALMACENES DE DATOS Y BASES DE DATOS TRANSACCIONALES

Un almacén de datos es un conjunto de datos históricos, internos o externos, y descriptivos de un contexto o área de estudio, que están integrados y organizados de tal forma que permiten aplicar eficientemente herramientas para resumir, describir y analizar los datos con el fin de ayudar en la toma de decisiones estratégicas. La ventaja fundamental de un almacén de datos es su diseño específico y su separación de la base de datos transaccional. Un almacén de datos:

- Facilita el análisis de los datos en el tiempo real OLAP
- No disturba el OLTP de la base de datos originales.

Diferenciaremos claramente entre bases de datos transaccionales y almacenes de datos. Dicha diferencia, se ha ido marcando más profundamente a medida que las tecnologías propias de ambas bases de datos se han ido especializando. Las diferencias son claras, como se muestra en la Tabla 2.1.

Las diferencias mostradas en la tabla, distinguen claramente la manera de estructurar y diseñar almacenes de datos respecto a la forma tradicional de hacerlo con bases de datos transaccionales.

Tabla 2.1 Diferencias entre la base de datos transaccionales y el almacén de datos.

	Base de datos transaccionales	**Almacén de datos**
Propósito	Operaciones diarias. Soporte a las aplicaciones	Recuperación de información, informes, análisis y minería de datos

Tipo de datos	Datos de funcionamiento de la organización.	Datos útiles para el análisis, la sumarización, etc.
Características de los datos	Datos de funcionamiento, cambiantes, internos, incompletos	Datos históricos, datos internos y externos, datos descriptivos.
Modelo de datos	Datos Normalizados	Datos en estrella, en copo de nieve, parcialmente desnormalizados, multidimensionales...
Número y tipo de usuarios	Cientos/miles: aplicaciones, operarios, administrador de la base de datos.	Decenas: directores, ejecutivos, analistas (granjeros, mineros).
Acceso	SQL. Lectura y escritura	SQL y herramientas propias (slice & dice, drill, roll, pívot...). Lectura.

Fuente: Elaboración propia.

Aunque ambas fuentes de datos (transaccional y almacén de datos) están separadas, es importante destacar que gran parte de los datos que se incorporan en un almacén de datos provienen de la base de datos transaccional. Esto supone desarrollar una tecnología de volcado y mantenimiento de datos desde la base de datos transaccional al almacén de datos. Además, el almacén de datos debe integrar datos externos, con lo que en realidad debe estar actualizándose frecuentemente de diferentes fuentes. El almacén de datos pasa a ser un integrador o recopilador de información de diferentes fuentes, como se observa en la Figura 2.5.

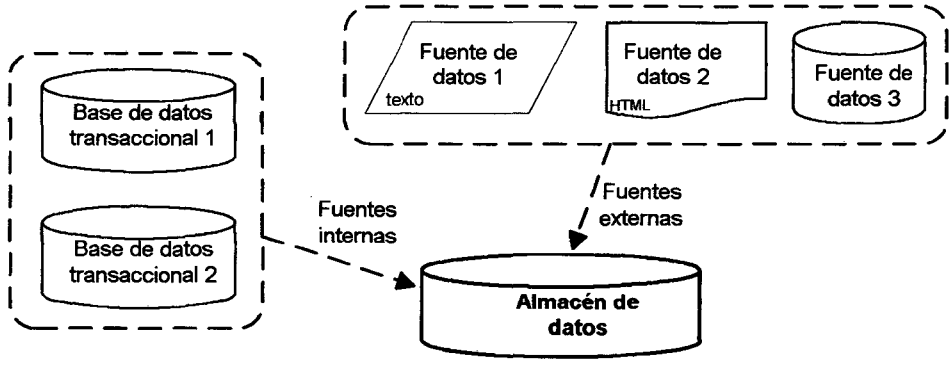


Figura 2.5. El almacén de datos como integrador de diferentes fuentes de datos.

Fuente: Elaboración propia

La organización y el mantenimiento de esta información plantean cuestiones técnicas, fundamentalmente sobre cómo diseñar el almacén de datos, como cargarlo inicialmente, cómo mantenerlo y preservar su consistencia. No obstante, son muchas más las ventajas de esta separación que sus inconvenientes. Además, esta separación facilita la incorporación de fuentes externas, que, en otro caso, sería muy difícil de encajar en la base de datos transaccional.

2.1.6 VENTAJAS E INCONVENIENTES DEL ALMACÉN DE DATOS.

El almacén de datos aporta facilidad e inmadurez en el manejo de la información, transformando datos orientados a las aplicaciones en información orientado a la toma de decisiones. Por todo ello, permite obtener grandes ventajas en la gestión, como puede ser:

- Análisis inmediato de resultados de compras, ventas o cualquier tipo de transacción comercial.
- Agilidad en el control de stocks
- Ahorro en los costos de producción

- Capacidad de analizar y explorar las diferentes áreas de trabajo.
- Gestión rápida de los programas de marketing.
- Relación total con el cliente.
- Seguridad en el control y análisis financiero.
- Facilidad en la gestión y análisis de recursos.
- Crear valor añadido.
- Cohesionar departamentos empresariales.
- Reaccionar rápidamente a cambios del mercado
- Mejorar la gestión global.

Entre los inconvenientes más graves, cabe destacar la gran inversión que supone este tipo de proyectos. Además, la tecnología no se encuentra del todo madura, y no siempre presenta un adecuado o una suficiente escalabilidad.

2.1.7 PROCESAMIENTO ANALÍTICO EN LÍNEA (OLAP)

La necesidad de un soporte más intenso para la toma de decisiones apresuró el lanzamiento de una nueva

generación de herramientas. Éstas, conocidas como procesamiento analítico en línea (OLAP) crean ambiente avanzado de análisis de datos que apoya la toma de decisiones, el modelado de negocios y las actividades de investigación de operaciones. Los sistemas OLAP comparten cuatro características principales:

- Utilizan técnicas multidimensionales de análisis de datos
- Proporcionan soporte avanzado para bases de datos
- Proporcionan interfaces de usuario final fáciles de usar
- Soporta la arquitectura cliente/servidor

A) ARQUITECTURA OLAP

Las características operativas de las herramientas OLAP se dividen en tres módulos principales:

- Interface de usuario gráfica OLAP (GUI)
- Lógica de procesamiento analítico OLAP

- Lógica de procesamiento de datos OLAP

Estos tres módulos OLAP, que residen en el ambiente cliente/servidor, hacen posible utilizar las tres características definitorias de OLAP: análisis de datos multidimensionales, soporte de base de datos avanzado e interface fácil de utilizar. La Figura 2.6. Ilustra los componentes y atributos de sistemas cliente/servidor OLAP.

Como se ilustra en la Figura 2.6, los sistemas OLAP están diseñados para utilizar tanto datos operativos como datos de almacén. Aunque la Figura 2.6 muestra que los componentes del sistema OLAP están localizados en una sola computadora, este escenario de usuario único es sólo uno de tantos.

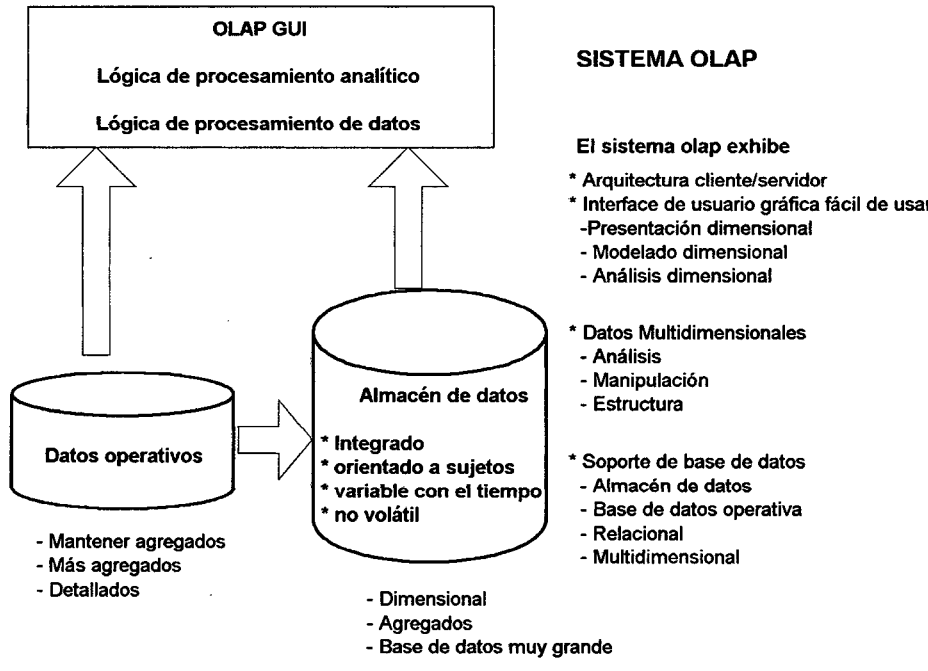


Figura 2.6 Arquitectura Cliente/servidor OLAP.

Fuente: ROB PETER, CORONEL CARLOS.

La interface gráfica de usuario podría ser un programa personalizado o, más probablemente, un módulo adicional integrado a Lotus 1-2-3 o Microsoft Excel, o alguna herramienta de análisis y consulta de datos de terceros. La Figura 2.7 ilustra un arreglo como ése.

Si se examina la Figura 2.7 se observará que el almacén de datos es creado y mantenido por un proceso o

herramienta de software independiente del sistema OLAP. Este software independiente realiza la extracción, filtración e integración necesarias para transformar los datos operativos en datos de almacén. Este escenario refleja el hecho de que, en la mayoría de los casos, las actividades de almacenamiento y análisis de los datos se manejan por separado.

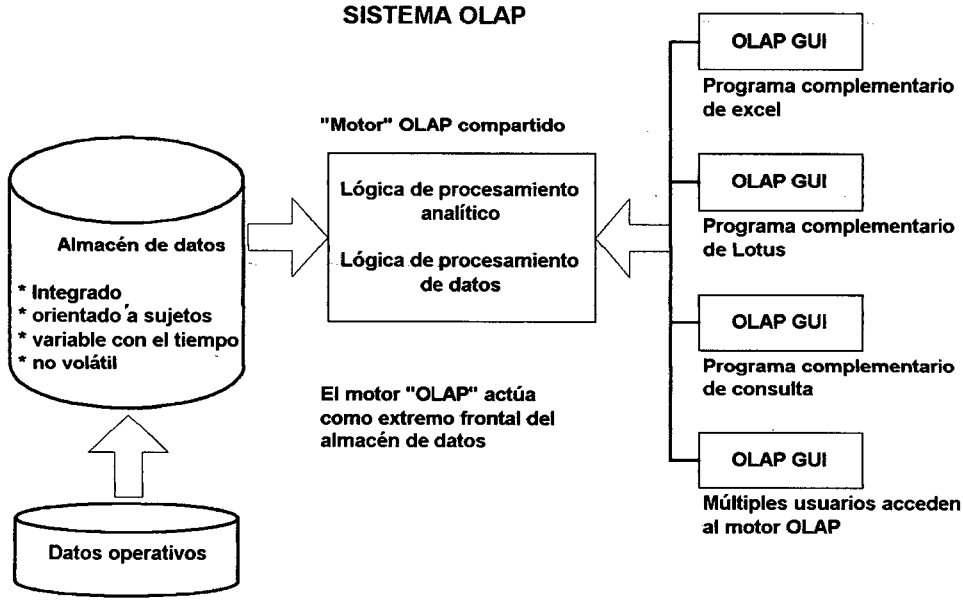


Figura 2.7 Arreglo de un servidor OLAP.

Fuente: ROB PETER, CORONEL CARLOS.

Si bien el almacén de datos representa los datos de soporte de decisiones integrados, orientados a sujetos, variables con el tiempo y no volátiles, el sistema OLAP sirve de extremo frontal a través del cual los usuarios acceden y analizan tales datos. Un sistema OLAP puede acceder directamente a los datos operativos, transformarlos y guardarlos en una estructura multidimensional. El sistema OLAP puede proporcionar el componente tienda de datos multidimensionales, como se muestra en la Figura 2.8.

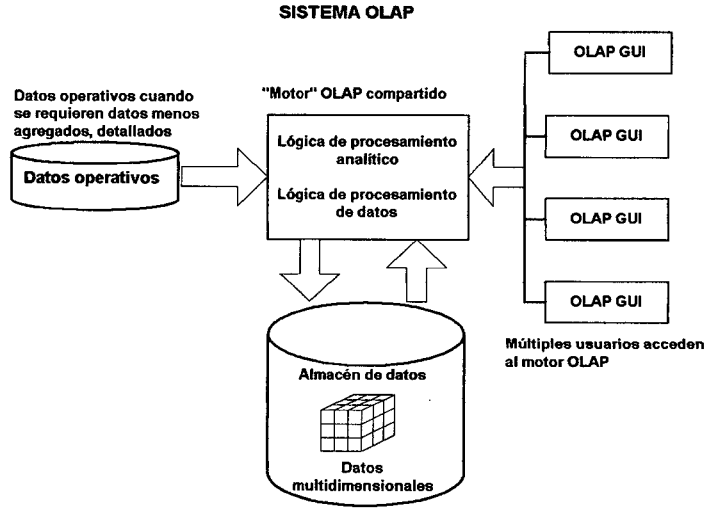


Figura 2.8 Servidor OLAP con arreglo de tienda de datos Multidimensionales.

Fuente: ROB PETER, CORONEL CARLOS.

La Figura 2.8 representa un escenario en el que el motor OLAP extrae datos de una base operativa y, luego, los guarda en una estructura multidimensional para su análisis.

Para proporcionar un mejor desempeño, los sistemas OLAP fusionan los métodos de almacén de datos y mercado de datos guardando pequeños extractos del almacén de datos en estaciones de trabajo de usuario. El objetivo es incrementar la velocidad de acceso y visualización de los datos (las representaciones gráficas de las tendencias y características de los datos). La lógica detrás de este método es la suposición de que la mayoría de los usuarios generalmente trabajan con subconjuntos de almacén de datos muy pequeños y estables. Por ejemplo, es probable que el analista de ventas trabaje con datos de ventas, mientras que es probable que un representante de un cliente trabaje con datos de clientes, y así sucesivamente. La Figura 2.9 ilustra este escenario.

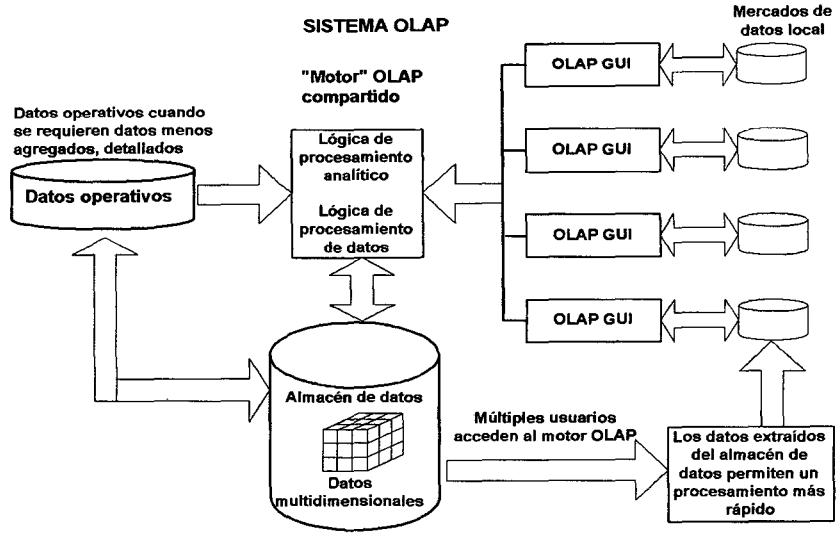


Figura 2.9 Servidor OLAP con minimercados de datos locales.

Fuente: ROB PETER, CORONEL CARLOS.

Ahora que ya se describieron las principales arquitecturas OLAP, cualquiera que sea la disposición de los componentes OLAP, una cosa es cierta: deben utilizarse datos multidimensionales. Pero ¿cómo se guardan y manejan mejor esos datos multidimensionales? Los proponentes de OLAP se encuentran muy divididos: algunos favorecen el uso de base de datos relacionales para guardar los datos multidimensionales, en tanto que

otros sostienen la superioridad de bases de datos multidimensionales para guardar datos multidimensionales.

B) OLAP RELACIONAL

El **procesamiento analítico en línea relacional (ROLAP)** proporciona funcionalidad OLAP con el uso de bases de datos relacionales y herramientas de consulta relacionales utilizadas para guardar y analizar datos multidimensionales. Este método está basado en las tecnologías relacionales existentes y representa una extensión natural de todas aquellas compañías que ya utilizan sistemas de administración de base de datos relacional en sus organizaciones. ROLAP agrega las siguientes extensiones a la tecnología RDBMS tradicional:

- Soporte de esquema de datos multidimensionales en el RDBMS
- Lenguaje de acceso a los datos y desempeño de consulta optimizados para datos multidimensionales.
- Soporte de base de datos muy grandes (VLDB)

SOPORTE DE ESQUEMA DE DATOS MULTIDIMENSIONALES DENTRO DE RDBMS

La tecnología relacional utiliza tablas normalizadas para guardar datos. La confianza en la normalización como metodología de diseño de base de datos relacionales es considerada como un obstáculo para su uso en sistema OLAP. La normalización divide las entidades de negocio en piezas más pequeñas para producir las tablas normalizadas. Por ejemplo, los componentes de datos de ventas podrían guardarse en cuatro o cinco tablas diferentes. La razón para la utilización de tablas normalizadas es reducir las redundancias, con lo cual se eliminan las anomalías en los datos, y facilitar su actualización.

Afortunadamente para aquellos que invirtieron mucho en tecnología relacional, ROLAP utiliza una técnica de diseño especial que permite que la tecnología RDMBS soporte representaciones de datos multidimensionales. Esta técnica de diseño especial se conoce como **esquema en estrella**.

LENGUAJE DE ACCESO A LOS DATOS Y DESEMPEÑO DE CONSULTA OPTIMIZADOS PARA DATOS MULTIDIMENSIONALES.

Otra crítica de la base de datos relacionales es que el SQL utilizado con RDBMS no es adecuado para la realización de análisis de datos avanzado. La mayoría de las solicitudes de datos de soporte de decisiones requiere el uso de consultas SQL de múltiples pasadas o múltiples sentencias SQL anidas. Para responder a esta crítica, ROLAP amplía el SQL de modo que pueda diferenciar entre los requerimientos de acceso a datos del almacén de datos (basado en el esquema en estrella) y los datos operativos (tablas normalizadas). De esta manera, el sistema ROLAP es capaz de generar apropiadamente el código SQL requerido para acceder los datos en el esquema en estrella.

La Figura 2.10 muestra la interacción de los componentes ROLAP cliente/servidor de 3 filas.

SOPORTE DE BASES DE DATOS MUY GRANDES

El soporte de VLDB[9] es un requerimiento obligatorio de las bases de datos DSS. Si se utiliza la base de datos relacional como DSS, también debe ser capaz de guardar cantidades de datos muy grandes. Los datos de soporte de decisiones normalmente se cargan en modo masivo provenientemente de datos operativos. La velocidad de las operaciones de carga es importante, porque los sistemas operativos funcionan 24 horas al día, 7 días a la semana.

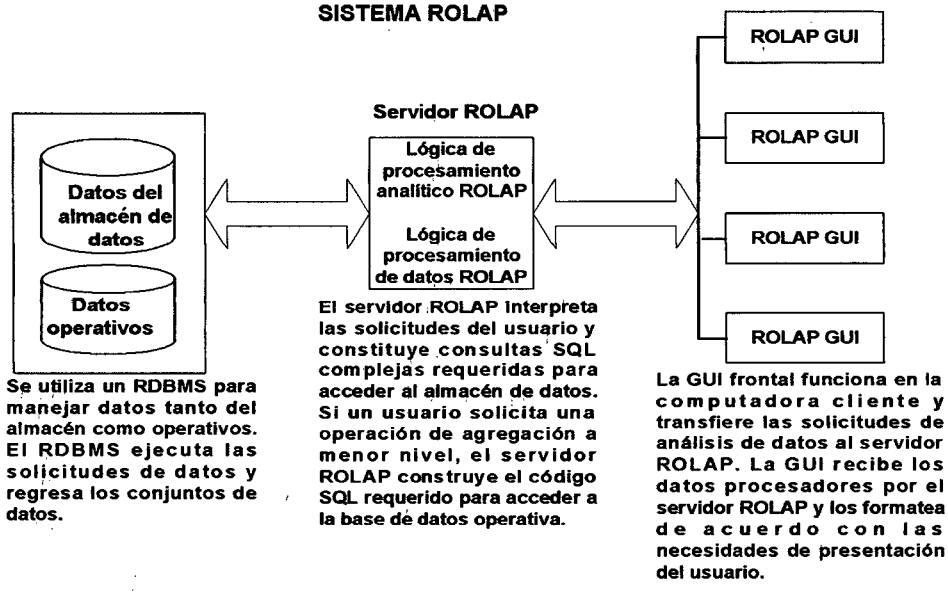


Figura 2.10 Arquitectura Cliente/servidor ROLAP típica.

Fuente: ROB PETER, CORONEL CARLOS.

[9] Base de datos muy grandes

C) OLAP MULTIDIMENSIONAL

El procesamiento analítico en línea multidimensional (MOLAP, por sus siglas en inglés) amplía la funcionalidad de OLAP a sistemas de administración de bases de datos multidimensionales (MDBMS, por sus siglas en ingles). (Un MDBMS utiliza técnicas patentadas especiales para guardar datos en arreglos en forma de matriz o de n-dimensiones.) La premisa de MOLAP es que las bases de datos multidimensionales son más adecuadas para manejar, guardar y analizar datos multidimensionales. La mayoría de las técnicas patentadas utilizadas en MDBMS se derivan de campos de ingeniería como diseño asistido por computadora/manufactura asistida por computadora (CAD/CAM) y sistemas de información geográfica (SIG).

Los proponentes relacionales también argumentan que la utilización de soluciones patentadas dificulta la integración del MDBMS con otras fuentes de datos y herramientas utilizadas en la empresa. No obstante, a pesar del hecho de que se requiere una inversión de tiempo y esfuerzo considerable para integrar la tecnología nueva y la

arquitectura de sistemas de información existentes, MOLAP puede ser una buena solución para aquellas empresas en las que las bases de datos pequeños a medianas son la norma y la velocidad del software de aplicación es crítica.

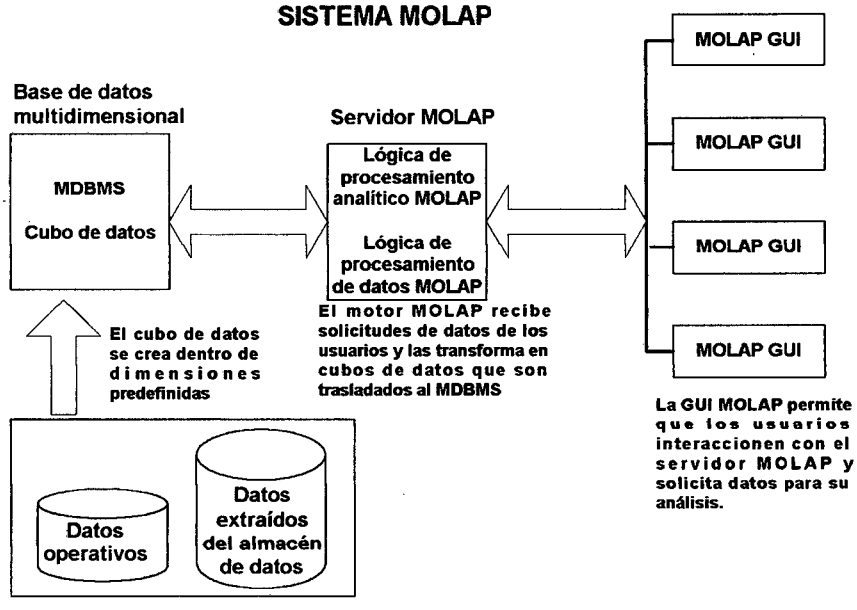


Figura 2.11 Arquitectura cliente/servidor MOLAP

Fuente: ROB PETER, CORONEL CARLOS.

D) OLAP RELACIONAL VS. MULTIDIMENSIONAL

La Tabla 2.2 resume algunos puntos a favor y otros en contra. No obstante, debe hacerse hincapié en que

algunas de las ventajas de uno sobre el otro pueden ser reacomodadas conforme avanza la tecnología. Por ejemplo, los procesadores más rápido y las computadoras más poderosas podrían hacer que los argumentos de velocidad y tamaño carezcan de valor. Tome en cuenta, también, que la selección de uno o del otro con frecuencia depende del punto de vista del evaluador.

Tabla 2.2 OLAP relacional vs. Multidimensional

CARACTERÍSTICA	ROLAP	MOLAP
Esquema	Utiliza esquema en estrella Pueden agregarse más dimensiones dinámicamente	Utiliza cubos de datos Las dimensiones adicionales requieren la recreación del cubo de datos.
Tamaño de la base de datos	Mediano a grande	Pequeño a mediano
Arquitectura	Cliente/servidor Basada en estándares Abierta	Cliente/servidor Patentada
Acceso	Soporta solicitudes ad hoc Dimensiones ilimitadas	Limitado a dimensiones predefinidas
Recursos	Alta	Muy altos

Flexibilidad	Alta	Baja
Estabilidad	Alta	Baja
Velocidad	Buena con conjuntos de datos pequeños; promedio con conjuntos de datos medianos a grandes	Rápida con conjuntos de datos pequeños a medianos; promedio con conjunto de datos grandes.

Fuente: ROB PETER, CORONEL CARLOS.

E) DIFERENCIAS ENTRE ROLAP Y MOLAP

Por lo general, las implementaciones de MOLAP presentan mejor rendimiento que la tecnología relacional; sin embargo, tienen problemas de escalabilidad, por ejemplo, la adición de dimensiones a un esquema ya existente. La Tabla 2.3 resume las diferencias entre ambas tecnologías.

Tabla 2.3 Diferencias entre ROLAP y MOLAP

	Multidimensional	Relacional
Datos	Detalle y precalculados (agregados)	Detalle y agregados
Estructura	Matrices	Tablas

	comprimidos	relacionales
Administración	Especialistas en BDMD	Administrador BD
Acceso	Lenguaje especializado	SQL

Fuente: ROB PETER, CORONEL CARLOS.

Las Tablas 2.4 y 2.5 detallan las ventajas y desventajas de cada tipo de almacenamiento.

Tabla 2.4 Ventajas y desventajas de ROLAP

Ventajas	**Desventajas**
• Se puede aprovechar la tecnología relacional, ya que facilita aprovechar las inversiones realizadas en hardware y en SGBD relacionales. • Uso de la seguridad e integridad de los SGBD relacionales • Capaz de manejar conjuntos de datos muy grandes, por encima de un terabyte.	• Es menor en rendimiento frente a MOLAP en BD pequeñas. • Limitación para consultas complejas, debido a que se puede requerir muchas

• Pueden utilizarse SGBD relacionales gratuitos. • Puede soportar un gran número de dimensiones. • Son escalable (adición de dimensiones a un esquema existente).	reuniones para obtener la consulta deseada. • Utilización de mucho almacenamiento en disco.

Fuente: ROB PETER, CORONEL CARLOS.

Tabla 2.5 Ventajas y desventajas de MOLAP

Ventajas	Desventajas
• Ofrece buen rendimiento cuando se trabaja sobre datos agregados, totales, subtotales, series temporales y diversos grados de detalle de los datos. • Facilita el estudio a alto nivel de datos, al ofrecer una mayor flexibilidad y rapidez de acceso para el análisis de los mismos. • Almacenamiento de datos y consultas bastantes eficientes. • La complejidad de la BD se	• La asimilación de los conceptos multidimensionales, en especial, cuando se tienen hipercubos de muchas dimensiones. • La construcción y poblado de las estructuras multidimensionales pueden demandar mucho tiempo. • Están limitados a tener

oculta a los usuarios. • El análisis se hace sobre datos agregados y métricas o indicadores precalculados. • Mayor rendimiento frente a ROLAP en el procesamiento de consultas en BD pequeñas. • Almacena agregados para facilitar un acceso rápido.	diez o menos dimensiones debido a la complejidad para el manejo de las mismas. • No se puede acceder a datos que no están en el cubo. • Debe trabajar con volúmenes de datos limitados, menos de 5GB. • Existen pocas herramientas gratuitas que lo soporten.

Fuente: ROB PETER, CORONEL CARLOS.

SGBD con soporte para ROLAP y MOLAP

Entre los SGBD que permiten utilizar almacenamiento de datos de tipo ROLAP y que han incorporado características adicionales para su manejo están Oracle, DB2 y SQL Server. Por otro lado, entre los SGBD que

permiten utilizar almacenamiento de datos de tipo MOLAP están:

- **SQL Server - Microsoft Analysis Services:** soporta la construcción y gestión de cubos multidimensionales, permite flexibilidad en los modos de almacenamiento, ya que también soporta ROLAP.
- **Hyperion:** fabricante de herramientas analíticas que se apoyan en OLAP. Hyperion Essbase OLAP Server es la plataforma empresarial para la elaboración de informes, análisis, modelos y presupuestos
- **Oracle Express:** contiene herramientas y aplicaciones que se apoyan en Oracle Express Server, un motor de cálculo y gestor de memoria caché de datos. Las herramientas Oracle OLAP toman en consideración todo lo referente a las necesidades de los usuarios, desde consultas y análisis simples de los datos contenidos en un DW, hasta análisis, presupuestación y modelaje sofisticados y desarrollo de aplicaciones OLAP orientados a objetos (Audifilm Grupo Brime, 2003).

2.1.8 MODELADO DE DATOS PARA ALMACENES DE DATOS

Los modelos multidimensionales se benefician de las relaciones inherentes a los datos para llenar de datos los matrices multidimensionales denominadas cubos de datos. (Estos pueden recibir el nombre de hipercubos si tienen más de tres dimensiones). Para aquellos datos que se prestan por sí mismos a un formato dimensional, el rendimiento de las consultas sobre matrices multidimensionales puede ser mucho mejor que en el modelo de datos relacional. Tres ejemplos de dimensiones en el almacén de datos empresarial serían los períodos, productos y locales de la empresa.

Los productos podrían mostrarse a modo de filas, con los ingresos de las ventas de cada local contenidos en las columnas (La Figura 2.12 representa esta organización bidimensional). Si añadimos una dimensión temporal, como los trimestres impositivos de la organización, tendríamos una matriz tridimensional, que quedaría representada mediante un cubo de datos.

En la Figura 2.13 vemos un cubo tridimensional que organiza los datos de venta de productos por trimestres y locales de ventas. Cada celda contiene los datos de un producto concreto, un trimestre impositivo concreto y una región concreta. Si incluyésemos dimensiones adicionales, podríamos obtener un hipercubo aunque no se podrían visualizar fácilmente más de tres dimensiones ni representarlas gráficamente.

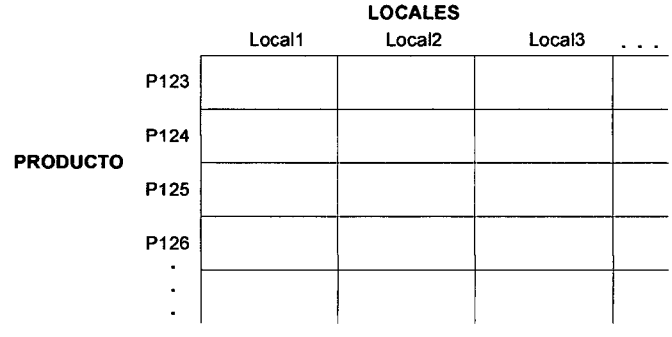


Figura 2.12. Matriz bidimensional

Fuente: Elaboración propia

Los datos pueden consultarse directamente en cualquier combinación de dimensiones, evitando las consultas de bases de datos complejas. Existen herramientas para visualizar los datos según sean las dimensiones elegidas por el usuario. El cambio de una jerarquía dimensional

(orientación) a otra se logra fácilmente en un cubo de datos mediante una técnica denominada **pivotación** (también llamada rotación). Por ejemplo, se podría pivotear el cubo de datos para mostrar los ingresos de ventas locales a modo de filas, quedando como columnas el total de ingresos por trimestre, y los productos de la empresa en la tercera dimensión (Figura 2.14). Por tanto, esta técnica equivale a tener una tabla de ventas locales por separado para cada producto, donde cada tabla representa las ventas trimestrales de ese producto local por local.

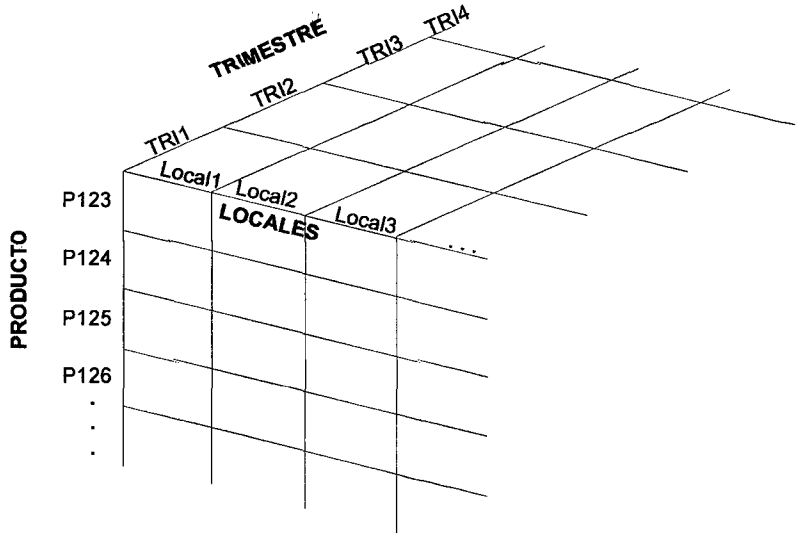


Figura 2.13. Cubo de datos

Fuente: Elaboración propia

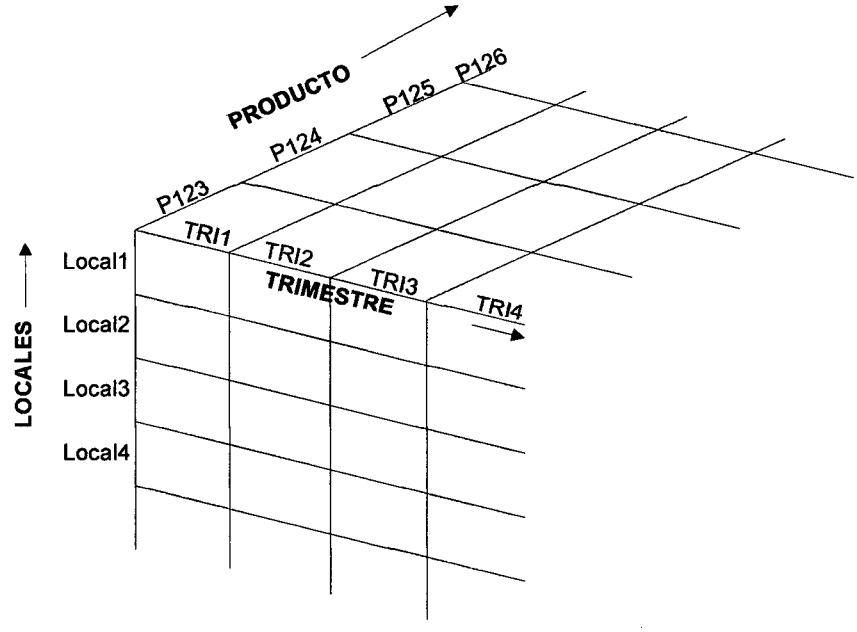


Figura 2.14. Versión pivoteada del cubo de datos

Fuente: Elaboración propia

Los modelos multidimensionales se prestan fácilmente a representaciones jerárquicas en lo que se conoce como exploración ascendente (*rroll-up*) y exploración descendente (*drill-down*). La **exploración ascendente** desplaza la jerarquía hacia arriba, agrupándose en unidades mayores a través de una dimensión (por ejemplo,

resumiendo los datos semanales en trimestres o en anuales).

La Figura 2.15 ilustra una exploración ascendente que va de productos individuales a categorías de productos más generales. En la Figura 2.16 se muestra una **exploración descendente** en la que se ofrece la función contraria, donde se da una visión más concreta, disgregando las ventas nacionales en ventas por local y después éstas en ventas por sublocales, además de clasificar los productos por tipos.

El modelo de almacenamiento multidimensional está compuesto por dos tipos de tablas: tablas de dimensiones y tablas de hecho. Una **tabla de dimensiones** está formada por tuplas de atributos de la dimensión. Una **tabla de hechos** está compuesta por tuplas, una por cada hecho registrado. Este hecho contiene alguna variable o variables medidas u observadas y la identifica con punteros a las tablas de dimensiones. La tabla de hechos contiene los datos. Las dimensiones identifican cada tupla en esos datos. La Figura 2.17 presenta un ejemplo de una

tabla de hechos que puede verse desde la perspectiva de tablas de varias dimensiones.

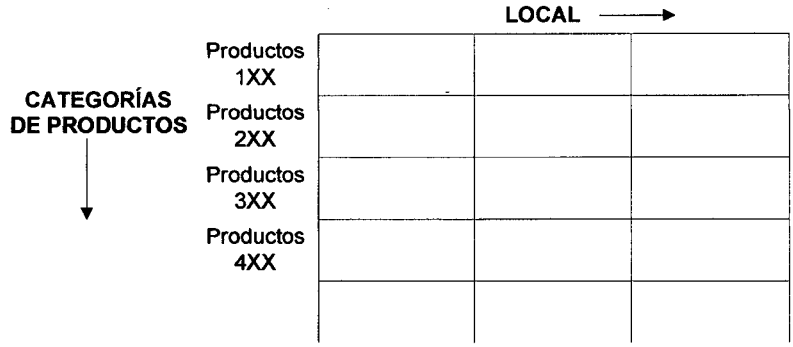


Figura 2.15. Operación de exploración ascendente (roll-up)

Fuente: Elaboración propia

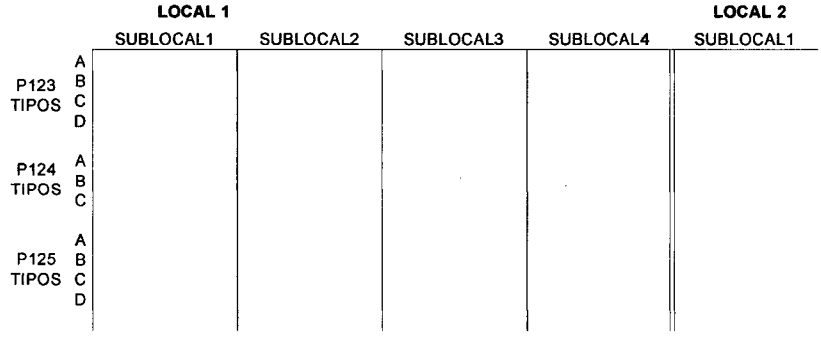


Figura 2.16. Operación de exploración descendente (drill-down).

Fuente: Elaboración propia

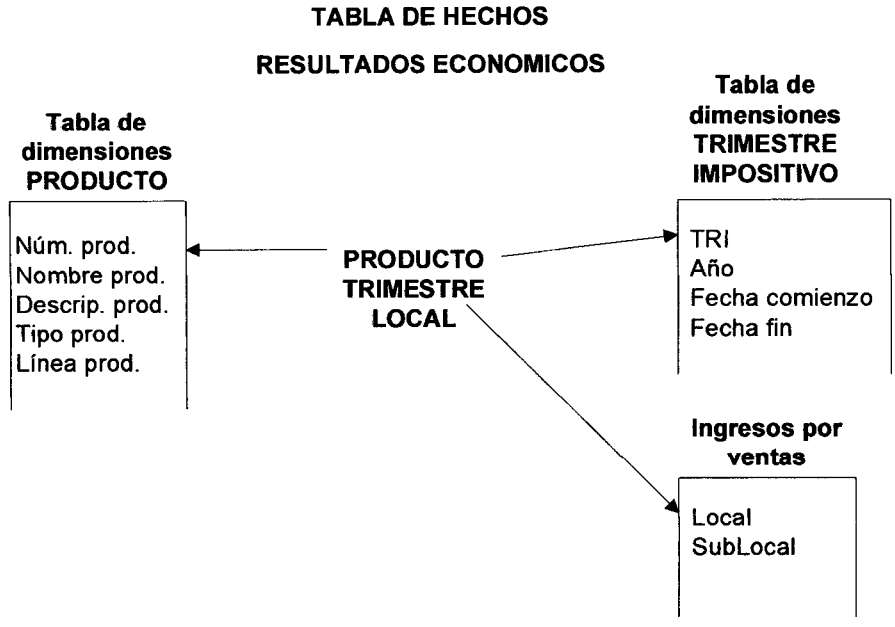


Figura 2.17 Esquema de estrella con tablas de hecho y dimensionales

Fuente: Elaboración propia

Dos esquemas multidimensionales comunes son el esquema de estrella y el esquema de copos. El **esquema de estrella** está formado por una tabla de hechos con una única tabla para cada dimensión (Figura 2.17). El **esquema de copos** es una variante del esquema de estrella en el que las tablas dimensionales de este último se organizan jerárquicamente mediante su normalización (Figura 2.18). Algunas instalaciones son almacenes de datos normalizados hasta la tercera forma normal a fin de

que se pueda acceder al almacén de datos con el máximo nivel de detalle. Una **constelación de hechos** es un conjunto de tablas de hechos que comparten algunas tablas de dimensiones.

La Figura 2.19 muestra una constelación de hechos con dos tablas de hechos, resultados económicos y predicción económica. Éstas comparten la tabla de dimensión denominada producto. Las constelaciones de hechos limitan las consultas que pueden hacerse al almacén.

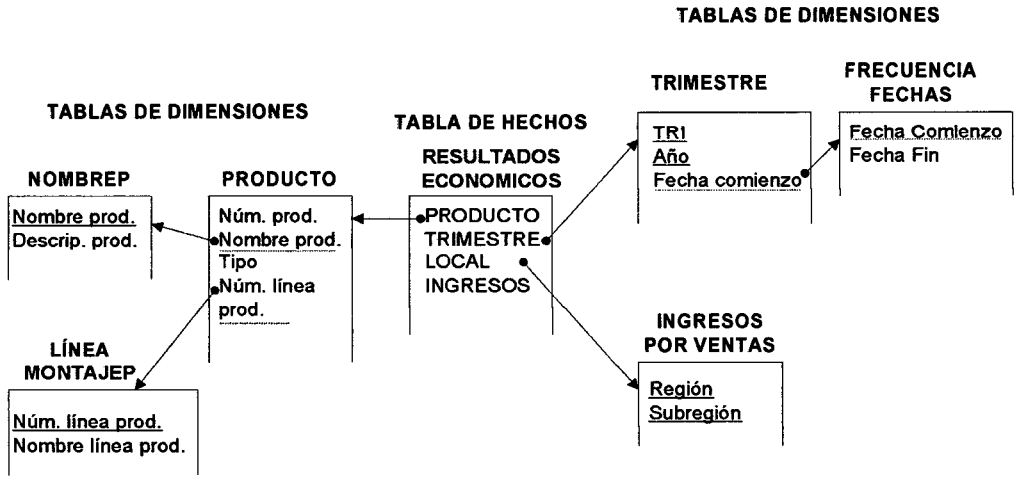


Figura 2.18 Esquema de copos.

Fuente: Elaboración propia

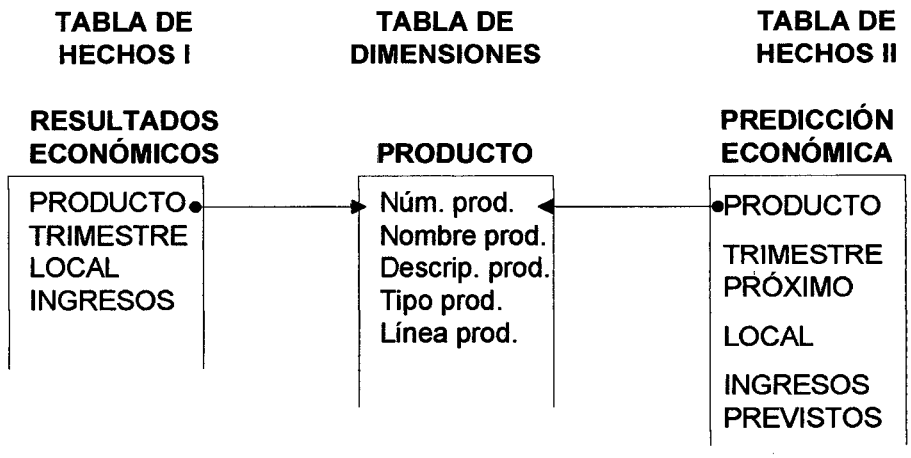


Figura 2.19 Una constelación de hechos.

Fuente: Elaboración propia

2.1.9 DATA MARTS

Un Data Mart, es un subconjunto del Data Warehouse, con un alcance de contenido limitado. Éste se usa para un solo departamento de una organización y/o un problema particular de análisis dentro de la organización. Un Data Mart por si solo, no es un Data Warehouse, ya que un Data Warehouse tiene más usuarios y más temas que un Data Mart, y provee una vista completa de las áreas

funcionales de la organización. Un Data Mart, al igual que un Data Warehouse, consiste en una base de datos. Así mismo, Vitt[10] define el Data Warehouse como un repositorio colectivo y centralizado que nutre o alimenta una serie de almacenes que tienen una orientación específica o dominio específico, o tema específico, llamados Data Marts.

El almacén de datos estará formado por muchas estrellas (jerárquicas o no), formando una "constelación". Por ejemplo, aparte de la estrella jerárquica para las ventas, podríamos tener otra estrella para personal. En este caso, hechos podrían ser que un empleado ha dedicado ciertos recursos en un proyecto durante un período en un departamento, Los hechos podrían llamarse "participaciones". Las medidas o atributos podrían ser "horas de participación" "número de participantes", "presupuesto", nivel de éxito del proyecto", etc. y las dimensiones podrían ser "tiempo" (para representar el período en el que ha estado involucrado), "departamento"

[10] Vitt Elizabeth: Técnicas de análisis para la toma de decisiones estratégicas

(para representar un empleado, equipo, departamento o división en la que se ha desarrollado) y el "proyecto" (subproyecto, proyecto o programa).

Cada una de estas estrellas que representan un ámbito específico de la organización se denomina popularmente "datamart" (mercado de datos). Lógicamente, cada datamart tendrá unas medidas y unas dimensiones propias y diferentes de las demás. La única dimensión que suele aparecer en todos los datamarts es la dimensión *tiempo*.

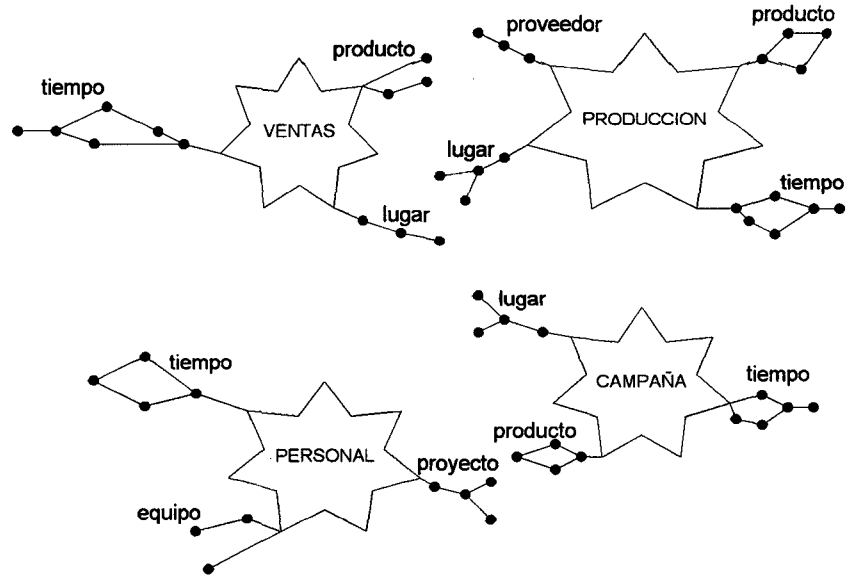


Figura 2.20. Representación icónica de un almacén de datos compuesto por varios datamarts.

Fuente: Elaboración propia.

El conocimiento de los meta datos es tan esencial como el conocimiento de los datos del Data Warehouse.

2.1.10 ESTADO DE LOS SISTEMAS ACTUALES

Es posible que la razón táctica más importante de construir un Data Warehouse sea lo inadecuado de los sistemas actuales y la falta de información empresarial, incluso cuando la empresa esta inundada en datos. Muchos sistemas de producción no satisfacen las necesidades del usuario empresarial. Por lo regular, los datos son inaccesibles e inconsistentes, tanto en forma como en significado. Por ejemplo, debido a la inconsistencia de datos, no coincide la información de ventas en diferentes reportes, la empresa carece de una imagen precisa de su ingreso. La falta de medidas comunes significa que los administradores no tienen una imagen clara del desempeño del negocio.

Los gerentes de comercialización y ventas requieren tener un acceso más rápido a los datos, mas reportes y a mayor velocidad, análisis expeditos y reacciones oportunas para administrar el negocio y aumentar los ingresos. Incluso con costos sustanciales en tecnología de la información para crear y generar reportes, éstos son tardíos y, por 1o tanto, la información pierde su novedad.

2.1.11 LA CONTRIBUCION DE MICROSOFT A LA INDUSTRIA DE DATAWAREHOUSING

Microsoft Corporation se encuentra contribuyendo al rápido crecimiento de data warehousing para construir sistemas de soporte a las decisiones. Una combinación entre productos de Microsoft y los de una alianza de proveedores de servicios y de software independientes, les permite a los clientes operar sistemas de data warehouses que sean potentes y tengan un precio accesible. La base de datos del Microsoft® SQL Server™, implementada como un motor de acumulación de información para data

warehouses, ofrece el precio/rendimiento, facilidad de administración, y la integración entre Windows NT y BackOffice que convierte al SQL Server en la plataforma de soluciones que se emplea preferentemente en muchos sistemas de data warehouse y de data mart.

Durante los últimos veinte años, Microsoft Corporation ha contribuido a que cada año la tecnología de la información esté al alcance de un mayor número de personas al reducir el costo y los retos asociados con la implementación de dicha tecnología. Microsoft se encuentra realizando en la actualidad una contribución similar en el campo de data warehousing y, por lo tanto, apoya de forma directa el rápido desarrollo de ese segmento de la industria de la tecnología de la información.

La base del concepto de data warehousing de Microsoft es el sistema de administración de bases de datos relacionales, Microsoft SQL Server (RDBMS), en el sistema operativo Windows NT. Intelligent Solutions, Inc.,

compara los data warehouses con los data marts de la siguiente manera:

Tabla 2.6 comparación de Data Warehouse y data mart

DATA WAREHOUSE	DATA MART
Construido para satisfacer las necesidades de información de toda la empresa.	Construido para satisfacer las necesidades de una función o unidad comercial específica.
Diseñado para optimizar la integración y la administración de los datos fuente.	Diseñado para optimizar la entrega de información de soporte a decisiones.
Administra grandes cantidades de historia a nivel atómico.	Primordialmente se concentra en administrar resúmenes y/o datos de muestreo.
Pertenece a, y se administra por, las organizaciones de Sistemas de Información (IS) de la empresa.	Puede ser propiedad de, y administrado por, el grupo de Sistema de Información (IS) en la Línea del Negocio.

Fuente: Elaboración propia

2.2 MODELO MULTIDIMENSIONAL

2.2.1 DATA WAREHOUSING

Inmon identifica algunos de los principales problemas que surgieron cuando los usuarios comenzaron a realizar actividades orientadas al análisis a partir de datos desestructurados extraídos de bases de datos transaccionales. Encontrar los datos pertinentes para una estrategia de consulta era todo un problema. Y sobre todo porque se producían inconsistencias en los informes acerca del mismo fenómeno debido a la inconsistencia de datos que se presentaban en las tablas.

Definición 2.1 ***(Inmon y Hackathorn, [IH94])***. *Un* ***Data Warehouse*** *es una colección de datos orientados al tema, integrados, temporales, y no-volátiles para la toma de decisiones.*

Definición 2.2 ***(OLAP Council, [OLAa])*** *"OLAP es una categoría de tecnología software que permite a los* **analistas**, *directivos y ejecutivos acceder a los datos de forma rápida, consistente e interactiva a través de una*

amplia variedad de vistas de la información que han sido obtenidas de datos sin procesar para reflejar la dimensionalidad real de la empresa como la entiende el usuario".

2.2.2 OLAP y OLTP

Como ya hemos visto, OLAP sirve para realizar un análisis exhaustivo de los datos que se han ido almacenando a lo largo de la historia de una empresa. Millones de registros que necesitan ser analizados por las personas más responsables de la empresa para lograr una mejor producción y reducción de costos.

Dado el objetivo que tienen estos sistemas, en algunas ocasiones se han denominado también "Sistemas de Soporte a la Decisión" y en muchos sistemas comerciales OLAP y DSS se han identificado.

Los requerimientos funcionales y de rendimiento que presentan las aplicaciones OLAP son completamente diferentes de las que se dan en los sistemas orientados al

procesamiento de transacciones on-line (on-line transaction processing) OLTP.

Las aplicaciones de tipo OLTP son fundamentalmente de transacciones cortas, atómicas y aisladas. Además estas transacciones requieren datos detallados y al día y afectan fundamentalmente a pocos registros a los cuales se accede fundamentalmente a través de la llave primaria. Las bases de datos sobre las que se opera son de gran tamaño (cientos de megabytes o algunos gigabytes) y por tanto la consistencia y capacidad de recuperación de las mismas son de vital importancia además del criterio esencial de rendimiento es optimizar la gestión de transacciones. Por todo ello la base de datos se diseña para estos objetivos.

En la Tabla 2.7 viene recogidas de forma esquemática las principales diferencias entre ambos enfoques.

Un sistema OLTP procesa un número muy elevado de transacciones por día. Cada transacción contiene una pequeña porción de datos. Un Data Warehouse a menudo

procesaría una transacción por día. Pero esta transacción contiene miles o millones de registros. Más que llamarlo transacción se debería denominar carga de datos productivos. En el caso de los servidores OLAP, estos pueden dar respuestas a un número pequeño de consultas que pueden necesitar el trabajar con información resumida.

Tabla 2.7: Principales diferencias entre sistemas OLTP y OLAP

Aspectos	**Base de datos clásica (OLTP)**	**Data Warehouse**
Usuarios	Diseñadores, DBAs, Operadores de entrada de datos	Decisiones, ejecutivos
Función	Operaciones diarias (on-line)	Soporte de decisiones, procesa-miento analítico
Diseño	Orientado a aplicaciones	Históricos, resumidos, multidimensionales, integrados

Datos	Actuales, atómicos, relacionales, aislados.	Ad-hoc
Uso	Repetitivo, rutinario	Lectura, consultas complejas
Acceso	Lectura/escritura, transacciones simples	Gestión de consultas, datos ajustados (organizados)
Necesidades	Gestión de transacciones, datos consistentes	

Fuente: DE MIGUEL CASTAÑO ADORACIÓN

Los usuarios de un Data Warehouse, por otro lado, están vigilando el funcionamiento de la organización. Estos vigilan qué datos son nuevos, y estudian los datos erróneos para corregirlos. Estos usuarios normalmente nunca tratan con un registro o cuenta en un momento dado. Más bien realizan consultas que implican informes más o menos grandes, y para realizarlas se requiere que grandes cantidades de registros sean buscados y

resumidos en una pequeña respuesta. Además, siempre están modificando el tipo de consultas que quieren realizar a la base de datos. Normalmente los usuarios de este tipo de sistemas están en la parte alta de una empresa, es decir, aquella en la que se toman las decisiones importantes para el futuro de la compañía (Figura 2.21), actuando el data warehouse como repositorio de conocimiento para la gestión del conocimiento en la empresa.

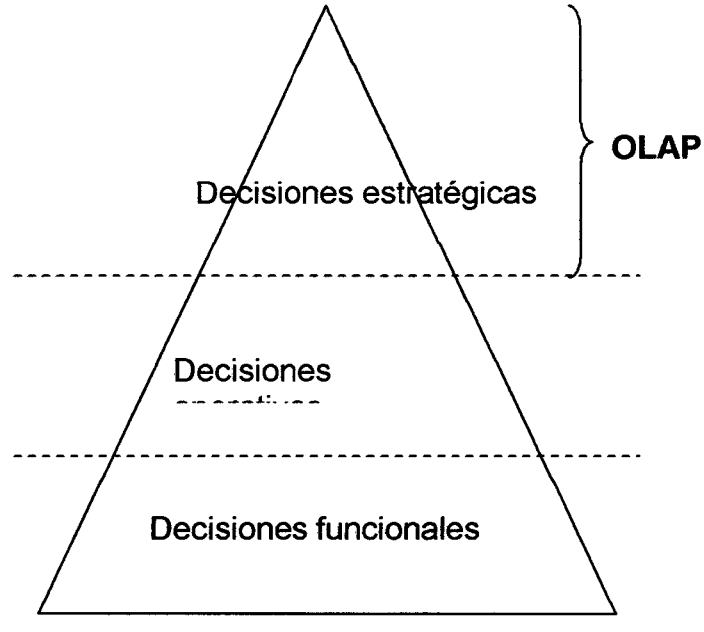


Figura 2.21: Pirámide de decisiones empresarial

Fuente: DE MIGUEL CASTAÑO ADORACIÓN

2.2.3 ESTRUCTURA DEL MODELO MULTIDIMENSIONAL

No existe un modelo estándar, existen una serie de características comunes a las diferentes propuestas, que se presentará posteriormente, una formalización de un modelo multidimensional.

En un modelo de datos multidimensional hay un conjunto de hechos o medidas que son el objeto de nuestro análisis: ventas, presupuestos, inventario, etc. La mayoría de los hechos más útiles son numéricos, continuamente valuados y aditivos. La razón de que sean de dichas características es la siguiente: las consultas a la tabla de hechos necesitarían a su vez consultas de cientos, miles o incluso millones de registros para construir el conjunto respuesta. Esta gran cantidad de registros será resumida en unas cuantas docenas de registros y serían necesarias operaciones sobre los datos: agregados.

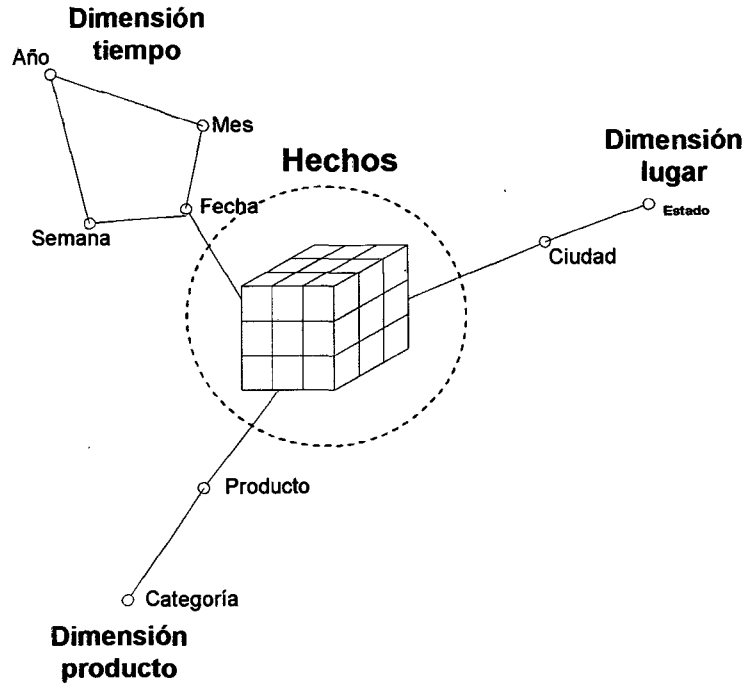


Figura 2.22: Ejemplo de estructura multidimensional

Fuente: Elaboración propia

Cada medida depende de un conjunto de Dimensiones, que proporcionan el contexto. En un espacio multidimensional para designar un punto son necesarias todas sus coordenadas, dependiendo del número de dimensiones que tenga el espacio. Pues esto es lo que ocurre en este modelo, ya que al indicar un valor para cada dimensión se situará un hecho concreto dentro del espacio formado. En la Figura 2.22 se recoge un ejemplo.

En esta Figura vemos que el modelo tiene tres dimensiones, formando un cubo. A los modelos con *n* dimensiones se les denomina hipercubos, y por esta razón a todos los modelos construidos se les denomina cubos. En la Figura tenemos las dimensiones: Producto, Tiempo y Lugar. Por tanto, si se estuviesen estudiando las Ventas como medida, para cada combinación de los valores de dichas dimensiones puede ser que existan valores para dicha medida. No para todas las coordenadas existirán hechos, dado que habría combinaciones que pueden no presentarse nunca.

Por tanto, como este factor va a estar presente en muchos de los cubos que se construyan en cualquier entorno, será necesario un buen tratamiento de lo que se denomina matrices dispersas, es decir, matrices n-dimensionales que tengan muchos huecos en su estructura.

Sobre las dimensiones se pueden definir jerarquías. Estas lo que van a permitir es acceder a los datos a diferentes niveles de detalle, es decir, ver los datos a diferentes granularidades. En el ejemplo de Figura 2.22, hemos

definido una jerarquía con dos niveles en la dimensión Producto: producto, propiamente dicho, y categoría; de tal forma que podríamos acceder a los hechos o a nivel de producto o agrupándolos según las categorías a las que pertenecen.

OPERACIONES

Es precisamente esta estructura jerárquica en cada dimensión la que servirá de base para definir todas las operaciones que se puedan realizar sobre el modelo.

Movimiento en **la estructura jerárquica:** Para moverse en los diferentes niveles de jerarquía de una dimensión se definen las siguientes operaciones:

- Roll-up: significaría subir en la jerarquía, esto es, aumentar el tamaño del grano al que están definidos los hechos. Al aplicar esta operación necesitamos resumir información para adaptar el nivel de detalle de los hechos. En este proceso de resumen utilizaremos operadores de agregación.

- Drill-down: Esta operación es justo la contraria a la anterior. Ahora lo que pretendemos es reducir el nivel de grano, obteniendo un mayor nivel de detalle. Esto se traduce en cambiar el nivel de definición de los hechos a niveles inferiores de las jerarquías.

Un ejemplo de la traducción de estas operaciones sobre un esquema multidimensional es recogida en Figura 2.23.

Operaciones de selección y proyección: Muchos análisis pueden que no dependan de todos los valores (selección) o de todas las dimensiones (proyección) que consideramos. Las operaciones que se encargan de esta funcionalidad son slice y dice:

- Slice: esta operación consiste en reducir la dimensionalidad del esquema eliminando alguna dimensión. Aplicar esta operación implica pérdida de detalle en los hechos (aumentamos la granularidad) por lo que tendremos que utilizar operadores de agregación para la obtención de los nuevos (Figura 2.24).

- Dice: es este caso, lo que hacemos es restringir los valores que consideramos en las dimensiones según alguna condición. No modificamos la estructura del datacubo en cuanto a las dimensiones y niveles de éstas, sino restringiendo los valores que consideramos. En este caso, no modificamos el nivel de detalle de los hechos pero sí su número, dado que aquellos que tuvieran como coordenadas valores que no consideramos debemos de eliminarlos.

Operación de pivotaje: Esta operación lo que persigue es la modificación de la definición de la estructura de los datacubos. Lo que implica es un cambio en el orden de las dimensiones. Un ejemplo viene recogido en la Figura 2.25.

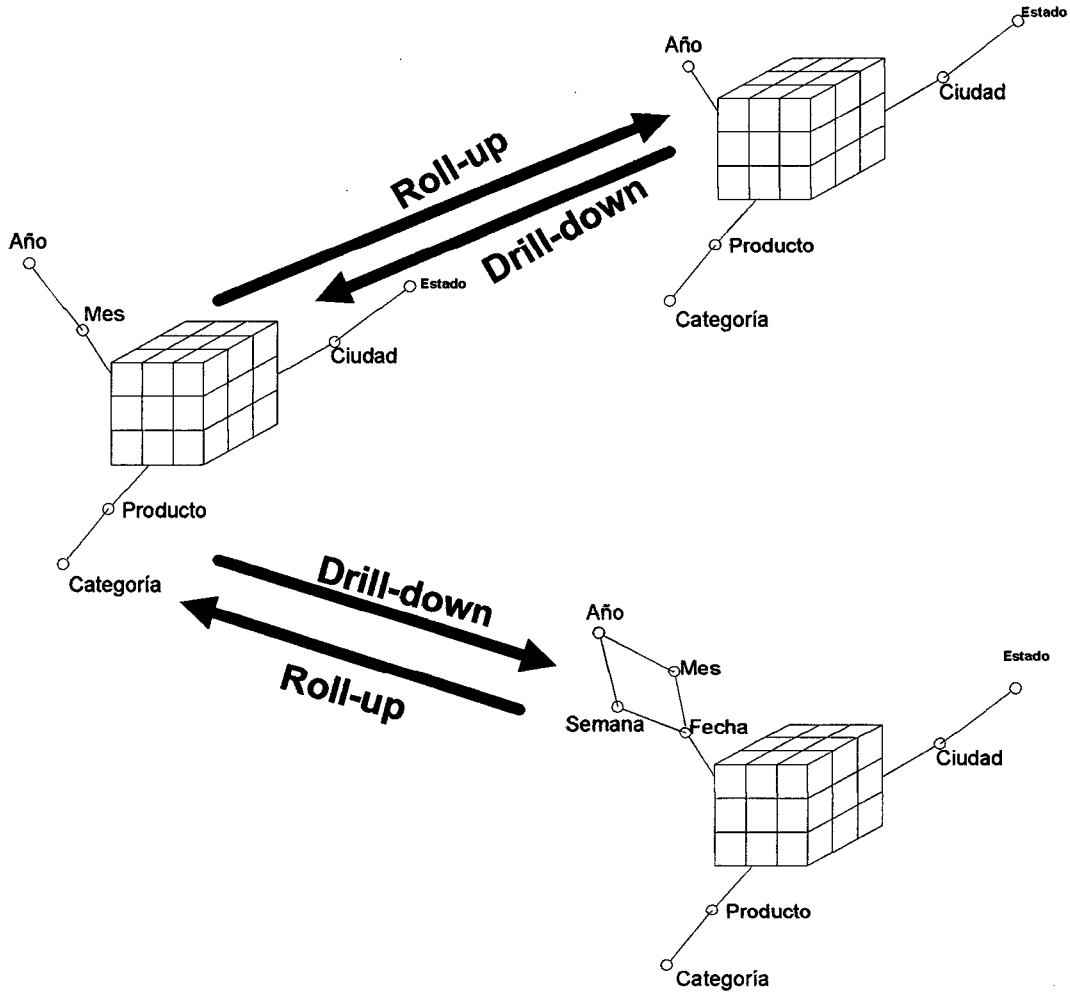


Figura 2.23: Ejemplo de aplicación de las operaciones roll-up y drill-down sobre el datacubo de la Figura 2.22.

Fuente: Elaboración propia

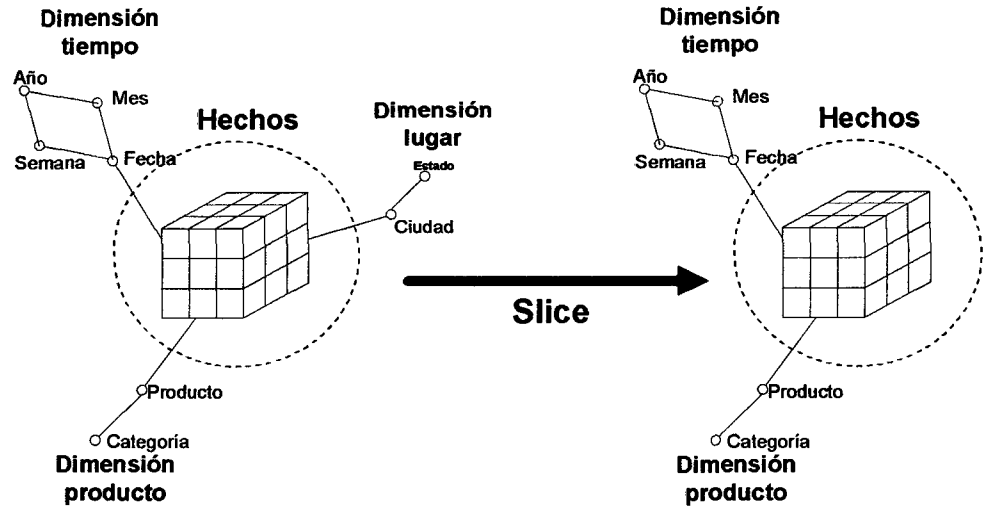


Figura 2.24: Ejemplo de aplicación de slice sobre el datacubo de Fig. 2.22

Fuente: Elaboración propia

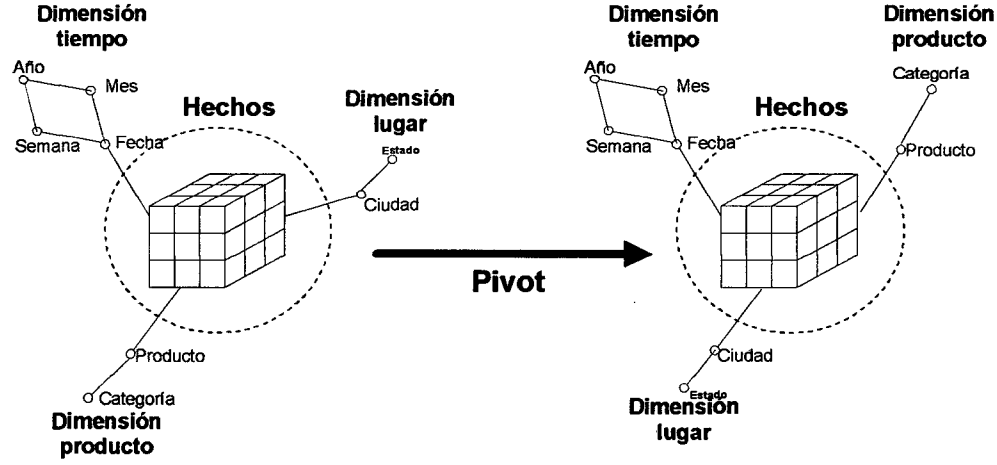


Figura 2.25: Ejemplo de aplicación de pivot sobre el datacubo de Fig. 2.22

Fuente: Elaboración propia

MODELOS DE IMPLEMENTACIÓN MULTIDIMENSIONAL

Existen diversas formas de representar el modelo multidimensional en un esquema físico. Se estructuran en dos tipos principalmente: esquemas ROLAP y MOLAP.

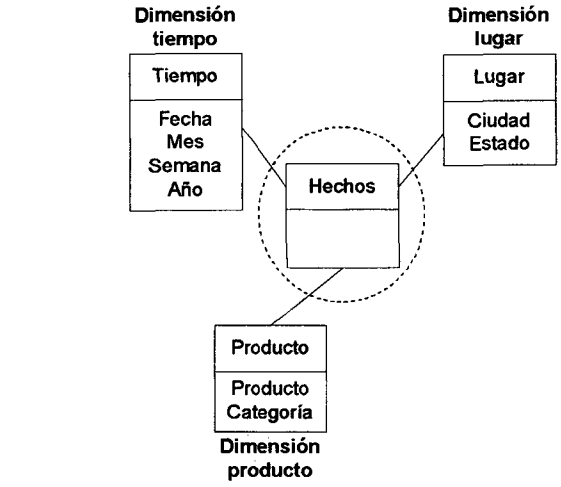
Servidores ROLAP: El modelo de datos multidimensional se puede implementar en servidores relacionales, dando lugar a lo que se conoce como Relational- OLAP servers (ROLAP), representando en ellos tanto el modelo como sus operaciones transformándolo todo en estructuras relacionales (tablas y relaciones). Veamos cuáles son los diferentes esquemas de bases de datos relacionales que reflejan esas visiones multidimensionales.

La mayoría de los Data Warehouses usan un esquema en *estrella* para representar los datos multidimensionales (Figura 2.26). La base de datos que se implementa consiste en una tabla simple de hechos y una tabla para cada dimensión. Cada tupla de la tabla de hechos consta de un puntero a cada dimensión, lo que proporciona las

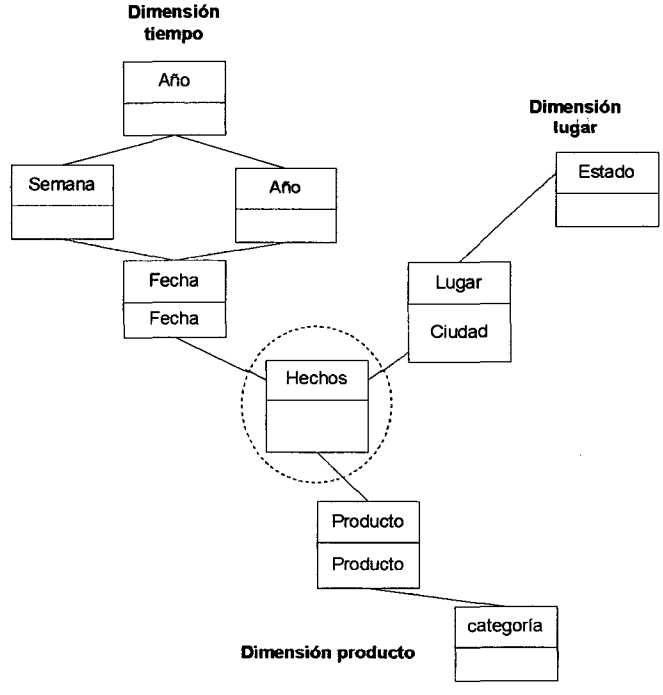
coordenadas multidimensionales y los datos de las medidas que el modelo considere (cantidad, costos, etc).

Los esquemas en estrella no proporcionan explícitamente soporte para las jerarquías de atributos. Los esquemas *en copo de nieve* tal como se muestra en la misma Figura son un refinamiento de los esquemas en estrella que proporcionan dicho soporte y además suponen que las tablas están normalizadas. El único inconveniente es que suponen el manejar mayor número de tablas (en las consultas se han de realizar un número mayor de *joins* entre tablas). El esquema en estrella puede generalizarse con la inclusión de distintas tablas de hechos que comparten tablas de dimensiones, es lo que se denomina *constelaciones de hechos.*

Adicionalmente a las tablas de hechos y dimensiones, los sistemas pueden almacenar tablas resumen que almacenan datos preagregados. En los casos más simples, los datos preagregados corresponden a la agregación de una tabla de hechos sobre una o más de sus dimensiones.



a)



b)

Figura 2.26: Implementación del esquema de la Figura 2.22 utilizando a) modelo en estrella b) modelo en copo de nieve.

Fuente: Elaboración propia

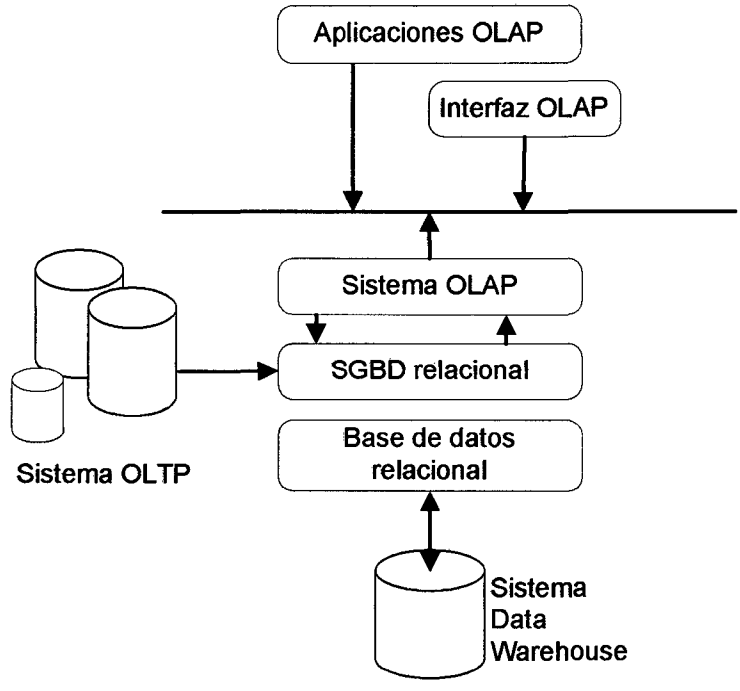


Figura 2.27: Arquitectura de un data warehouse para sistemas ROLAP. Fuente: Wu, Ch., Buchmann, P.

Según Wu y Buchman, en estos sistema la arquitectura del data warehouse sería la recogida en la Figura 2.27. Como puede verse, tanto la capa de almacenamiento como la de manejo de datos utilizan un enfoque relacional. La capa del *sistema OLAP relacional* proporciona un acceso multidimensional a los datos subyacentes, proporcionando operaciones multidimensionales a los usuarios.

Servidores MOLAP: En contraste con ROLAP, el OLAP multidimensional (MOLAP) es un modelo de datos de propósito especial y las operaciones se hacen directamente.

En lugar de almacenar la información como registros y los registros como tablas, las bases de datos multidimensionales almacenan los datos como matrices. Las bases de datos multidimensionales son capaces de proporcionar un rendimiento de consulta muy alto lo que se consigue anticipando y restringiendo la forma en que se accede a los datos. En general, la información en una base de datos multidimensional es de una granularidad más gruesa que la que se considera en una base de datos relacional estándar, y por tanto los índices asociados son menores y pueden residir en memoria. Una vez que el índice se analiza, se capturan algunas páginas de las bases de datos, algunas herramientas están diseñadas para trabajar con estas páginas en memoria compartida lo que aumenta aún más el rendimiento. Otro aspecto muy interesante de las bases de datos multidimensionales es

que la información está físicamente almacenada en arrays lo que significa que los valores de las celdillas se pueden actualizar sin que afecte a los índices.

Los servidores MOLAP existentes no tienen muchos elementos en común. A diferencia del modelo relacional no existe un acuerdo sobre lo que debe ser un modelo de datos MOLAP y tampoco hay un método estándar de acceso tales como lenguajes de consultas. En este último sentido, el OLAP Council propuso el MDAPI (API para el modelo multidimensional).

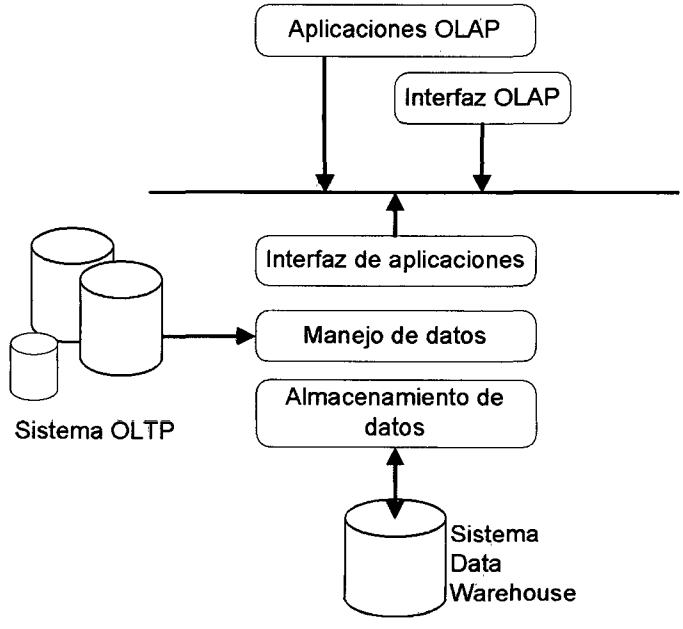


Figura 2.28: Arquitectura de un data warehouse para sistemas MOLAP.

Fuente: Wu, Ch., Buchmann, P.

Servidores HOLAP: Los enfoques anteriores fueron los primeros en surgir. Posteriormente lo que se pensó fue en unir ambos enfoques de tal manera que se puedan aprovechar las ventajas de ambos enfoques. Los sistemas que siguen esta filosofía se denominan OLAP híbrido o HOLAP (Hybrid OLAP).

Esta hibridación entre los enfoques se puede ver desde diferentes puntos de vista.

Definición 2.3 *un sistema HOLAP soporta (e integra) el* almacenamiento *de los datos multidimensional y relacional de una manera equivalente con el objetivo de beneficiarse de las correspondientes características y técnicas de optimización.*

Los sistemas deben de cumplir unas características para considerarse realmente HOLAP:

- Transparencia respecto a los sistemas MOLAP y ROLAP subyacentes. Los sistemas HOLAP integran subsistemas MOLAP y ROLAP. Estos sistemas deben de esconder las peculiaridades de cada uno, siendo invisible para el usuario el tipo de almacenamiento que se utilice (relacional o multidimensional).

- Modelos de datos comunes y esquema multidimensional global. El sistema debe de proporcionar un modelo de datos y un lenguaje de consulta común.

Además, establecen otras características en el caso de que estos sistemas sean distribuidos.

Los sistemas comerciales han evolucionado siguiendo la filosofía HOLAP. Algunos ejemplos son SQL Server de Microsoft, Oracle Express, SAS OLAP Server y DB2 OLAP Server de IBM.

Servidores O3LAP

Una cuarta vía de implementación de estos sistemas es la utilización de sistemas de bases de datos basados en objetos o incluso objetos persistentes proporcionados por lenguajes que siguen este paradigma de programación. Estos sistemas son conocidos bajo las siglas O3LAP: Object-Oriented OLAP.

Según Buzydlowski este enfoque presenta ventajas tanto a nivel de implementación (nivel físico) y conceptual. En el nivel físico las ventajas que presentan serían.

- Existen estándares para las bases de datos orientadas a objetos, como la propuesta por ODMG (Object-Oriented Database Management Group).

- En el campo de las bases de datos orientadas a objetos se han realizado numerosas investigaciones, de forma que algunos problemas como la autorización

de usuarios o la actualización de la base de datos mediante transacciones han sido estudiados.

- Algunas características fácilmente implementables por BDOO pueden ser muy útiles a la hora de desarrollar data warehouses. También la existencia de estándares de objetos distribuidos puede ayudar a su implementación.

En el modelado conceptual, principalmente se refieren a la posibilidad que dan los objetos para trabajar con diferentes tipos de datos dentro de un modelo multidimensional. Dado que los objetos encapsulan tantos datos, visualización y métodos de manipulación, se podría incluir información como imágenes u otros tipos de información estructurada de forma simple.

2.3 MODELOS MULTIDIMENSIONALES CLÁSICOS

En esta sección se hace una presentación muy breve de los modelos multidimensionales más importantes que se han encontrado en la bibliografía

Modelo de Li y Wang

Para Li y Wang, un esquema de cubo es un conjunto de parejas formadas por un nombre y un conjunto de atributos. Estas parejas constituyen las dimensiones del cubo. La instancia de un cubo, está dada por una pareja formada por un conjunto de relaciones, una para cada dimensión, y una función. Esta función, mapea coordenadas (tuplas) formadas por las tuplas de cada dimensión en un determinado conjunto de valores escalares.

Resumiendo la estructura, un cubo en este modelo es un conjunto de relaciones, una para cada dimensión, y una función que mapea coordenadas formadas por tuplas de esas relaciones en una única medida. La noción de relación que utiliza es la misma que en el

Modelo Relacional. El lenguaje de manipulación que propone es una extensión del Álgebra Relacional con operadores de ordenamiento y agrupamiento. La especificación resultante es bastante intrincada, sobre todo para obtener algo muy similar a SQL desde el punto de vista del lenguaje y al Modelo Estrella desde el punto de vista de la estructura. Desde el punto de vista multidimensional, el modelo no soporta dimensionalidad genérica. Un aspecto diferente del modelo, es que en el esquema no hay ninguna referencia la medida. Sólo se puede saber alguna información de la misma a partir de la instancia del cubo.

Los autores proponen una formalización de un modelo multidimensional como la relación de múltiples dimensiones con los hechos. Las jerarquías en las dimensiones, como en el modelo anterior, no aparecen de forma explicita sino mediante agrupaciones de valores por otros.

Un DataCubo es considerado instanciación de un esquema n-dimensional de cubo.

Definición 2.4 Sea n un entero positivo y V un conjunto de valores escalares. Un esquema *n-dimensional de cubo* es un conjunto:

$\{(D_1,R_1), ..., (D_n,R_n)\}$, donde:

$D_1, ...,D_n$ son nombre distintos de dimensiones y

$R_1, ...,R_n$ *son conjuntos de nombres de* atributos.

Un cubo n-dimensional sobre el esquema $\{(D_1,R_1), ..., (D_n,R_n)\}$ es un par (F, μ), donde:

- $F = \{(D_1,R_1), ..., (D_n,R_n)\}$ con r_i siendo una relación en R_i para cada $1 \leq i \leq n$,
- y μ es una aplicación de $\{(D_1, t_1), ..., (D_n, t_n) | \forall\ 1 \leq i \leq n : t_i \in r_i\}$ a V.

Para completar el modelo, presentan una extensión del álgebra relacional, denominada grouping algebra. En ella, se incluyen operaciones que requieran ordenaciones (order-oriented operations), es decir, divisiones del dominio en intervalos según un orden dado, y agregaciones (aggregation operations).

Modelo de R. Kimball

En este modelo se trata más de un modelo de implementación que de un modelo conceptual. Kimball propone la construcción de un modelo multidimensional utilizando para ello un esquema relacional. En este, los hechos corresponderían a una tabla y cada dimensión a otra con una relación de llave externa con los hechos. Este es el modelo de implementación *en estrella* que hemos presentado anteriormente al hablar de los servidores ROLAP.

En este modelo, el movimiento en las jerarquías se realiza utilizando operaciones GROUP BY relacional con los operadores de agregación habituales (suma, máximo, mínimo, etc.).

Modelo Gyssens y Lakshmanan.

El modelo que se propone en es prácticamente el mismo que el anterior desde el punto de vista estructural. Sin embargo, presenta algunas diferencias importantes en su especificación y propiedades.

La estructura básica en este modelo se llama tabla, aunque en realidad es un cubo. El esquema de una tabla, se define como una terna <D, R, Pair> donde D es un conjunto $\{d_1,...,d_n\}$ de nombres de dimensión, R es un conjunto de atributos y Pair es una función que mapea nombres de dimensión en subconjuntos de R. Esta función garantiza que los conjuntos de atributos que asocia a dimensiones diferentes son disjuntos dos a dos, reflejando de esta forma la noción de que las dimensiones de un espacio son ortogonales entre sí.

El conjunto M de las medidas, está conformado por los atributos de R que no fueron asignados a ninguna dimensión por la función Pair.

Resumiendo, es otra implementación del Modelo Estrella. Sin embargo, su especificación es más clara que la de Li y Wang. El lenguaje de consulta que proponen es una extensión del Álgebra Relacional con operadores de agrupamiento y funciones agregadas. Además, provee funciones para intercambiar atributos

de dimensiones y medidas, con lo que provee el soporte básico para la dimensionalidad genérica.

Modelo Agrawal, Gupta y Sarawagi.

Fue uno de las primeras formalizaciones que se ha hecho de un modelo multidimensional. Se propone un modelo con la siguiente funcionalidad:

- Tratamiento simétrico no sólo de todas las dimensiones sino también de las medidas.
- Soporte para múltiples jerarquías en cada dimensión.
- Soporte para el cálculo de agregados ad-hoc, es decir, no se limitan los agregados a aquellos predefinidos.
- Soporte para un modelo de consulta.

Para dar respuesta a esta funcionalidad, proponen un modelo de datos donde un DataCubo se define como sigue.

Definición 2.5 *Un DataCubo C es una estructura con la siguiente estructura:*

- *k dimensiones, y para cada dimensión un nombre* D_i*, un dominio* dom_i *desde el cual se* toman los valores.
- Elementos definidos como una aplicación E(C) desde $dom_i \times ... \times dom_k$ a un n-tupla, 0 ó 1. Con esto, $E(C)(d_1, ..., d_k)$ se refiere al elemento en la posición $d_1, ..., d_k$ del DataCubo C. El valor 0 representa que esa posición no tiene definidos valores, 1 si la combinación existe pero no sabemos nada mas, y la tupla si tenemos información adicional.
- Una n-tupla con los nombres describiendo cada uno de los elementos de las n-tuplas que define el DataCubo.

Sobre esta estructura definen, aparte de las operaciones habituales, introducir una dimensión como hecho (*push*), sacar un hecho para ser tratado como una dimensión (*pull*), y la combinación de datacubos (*join*). Este conjunto de operaciones es mínimo y cerrado (el resultado es siempre un datacubo). El modelo no define de forma explícita las jerarquías en la dimensiones, sino mediante operaciones que agrupan los valores en las dimensiones.

Este modelo es realmente multidimensional, o sea, sus estructuras no se describen directamente sobre las estructuras del Modelo Relacional. Sin embargo, las estructuras del modelo si se pueden mapear sobre las del Modelo Relacional.

La estructura básica es un *hipercubo* compuesto por dos elementos: un conjunto de dimensiones y una función que mapea coordenadas formadas por valores de cada una de las dimensiones en tuplas o booleanos. Una dimensión es un nombre con un dominio asociado.

Como ventajas, se puede mencionar que las operaciones que sugieren están más cerca de las operaciones multidimensionales que las propuestas por los modelos anteriores. Además, la idea de "cubo booleano" que introduce, es la que toman prácticamente todos los modelos restantes para implementar la dimensionalidad genérica. La principal desventaja, se plantea desde el punto de vista del modelado de datos, ya que no diferencia entre esquema e instancia de las estructuras.

Modelo de Gray *et al.*

No presenta como tal un modelo multidimensional sino que extiende las operaciones de agregación tipo *group by* de los sistemas relacionales para el caso de datos multidimensionales. Las operaciones que se proponen son:

ROLLUP: se trata de una extensión de la operación *group by* en el que se realiza la agregación según *n* dimensiones ordenadas aplicando el esquema siguiente:

GROUP BY n-dimensiones ∪ GROUP BY n-1 primeras dimensiones ∪ GROUP BY n-2 primeras dimensiones ∪ ... ∪ GROUP BY 1 dimensión

CUBE: se trata de aplicar la operación anterior (rollup) sobre todas las dimensiones, es decir, el resultado es la unión de aplicar rollup sobre un conjunto de dimensiones. El resultado que se obtiene es la agregación de cada posible subconjunto de las dimensiones que consideremos.

Modelo de Cabibbo y Torlone

En este modelo se presenta un modelo multidimensional (llamado *MD*) con dos partes muy diferenciadas:

- Dimensiones: que representan categorías lingüísticas que corresponden a las diferentes aproximaciones a los datos.
- f-tables: que representan a los hechos y se relacionan con las dimensiones. Se definen como funciones que relacionan coordenadas simbólicas con las medidas.

Definición 2.6 *Una dimensión MD está formada por:*

- un conjunto finito de niveles L' $\subseteq$ L, donde L es un conjunto de nombres para los niveles;
- un orden parcial $\leq$ sobre los niveles en L', donde si $l_1 \leq l_2$ decimos que l_1 es agrupado por l_2;
- una familia *de funciones, incluyendo la función* $R-UP_{l_1}^{l_2}$ *desde* $DOM(l_1)$ *a* $DOM(l_2)$ *que para cada par de niveles* $l_1 \leq l_2$, *donde si* $R-UP_{l_1}^{l_2}(o_1) = o_2$ *significa que* o_1 *es agrupado por* o_2.

En este modelo, como puede verse, se define de forma explícita las jerarquías en las dimensiones, estableciendo una ordenación entre

los niveles mediante un orden parcial y funciones $R-UP_{l_1}^{l_2}$ que establecen la relación entre los elementos de estos niveles.

Definición 2.7 *Un esquema MD está formado por:*

- *un conjunto finito D de dimensiones;*
- *un conjunto F de esquemas f-tablas de la forma* $f[A_1 : l_1 < d_1 >, ..., A_n : l_n < d_n >] : l_0 < d_0 >$, *donde f es un nombre, cada* A_i $(1 \leq i \leq n)$ *es un nombre distinto llamado* atributo, *y cada* l_i $(0 \leq i \leq n)$ *es un nivel de la dimensión* d_i;

El modelo presenta dos estructuras básicas: Dimensiones y F-Tablas.

Para definir las instancias, se define primero la noción de coordenada simbólica. Una coordenada simbólica sobre una f-tabla es una función que mapea cada nombre de atributo en un valor del dominio del nivel asociado al atributo. De esta forma, una coordenada es una tupla.

Una instancia de f-tabla, es una función que mapea coordenadas simbólicas en valores del dominio de la medida.

El modelo presenta un lenguaje de consultas basado en un cálculo de conjuntos.

Resumiendo, las estructuras que presenta el modelo son multidimensionales y están bien definidas.

Los caminos posibles para realizar roll-up, están explícitamente en las estructuras de las dimensiones. Esto permitiría al analista mejorar la precisión de sus especificaciones.
El cálculo que se presenta, permite manipular la dimensionalidad genérica de una forma muy natural, a través del uso de f-tablas con recorrido booleano. S in embargo, el modelo adolece de un problema:

La mayoría de las veces las funciones de roll-up son extensionales, es decir, están dadas por una tabla que indica como se relaciona cada elemento de un nivel con cada elemento del nivel superior. Esto hace pensar que las funciones de roll-up deberían estar en la instancia y no en el esquema de las dimensiones.

Modelo de Datta y Thomas

El modelo propuesto hace una clara diferenciación entre los hechos y las dimensiones.

Definición 2.8 Un DataCubo es una 6-tupla, < D,M,A,f,V,g > donde los cuatro componentes son:

- Un conjunto de n dimensiones D = $\{d_1, ..., d_n\}$, donde cada d_i es un nombre dimensión extraído del dominio de dimensiones (dom dim(i));
- un conjunto de k medidas M = $\{m_1, ...,m_k\}$ donde cada m_i es una medida extraída del dominio de **medidas** (dom measures(i));
- El conjunto de dimensiones y medidas es disjunto ($D \cap M = \varnothing$);
- un conjunto de t atributos A = $\{a_1, ..., a_t\}$ donde cada a_i es un atributo extraído del dominio de atributos (dom attr(i));
- una aplicación uno-a-muchos f : D → A, tal que los conjuntos de atributos correspondientes a dos dimensiones son disjuntos ($\forall$i, j i ≠ j, $f(d_i) \cap f(d_j) = \varnothing$);
- *V un conjunto de tuplas de tamaño k tal que* $v_i =< \mu_1, ..., \mu_k >$, *donde cada* μ_i *es una instancia de la medida i-ésima;*

- *g es una aplicación g : domdim(i) × ... × domdim(n) → V , que asocia a cada conjunto de coordenadas los hechos relacionados.*

Sobre esta estructura, formaliza las operaciones habituales sobre el modelo, contemplando algunas más:

- *Producto cartesiano*: un operador binario que construye un nuevo datacubo *mediante la combinación de otros dos. Un caso particular del producto cartesiano es el operador JOIN que se usa cuando los datacubos tienen una o más dimensiones en común.*

- *Diferencia: este operador devuelve un datacubo cuyo contenido son las diferencias entre otros* dos compatibles (elimina la parte común a ambos).

Las jerarquías en las dimensiones no aparecen de forma explícita, por lo que la agrupación de valores se realiza mediante una

operación que agrupa valores (*partición*). Sobre el modelo, definen un álgebra de consulta con estas operaciones.

Modelo de Golfarelli, Rizzi y Maio.

El modelo es presentado como una notación para representar estructuras multidimensionales que se obtienen de la especificación Entidad- Relación de la base operativa. La estructura básica es el *esquema de hechos.*

La notación tiene características interesantes:

- *Permite la especificación de cuáles son las funciones de agregación mediante las que se pueden agregar las medidas. Son pocos los modelos que permiten esto. Esto se puede ver como alguna forma de restricción de integridad.*
- *Permite visualizar* gráficamente las jerarquías de las dimensiones.

Sin embargo, el modelo no soporta dimensionalidad genérica y en esa versión, no presentaba ninguna formalización del mismo.

Además, no permite visualizar los posibles caminos para la realización de drill-across.

En los aspectos metodológicos que se discuten, se destaca la noción de carga de trabajo y una solución para la visualización de los caminos de drill-across que tiene el aspecto de una posible metodología de integración de esquemas multidimensionales.
Por otro lado, el modelo sólo permite un número limitado de restricciones basadas en la notación gráfica. Entre las más relevantes está la no aditividad con respecto a una dimensión.

Modelo System 42.

Se presenta un modelo conceptual muy particular. Es una extensión directa del Modelo Entidad-Relación, aplicada a las nociones multidimensionales.
Este modelo se construye especializando la noción de entidad y relación. Un nivel, es un tipo de entidad. El roll-up es un tipo de relación binaria y el cubo (Fact Name) es una relación N-aria, en donde las medidas se presentan como atributos de la relación.

El modelo presenta como principal ventaja el hecho de estar basado en otro modelo ampliamente difundido como el Entidad-Relación. Gracias a esto debería ser posible extender conceptos de E/R hacia este modelo como las restricciones no estructurales o las técnicas de reutilización de esquemas. Sin embargo, por este mismo motivo, no presenta un mecanismo de refinamiento o representación modular ni soporte para dimensionalidad genérica.

Modelo de Lehner.

Se presenta el "Modelo Multidimensional Anidado". Las estructuras básicas del modelo son los Objetos Multidimensionales (MO) que son en realidad, cubos.

Para definir estos cubos, se definen las dimensiones clasificando sus atributos en:

- **Primarios**. Hay un valor para estos atributos en cada hoja de la jerarquía de la dimensión. Permiten la identificación de los objetos de la misma y representan el nivel más desagregado de la jerarquía.

- **De Clasificación**. Para estos atributos, hay un valor en cada nodo interno o raíz de la jerarquía de la dimensión. Permiten la construcción de la jerarquía. Representan los niveles de la dimensión.
- **Descriptivos o Dimensionales**. Describen características de los elementos de la dimensión. Pueden estar asociados a varios atributos de clasificación, pero no tienen por qué estar asociados a todos. Esto permite la manipulación de valores según un contexto definido por los atributos de clasificación. En general, representan medidas.

La jerarquía de clasificación tiene en cada nodo interno y en la raíz un valor para un atributo de clasificación y en las hojas, un valor para un atributo primario.

Un cubo (MO), se ve como un conjunto de **Objetos Multidimensionales Primarios (PMO)** que se pueden ver como cada celda de la matriz. Este conjunto de objetos se define mediante condiciones sobre los atributos de clasificación de cada dimensión que participa en el cubo. Con esto se define un PMO para cada cruzamiento de valores de las dimensiones.

A cada PMO, se le asocia un **Objeto Multidimensional Secundario (SMO)**. Cada SMO consta de un conjunto de atributos descriptivos, a partir de los cuales se pueden construir las medidas.

El modelo es bastante complejo. Sin embargo, presenta características muy particulares:

Permite la manipulación de medidas contextuales, es decir, que algunos elementos de la dimensión las tienen y otros no. De esta forma, los elementos de la dimensión productos que tienen grupo "video" pueden tener una medida "cantidad de canales", que no tiene sentido sobre los elementos de la dimensión producto que tienen grupo "equipo de audio".

En la definición del conjunto de PMO's, se puede especificar el tipo de resumen que se puede realizar con los SMO. Si el tipo es aditivo, se puede aplicar las operaciones SUM, AVG, MIN, MAX o COUNT, mientras que si es promedio, se pueden aplicar las operaciones AVG, MIN y MAX y si es constante no se puede aplicar ninguna operación.

Modelo DWQ.

El modelo DWCDM (Data Warehouse Conceptual Data Model) presenta dos lenguajes: uno gráfico y otro formal basado en Lógicas Descriptivas (DL).

El modelo gráfico, está basado en el Modelo Entidad-Relación. En este modelo, el mecanismo de representación de la realidad es drásticamente diferente al que normalmente se utiliza al construir un esquema Entidad-Relación.

El lenguaje gráfico se enfoca a la descripción de agregaciones ya sea sobre cubos o sobre elementos de las dimensiones.

Esta representación gráfica, luego es traducida a un lenguaje basado en Lógica Descriptivas. Las Lógica Descriptivas manejan dos elementos básicos: conceptos y roles. Un concepto representa un conjunto de valores y un rol una relación binaria. Los operadores básicos son similares a los de un álgebra de conjuntos y presenta formas restringidas de cuantificación.

El punto más interesante de este modelo es que la satisfactibilidad de la especificación en DL que se obtiene al traducir el diagrama es decidible.

El lenguaje formal, permite expresar algunas restricciones de cardinalidad y posiblemente, permita otras. Sin embargo, de los artículos disponibles sobre el modelo, no surge que permita la expresión de condiciones generales como que determinado valor tiene que ser menor que otro.

Otro inconveniente es que no se ha encontrado en la bibliografía consultada, una descripción suficientemente detallada del lenguaje gráfico.

2.4 SOPORTE PARA BBDD MULTIDIMENSIONALES EN ORACLE.

2.4.1 ALMACENES DE DATOS EN ORACLE

Vamos describir las características de Oracle9i Enterprise Edition que están especialmente diseñadas para mejorar las prestaciones y la capacidad de gestión de los almacenes de datos.

ORACLE9i

Oracle9i Enterprise Edition es uno de los SGBD relacionales más utilizadas para almacenes de datos. Oracle ha conseguido semejante éxito centrándose en los requisitos básicos fundamentales de los almacenes de datos: prestaciones, estabilidad y capacidad de gestión. Los almacenes de datos albergan mayores volúmenes de datos, dan soporte a más usuarios y requieren unas mayores prestaciones, por lo que estos requisitos fundamentales constituyen aspectos clave para la adecuada implementación de un almacén de datos corporativo. Sin embargo, Oracle va más allá de estos requisitos clave,

proporcionando la primera auténtica "plataforma de almacén de datos". Las aplicaciones de almacén de datos requieren técnicas de procesamiento especializadas que proporcionen soporte para las consultas complejas ad hoc que analicen grandes cantidades de datos. Para satisfacer estos requisitos especiales, Oracle ofrece diversas técnicas de procesamiento de consultas, mecanismos sofisticados de optimización de consultas para seleccionar las rutas de acceso a los datos más eficientes y una arquitectura escalable que aprovecha al máximo las conFiguraciones de hardware paralelo. Las características de Oracle dirigida a soportar en concreto las aplicaciones de almacén de datos incluyen:

- Gestión de resúmenes
- Funciones analíticas
- Índices de mapa de bits
- Métodos de combinación avanzados
- Optimizador SQL sofisticado
- Gestión de recursos.

2.4.2 DISEÑO DE ALMACENES DE DATOS CON ORACLE

Vamos a describir Oracle Warehouse Builder (OWB) como uno de los componentes clave de la solución para almacenes de datos de Oracle. Este componente permite el diseño e implantación de almacenes de datos, mercados de datos y aplicaciones de inteligencia empresarial. OWB es una herramienta de diseño y también una herramienta de extracción, transformación y carga (ETL; extraction, transformation and loading). Un aspecto importante de OWB desde la perspectiva de las empresas es que permite la integración de los entornos tradicionales de almacenes de datos con los nuevos entornos de e-Business. Vamos a proporcionar primero una panorámica de los componentes de OWB y de sus tecnologías subyacentes y luego veremos cómo puede aplicarse OWB a las tareas típicas de diseño de un almacén de datos.

A) Componentes de Oracle Warehouse Builder

OWB proporciona los siguientes componentes funcionales principales:

- Un **repositorio** compuesto de un conjunto de tablas en una base de datos Oracle al que se accede a través de un nivel de acceso basado en Java.
- Una **interfaz gráfica de usuario** (GUI) que permite acceder al repositorio. La interfaz incluye editores gráficos y un amplio conjunto de asistentes. Esta interfaz está escrita en Java, lo que hace que sea portable.
- Un **generador de código**, también escrito en Java, genera un código que permite la implantación de almacenes de datos.
- **Integradores,** que son componentes dedicados a extraer datos de un tipo concreto de fuente.
- Una **interfaz abierta** que permite a los desarrolladores ampliar las capacidades de extracción de OWB y aprovecharse de los beneficios que proporciona la arquitectura OWB.
- **Runtime** (entorno de ejecución), que es un conjunto de tablas, secuencias, paquetes y disparadores que se instalan en el esquema de destino.

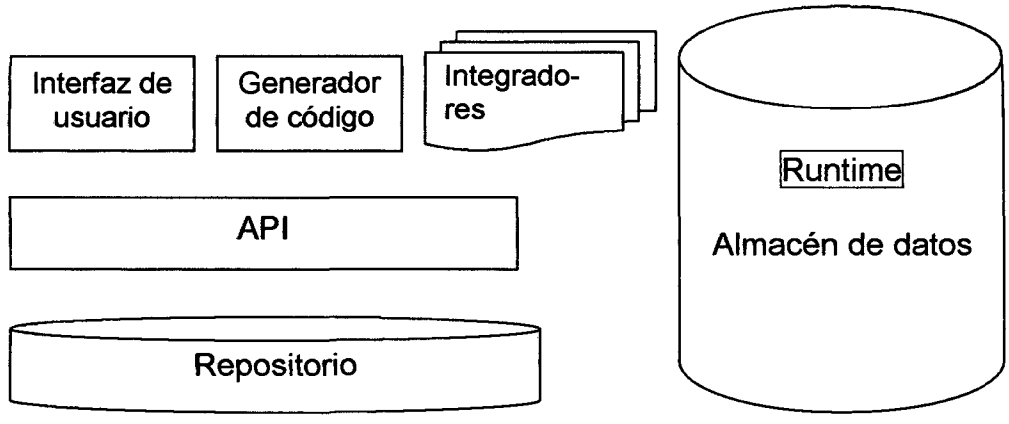


Figura 2.29 Arquitectura de Oracle Warehouse Builder.

Fuente: Elaboración propia

La arquitectura de Oracle Warehouse Builder se muestra en la Figura 2.29 Oracle Warehouse Builder es uno de los componentes clave de los almacenes de datos Oracle. Los otros productos con los que OWB debe integrar dentro del almacén de datos incluyen:

- Oracle: el motor de OWB (ya que no hay servidor externo);
- Oracle Enterprise Manager: para planificación;
- Oracle Workflow: para gestión de dependencias;
- Oracle Pure•Extract: para acceso a mainframes MVS;
- Oracle Pure•Integrate: para control de calidad de los datos;
- Oracle Gateways: para acceso a datos almacenados en sistemas relacionales y tipos mainframe.

B) Utilización de Oracle Warehouse Builder

Definición de los orígenes de datos

Una vez determinados los requisitos e identificados todos los orígenes de datos, puede utilizarse una herramienta como OWB para construir el almacén de datos. OWB puede gestionar un conjunto de orígenes de datos diversos gracias a los integradores. OWB incluye también el concepto de módulo, que es una agrupación lógica de objetos relacionados. Hay dos tipos de módulos: origen de datos y almacén de datos. Por ejemplo, un módulo de origen de datos puede contener todas las definiciones de las tablas de una base de datos OLTP que esté actuando como fuente de datos para el almacén.

Archivos sin estructura

OWB soporta dos tipos de archivos sin estructura: delimitados por caracteres y de longitud fija. Si el origen de los datos es un archivo sin estructura, el usuario selecciona el integrador para archivos sin estructura y especifica la ruta y el nombre del archivo. El proceso de creación de los metadatos que describen un archivo es diferente del que se utiliza para una tabla de la base de datos.

Datos web

Con la cada vez mayor utilización de Internet, el nuevo desafío al que se enfrentan los almacenes de datos es capturar datos procedentes de sitios web. Hay diferentes tipos de datos en los entornos de e-Business: datos web transaccionales almacenados en las bases de datos subyacentes; datos de flujos de clics almacenados en los archivos de registro de los servidores web; datos de registro en las bases de datos o en los archivos de registro; y datos consolidados de flujos de clics en los archivos de registro de las herramientas de análisis web.

Generación de código

El generador de código es el componente de OWB que lee las definiciones de destino y los mapeos fuente- destino y genera el código para implementar el almacén de datos. El tipo de código generado varia dependiendo del tipo de objeto que el usuario quiera implementar.

Diseño lógico y físico

Antes de generar el código, el usuario ha estado trabajando principalmente en el nivel lógico, es decir, en el nivel de las definiciones de los objetos. En dicho nivel, lo que al usuario le preocupa es capturar todos los detalles y relaciones (la semántica) de un objeto, pero todavía no le preocupa definir ninguna característica de implementación. Por ejemplo, considere una tabla que haya que implementar en una base de datos Oracle. En el nivel lógico, lo que al usuario le preocupa es el nombre de la tabla, el número de las columnas, los nombres y tipos de datos de las columnas y las relaciones que esa tabla pueda tener con otras tablas. Sin embargo, en el nivel fisico, la cuestión es: ¿cómo puede implementarse esta tabla óptimamente en una base de datos Oracle? El usuario debe ahora preocuparse de cosas tales como los espacios de tablas, índices y parámetros de almacenamiento. OWB permite al usuario ver y manipular un objeto tanto en el nivel lógico como en el físico. La definición lógica y los detalles físicos de implementación se sincronizan automáticamente.

2.4.3 APLICACIONES OLAP EN ORACLE

En los grandes entornos de almacén de datos, pueden realizarse muchos tipos de análisis como parte de la construcción de una plataforma para soportar sistemas de inteligencia empresarial. Además de las consultas SQL tradicionales, los usuarios necesitan realizar operaciones analíticas más avanzadas con los datos. Dos de los tipos principales de análisis son el procesamiento analítico en línea (OLAP) y la minería de datos. Describiremos la manera en que Oracle proporciona la tecnología OLAP como uno de los componentes más importantes de su plataforma de inteligencia empresarial.

A) Entorno OLAP de Oracle

La principal ventaja de un almacén de datos es su capacidad para dar soporte a los sistemas de inteligencia empresarial. Hasta la fecha, las aplicaciones estándar de generación de informes y de realización de consultas adhoc se ejecutaban directamente a partir de tablas relacionales, mientras que las aplicaciones más sofisticadas de inteligencia empresarial utilizaban bases de datos analíticas especializadas. Estas bases de datos analíticas

especializadas proporcionan normalmente soporte para cálculos multidimensionales complejos y para funciones predictivas; sin embargo, dependen de la replicación de grandes volúmenes de datos en bases de datos propietarias.

B) Plataforma para aplicaciones de inteligencia empresarial

La base de datos Oracle9i proporciona una plataforma para las aplicaciones de inteligencia empresarial. Los componentes de esta plataforma incluyen la base de datos Oracle9i y Oracle OLAP como utilidad especializada dentro de la base de datos. Esta plataforma proporciona:

- un rango completo de funciones analíticas, incluyendo funciones multidimensionales y predictivas:
- soporte para la obtención de tiempos cortos de respuesta a las consultas, similares a los que normalmente se asocian con las bases de datos analíticas especializadas;
- una plataforma escalable para almacenar y analizar conjuntos de datos multiterabyte;
- una plataforma abierta para aplicaciones multidimensionales y aplicaciones basadas en SQL:
- soporte para aplicaciones basadas en Internet.

C) Base de datos Oracle9i

La base de datos Oracle9i es el fundamento de la tecnología Oracle OLAP, ya que proporciona un mecanismo de almacenamiento de datos escalable y seguro, además de funciones de gestión de resúmenes, un sistema de metadatos, funciones analíticas SQL y características de alta disponibilidad.

Entre las características de escalabilidad que proporcionan soporte para almacenes de datos multiterabyte podemos citar:

• el particionamiento, que permite descomponer los objetos del almacén de datos en componentes físicos más pequeños que pueden gestionarse de forma independiente y en paralelo;

• la ejecución paralela de consultas, que permite a la base de datos utilizar múltiples procesos para responder a una determinada consulta que se curse a través de la API Java OLAPI;

• soporte para NUMA y para sistemas en clúster, lo que permite a las organizaciones utilizar y gestionar de manera efectiva sistemas hardware de gran envergadura;

• el gestor de recursos de la base de datos de Oracle, que ayuda a gestionar comunidades de usuarios diversas y de gran tamaño,

controlando la cantidad de recursos que se permite utilizar a cada tipo de usuario.

Gestión de resúmenes

Las vistas materializadas proporcionan facilidades para gestionar de manera efectiva los datos del almacén. Si las comparamos con las tablas de resumen, las vistas materializadas ofrecen diversas ventajas:

- son transparentes para las aplicaciones y los usuarios;
- permiten gestionar la obsolescencia de los datos;
- pueden actualizarse automáticamente cuando cambien los datos de origen.

Metadatos

Todos los metadatos se almacenan en la base de datos Oracle. Los objetos de bajo nivel, como las dimensiones, tablas y vistas materializadas se definen directamente a partir del diccionario de datos Oracle, mientras que los objetos OLAP de mayor nivel se definen en el catálogo OLAP. Este catálogo contiene objetos tales como carpetas de cubos y de medidas, así como extensiones a las

definiciones de otros objetos, como por ejemplo las dimensiones. El catálogo OLAP define completamente las dimensiones y hechos, por lo que determina completamente el esquema en estrella del almacén de datos.

Funciones analíticas SQL

Oracle ha mejorado las capacidades de procesamiento analítico de SQL introduciendo una nueva familia de funciones analíticas SQL. Estas funciones analíticas permiten calcular:

- clasificaciones ordenadas y percentiles;
- valores basados en ventanas móviles;
- análisis de adelanto/retraso;
- análisis de tipo primero/último;
- estadísticas de regresión lineal.

Las funciones de clasificación ordenada incluyen distribuciones acumulativas, clasificaciones porcentuales y N-mosaicos. Los cálculos basados en ventana móvil identifican agregados móviles y acumulativos, como por ejemplo sumas y promedios. El análisis de

adelanto/retardo permite la realización de referencias directas y inter-filas para realizar cálculos de cambios periodo a periodo.

Tabla 2.8. Funciones analíticas SQL de Oracle.

Tipo	**Utilizadas para**
Clasificación ordenada	Cálculo de clasificaciones ordenadas, percentiles y N-mosaicos para los valores de un conjunto de resultados.
Ventanas móviles	Cálculo de agregados acumulativos y móviles. Pueden aplicarse a las siguientes funciones: SUM, AVG, MIN, MAX, COUNT, VARIANCE, STDDEV, FIRST_VALUE, LAST_VALUE y a las nuevas funciones estadísticas.
Generación de informes	Cálculo de cuotas, como por ejemplo cuotas de mercado. Pueden utilizarse con las siguientes funciones: SUM, AVG MIN, MAX, COUNT (con o sin DISTINCT), VARIANCE, STDDEV, RATIO_TO_REPORT y a las nuevas funciones estadísticas.
LAG/LEAD	

FIRST/LAST.	Localización de un valor en una fila que esté situada a una distancia especificada de la fila actual.
Regresión lineal	Primero o último valor en un grupo ordenado.
Percentil inverso	Cálculo de regresiones lineales y otras estadísticas (pendiente, punto de corte, etc.).
Distinción y clasificación hipotética	El valor de un conjunto de datos que se corresponde con un percentil especificado.
	La clasificación o percentil que una fila tendría si se la insertara en un conjunto de datos especificado.

Fuente: Elaboración propia

CAPÍTULO III

MARCO METODOLÓGICO

3.1 TIPO DE INVESTIGACIÓN

Para llevar adelante el desarrollo de esta tesis se esta aplicando el método exploratorio – descriptivo. El presente trabajo esta dirigido a estudiantes y profesionales que salgan del entorno de los textos tradicionales que tratan este campo desde un punto de vista teórico, que ignoran las aplicaciones que son lo más importante. Se presentan las bases de datos multidimensionales o almacenes de datos con el fin de responder a las necesidades de procesamiento analítico y toma de decisiones de las organizaciones. Se analiza el concepto, junto con las ventajas e inconvenientes, de un almacén de datos y sus principales componentes. Asimismo de da a conocer las principales operaciones soportadas por los SGBD multidimensionales.

En cuanto al diseño de los almacenes de datos se presenta el conocido como diseño en estrella, que permite su implementación en SGBDR, así como los enfoques basados en el modelado conceptual del almacén de datos.

3.2 DISEÑO DE INVESTIGACIÓN

Es una investigación exploratoria descriptiva.

3.3 TÉCNICAS DE RECOLECCIÓN DE DATOS

La información se recopiló de la siguiente forma:

- Recopilar información y realizar una información teórica de los conceptos en que se fundamenta el modelo de datos multidimensional y trabajos orientados a la investigación.
- Análisis documental de material bibliográfico disponible sobre datawarehouse.
- Consulta bibliográfica
- Uso de correo electrónico.

3.4 TÉCNICAS DE ANÁLISIS DE DATOS

- Se aplicó la técnica de investigación científica que nos permitió obtener información, analizarlo e interpretarlo.
- Método descriptivo, para determinar el estado actual de modelo multidimensional para un diseño óptimo de base de datos.
- Método diagnóstico, evaluativo, para conocer que acciones se debe realizar para optimizar base de datos.

Para mantener competitiva una organización necesita una buena gestión de datos, que minimice las duplicidades en su tratamiento y que asegure la calidad de los mismos, de manera que puedan servir como fuente para la toma de decisiones estratégicas y tácticas. Éste es precisamente el enfoque del almacén de datos (datawarehouse), que pretende servir como un área de almacenamiento de datos integrados para la toma de decisiones.

CAPÍTULO IV

MODELO MULTIDIMENSIONAL CONCEPTUAL ORIENTADO A OBJETOS

En este capítulo presentaremos los constructores de modelado que proporciona al modelo para representar las propiedades estructurales y dinámicas de las aplicaciones OLAP a nivel conceptual. En cuanto a la parte estructural, se presentan los constructores que permiten especificar los hechos y las dimensiones con sus respectivas propiedades. Con respecto a la parte dinámica, los requisitos iniciales de usuario se representan mediante clases cubo.

Además, se proporciona un conjunto básico de operaciones OLAP que se pueden aplicar a partir de estas clases cubo para llevar a cabo un análisis más profundo de los datos devueltos por dichas clases. Finalmente, se

modela el comportamiento de los requisitos de usuario mediante la evolución que experimenten estas clases cubo en función de las operaciones OLAP que se apliquen.

Algunas operaciones permitirán obtener clases cubo como estados de otra clase cubo; otras operaciones permitirán interactuar a dos clases cubo iniciales mediante la operación OLAP aplicada.

4.1 Introducción

El diseño conceptual de las aplicaciones OLAP se debería atacar desde una doble perspectiva:

- la parte estructural, que es la representación de las propiedades estructurales del modelo multidimensional como por ejemplo hechos, dimensiones, aditividad, etc. y,

- la parte dinámica, que concierne aspectos sobre la definición de requisitos iniciales de usuario y operaciones OLAP a aplicar sobre los mismos.

En las aplicaciones OLTP, el modelo conceptual realizado no es directamente consultado por el usuario final para definir requisitos. Sin embargo, en las aplicaciones OLAP, el usuario final utiliza la estructura del modelo MD considerado para formular requisitos iniciales. Por tanto, en cierta manera, la estructura del modelo MD determina el tipo de requisitos que se pueden definir.

De igual forma, el tipo de requisitos iniciales que el usuario desee satisfacer, influenciarán el diseño de la parte estructural del modelo multidimensional. Luego, consideramos que representar los requisitos iniciales de usuario en la fase del diseño conceptual de las aplicaciones OLAP aumentará la calidad de los modelos conceptuales obtenidos.

Por ejemplo, supongamos que se ha utilizado el modelo ER para la fase de modelado conceptual de una aplicación OLTP. El usuario final no consulta ni navega por las estructuras definidas en dicho modelo para definir requisitos. Sin embargo, en una aplicación OLAP, por ejemplo, las jerarquías de clasificación definidas en una dimensión determinarán sobre qué elementos se pueden aplicar las operaciones roll-up y drill-down . De igual forma, si un usuario, por ejemplo, desea llevar a cabo un análisis agrupando los datos por la ciudad donde residen los clientes, esto

determinaría que ciudad ha de ser un nivel de jerarquía para la dimensión cliente.

En este contexto, tenemos un modelo conceptual Orientado a Objetos (OO) que permite representar de una forma natural las propiedades tanto estructurales como dinámicas de las aplicaciones OLAP basándose en el paradigma 00. En concreto, este capítulo se centra en definir los constructores de modelado que proporciona el modelo GOLD y que permiten representar todas estas propiedades. Más adelante se presentará la notación gráfica que permite representar dichos constructores y "ocultarlos" convenientemente para facilitar su manejo.

En primer lugar, se definen los constructores que permiten definir los elementos básicos de la parte estructural del modelo multidimensional y sus relaciones (hechos, dimensiones, jerarquías de clasificación, asociación entre niveles de jerarquía, etc.). Las relaciones básicas utilizadas (especialización, asociación y agregación compartida) nos proporcionan un gran detalle sobre los objetos implicados en la relación como por ejemplo, la cantidad de objetos que intervienen en cada una de ellas, lo que nos lleva a considerar aspectos básicos como relaciones

"muchos a muchos" entre un hecho y una dimensión, jerarquías no estrictas, etc .

A partir de la estructura del modelo multidimensional, los requisitos de usuario se definen mediante clases cubo. Según la estructura del modelo y de estas clases cubo, se identifican unos patrones que nos permiten definir el conjunto de operaciones OLAP a aplicar sobre estas clases cubo. El paradigma 00 nos permite encapsular en estas clases cubo propiedades estructurales y dinámicas de los requisitos de usuario. Además, la evolución de los requisitos de usuario se modela de una forma sencilla en función de las operaciones OLAP que se apliquen sobre estas clases.

4 .2 Parte estructural

En esta sección se presentan los constructores de modelado que permiten representar los hechos y dimensiones con sus características propias. Como paso previo, se comienza con la definición de elementos básicos en cualquier aproximación OO como son los atributos y sus dominios y las clases

Definición 4.2.1. Se define A como un conjunto de atributos (a_1 , a_2 , . . ., a_n) definidos sobre dominios.

Un atributo (a_i) se define sobre un dominio específico. El tipo de este dominio es el Tipo Abstracto de Dato (TAD) elegido. Dicho dominio consiste a su vez de un conjunto de valores que son proporcionados por funciones de dominio y, un conjunto de operadores permitidos sobre estos valores.

Por ejemplo, podemos tener un atributo cantidad definido sobre un dominio entero, que es el Tipo Abstracto de Dato escogido. Este TAD define que los posibles valores que puede adoptar un atributo de tipo entero son 0, 1, 2, . . . etc . Además, el TAD proporciona un conjunto de operadores permitidos sobre estos valores como por ejemplo +, -, *, etc.

Como ya es conocido, en el paradigma orientado a objetos se agrupan en clases y comparten así una misma estructura y comportamiento. Para nuestra aproximación, proporcionaremos una sencilla signatura de clase que como características relevantes para los modelos multidimensionales presenta la posibilidad de definir explícitamente:

- el atributo identificador,

- el atributo descriptor y,
- atributos derivados a partir de otros atómicos mediante ciertas reglas de derivación

Definición 4.2.2 (Plantilla de clase). Se define una plantilla de clase como una tupla (H,A,E), donde,

- H es el nombre de la clase.
- $A = A_t \cup A_d$ es el conjunto de atributos, donde:
  - ❖ A_t es el conjunto de atributos atómicos. Dentro de este conjunto de atributos podemos tener dos atributos especiales no necesariamente distintos[11]
    - o el atributo identificador (OID),
    - el atributo descriptor (D) que será el utilizado por defecto para el posterior análisis de los datos
  - ❖ A_d es el conjunto de atributos derivados que se calculan a partir de otros mediante ciertas reglas de derivación que son expresiones lógicas correctas[12] que pueden contener operadores de agregación relacionales (Ejm. SUM, AVG, etc.).

[11] El mismo atributo puede ser identificador y descriptor a la vez

[12] Adoptamos como expresiones lógicas correctas las reglas de derivación del modelo relacional (Codd, 1990)

- E es el conjunto de eventos permitidos sobre los objetos de dicha clase, siempre existen dos eventos definidos por defecto que son "new" y "destroy" para crear y destruir objetos respectivamente.

El atributo descriptor (D) será el atributo utilizado por defecto para el análisis de los datos al aplicar las operaciones OLAP. Como se verá más adelante, su definición es obligatoria para las clases que representen niveles de jerarquía. Por ejemplo, supongamos que tenemos una clase proveedores, si se define el atributo nombre como el atributo descriptor, al visualizar los objetos de la clase proveedores y aplicar operaciones OLAP como por ejemplo roll-up o drill-down, por defecto se visualizarán sus nombres.

Con respecto al atributo identificador, en las bases de datos orientadas a objetos (BDOO) la existencia de un atributo identificador es intrínseca a la definición de clase y cada objeto crea do en el sistema lleva consigo un identificador que lo distingue unívocamente de los demás objetos creados.

Sin embargo, en el contexto de las aplicaciones OLAP, cada nivel de jerarquía definido ha de tener un atributo que identifique unívocamente a

las instancias de dicho nivel. Esto es necesario especificarlo explícitamente en la definición del modelo multidimensional de la propia herramienta OLAP ya que será utilizado internamente cuando se apliquen las operaciones de navegación OLAP (roll-up y drill-down)

Por otro lado, una posible solución para implementar una relación "muchos a muchos" entre un hecho y una dimensión en particular es añadir atributos identificadores en los hechos que ayuden a identificar unívocamente a las instancias de los hechos.

Por lo tanto, según las dos razones presentadas anteriormente, si deseamos abordar el paso de la generación semi-automática de un esquema de la base de datos par que pueda ser directamente interrogado por una herramienta de análisis OLAP del mercado, es necesario que la plantilla de clase admita la definición de atributos identificadores (OID) para que dicha generación semiautomática se lleve a cabo de forma correcta.

En cuanto a los atributos derivados distinguimos entre dos tipos:

- los que no contienen ningún operador de agregación relacional en su regla de derivación (ejm. precio_total = precio * cantidad) y,

- los que contienen algún operación de agregación relacional en dicha regla de derivación (ejm. cant_total = SUM(cantidad))

Cuando se utiliza un operador de agregación relacional para la definición de un atributo derivado, en el posterior análisis de los datos no se podrá aplicar ningún otro operador de agregación sobre dicho atributo derivado. En el ejemplo anterior, cant_total se calculará siempre como SUM(cantidad) y no se permitirá utilizar ningún otro operador de agregación sobre dicho atributo ni en la definición de requisitos ni en las posibles operaciones OLAP que se apliquen sobre él.

Esto permite una gran flexibilidad en el diseño ya que permite fijar el único operador de agregación que se pueda aplicar sobre un atributo en el momento de su definición. Si se quiere dar la posibilidad de aplicar distintos operadores de agregación en la fase de análisis sobre un atributo, éste no deberá contener ningún operador de agregación en su regla de derivación

En nuestro modelo, las relaciones que se pueden establecer entre clases se definen como un subconjunto de aquellas definidas en UML. A

continuación describimos los aspectos relevantes de cada una de ellas que adoptamos en el presente trabajo

- **Asociación:** una asociación representa relaciones entre instancias de clases (ejemplo una persona trabaja para una compañía). En general los roles[13] de las asociaciones no estarán etiquetadas con nombres, aunque no se pone ninguna restricción al respecto. En principio, suponemos que la navegabilidad de la asociación es siempre en ambos sentidos (ejm. dada una comunidad podremos conocer las ciudades con las que se relaciona y, dada una ciudad podremos conocer la comunidad con la que se relaciona). En cuanto a la cardinalidad (multiplicidad), en principio no se pone ninguna restricción sobre la cantidad de objetos que intervienen de cada clase en una asociación, si bien en algún constructor concreto del modelo se puede limitar dicha multiplicidad.

- **Agregación** compartida. La agregación es un caso especial de asociación que indica que las relaciones entre clases es una relación del tipo "parte-todo" (ejm. Un coche está compuesto de cuatro ruedas). Una agregación compartida es un tipo de agregación

[13] papeles

en la que las partes pueden ser partes de cualquier todo (ejm. Un equipo se compone de personas y, una persona a su vez, puede pertenece a varios equipos ; es decir, que la pertenencia de persona a equipo no es exclusiva) .

- **Generalización**. La generalización es una relación entre una clase genérica y una específica. La clase específica (llamada también sub-clase, clase hija o clase especializada) hereda tanto la parte estática como dinámica de la clase general (llamada también super-clase, clase padre o clase generalizada); es decir, todos los atributos, operaciones y asociaciones de la clase padre son heredados por la clase hija

Por tanto, un modelo multidimensional está formado por un conjunto de dimensiones, un conjunto de hechos y un conjunto de clases cubo base. Estas clases cubo especifican los requisitos iniciales de usuario capturados en la fase de diseño conceptual y, serán descritas con mayor detalle al introducir la parte dinámica del modelo. A continuación, definimos los elementos básicos de que consta un modelo multidimensional

Definición 4.2.3. Se define un modelo multidimensional (MD) como una tupla (D,F,CC), donde,

- D es un conjunto de dimensiones $D=\{d_1 , d_2 , \ldots , d_n,\}$, donde d_n, es un nombre de dimensión
- F es un conjunto de hechos $F=\{f_i, f_2, \ldots , f_k\}$, donde f_k es un nombre de hecho
- CC es un conjunto de Clases Cubos Base $CC=\{ cc_1, cc_2, \ldots , cc_m\}$ donde cc_m, es un nombre de una clase cubo

Ejemplo: según el caso de estudio de Ventas de productos, introduciremos los modelos multidimensionales, como sigue:

Ventas_de_productos=(D,F,CC), donde:

D = {Producto, Almacén, Cliente, Tiempo}

F = {Ventas_productos}

CC = { . . .}

Donde el nombre del modelo multidimensional es Ventas_de_productos y, que está compuesto por un hecho (Ventas_productos), cuatro dimensiones (Producto, Almacén, Cliente, y Tiempo). En cuanto a las clases cubo, no se define específicamente ningún requisito inicial de usuario en este primer ejemplo por no haber sido presentadas convenientemente.

Una vez presentadas las componentes fundamentales del modelo multidimensional, a continuación se definirán los constructores de modelado que permitirán especificar convenientemente las dimensiones y los hechos. Empecemos con las dimensiones.

4.2.1 Dimensiones

Las dimensiones se definen mediante clases, las cuales pueden relacionarse entre sí mediante relaciones de asociación o generalización para formar jerarquías de clasificación o especialización respectivamente. Ambas jerarquías se representan mediante un grafo acíclico dirigido de jerarquía de clasificación (GAD) y un grafo de jerarquía de especialización (H) respectivamente.

Definición 4.2.4. Se define una dimensión d_i como una tupla (C, T, GAD, H), donde,

- $C=\{c_1 , c_2, . . ., c_d \}$ es el conjunto de clases que componen la dimensión, donde,
  - C_1 es la clase dimensión que contendrá los atributos que caracterizan a los objetos de dimensión en el nivel de jerarquía más básico, es decir, el mínimo nivel de detalle posible. El nombre de la dimensión d_i será el nombre de esta clase c_1
  - $c_2, . . ., c_{d-1}$ son clases base utilizadas para la definición de los distintos niveles de jerarquía de clasificación o especialización.
  - Cada dimensión contiene una clase especial C_d denominada <nombre_dimensión> .all que contiene un solo objeto que está compuesto por todas las instancias de objetos de las clases con las que se relaciona.
- T es un atributo booleano que determina si la dimensión es del tipo tiempo o no.

- GAD es un grafo acíclico dirigido de jerarquía de clasificación definido sobre un subconjunto de clases $C' \subseteq C$ con raíz en c_1 GAD=(C, V), siendo C el conjunto finito de clases (niveles de jerarquía de clasificación) $c_1, c_2, \ldots, c_k$ y V={ (c_i, c_j ; card_origen, card_destino[;c]) / $i \neq j \wedge 1 \leq i, j \leq k \wedge c_j$ es una asociación de $c_i \wedge c_1$ es la clase de dimensión} .

  - Cada clase $c_1, \ldots, c_{d-1} \in C'$ definida bajo el GAD tendrá definida obligatoriamente un atributo descriptor y un atributo identificador.
  - Card_origen, card_destino describen las cardinalidades mínimas y máximas de la asociación entre c_i y c_j respectivamente, es decir, la cantidad de objetos mínimos y máximos que participan en la asociación. Así tenemos:
    - 1 : interviene un único objeto.
    - 0..1 : ningún objeto o uno únicamente.
    - 1 . . *: uno o más objetos.
    - *: 0, 1 ó más objetos .

  Las cardinalidades entre una clase c_i y <nombre_dimensión> all es siempre: card_origen=1..* y card_destino=1 para

asegurarnos una correcta agregación de todos los valores en uno solo.

- [;c] : es un atributo opcional definido sobre una asociación que delimita si la jerárquica de clasificación entre las clases es completa, lo que significa que una vez que un objeto compuesto se ha asociado con n objetos componentes; dicha relación no puede variar.

- H es un grafo de jerarquía de especialización definido sobre un subconjunto de clases $C' \subseteq C$ con raíz en c_1. H=(C, V), siendo C el conjunto finito de clases $c_1, c_2, . ., c_k$ y $V=\{ (c_i, sp, c_j) / i \neq j \wedge 1 \leq i, j \leq k \wedge cj$ es una especialización de c_j mediante s_p (o c_j es una generalización de $c_i) \wedge c_1$ es la clase de dimensión} .

  Además, si una clase c_i pertenece a una jerarquía de clasificación, dicha clase no puede ser clase especializada de otra. Dicho de otra forma, si dadas dos clases c_i, c_j , donde $c_i \in H$ y $c_j \in GAD$, entonces $ci \neq cj$.

- Cada clase $c_1, c_2, . ., c_{d-1} \in C'$ definida bajo H heredará el atributo descriptor y el atributo identificador de su clase padre.

Una dimensión se compone de n clases, donde dos de ellas son especiales. Por un lado la clase de dimensión que representa el mínimo nivel de detalle y da el nombre a la dimensión[14] y, por otro, la clase <nombre_dimensión> .all que es la clase que representa al mayor nivel de jerarquía posible para cada camino del grafo.

Definición 4.2.5. Sea GAD el grafo acíclico dirigido definido sobre un subconjunto $C' \subseteq C$ de clases de una dimensión D, definimos un **camino de jerarquía de clasificación** sobre dicha dimensión $HP_D=(c_1, c_2, . ., c_d, c_1.all)$ como un camino del GAD de tal forma que $c_1, c_2, . ., c_d, \in C'$.

Cada camino de jerarquía de clasificación dentro del GAD comienza en el menor nivel de jerarquía (la clase de dimensión, c_1) y finaliza en el mayor nivel de la jerarquía posible (definido por la clase especial <nombre-dimensión.all>). Así, cada camino se considera como una lista linealmente

[14] Por ello a partir de ahora nos referiremos a dimensión cuando en realidad estamos utilizando el nombre de la clase dimensión.

ordenada de niveles de jerarquía en el que cada nivel viene representado por una clase. El único objeto (ALL) que contiene esta clase especial será utilizado por las operaciones OLAP para agregar todos los valores en un único valor.

Definición 4.2.6. Sea H el grafo de jerarquía de especialización definido sobre un subconjunto $C' \subseteq C$ de clases de una dimensión D, definimos un **camino de jerarquía de especialización** sobre dicha dimensión $SP_D=(c_1, sp_n , sp_{n-1} , . . .,sp_1 ,c_1 .all)$ como un único camino de niveles de especialización donde sp_1, sp_2, ..., sp_n son los nombres de los n niveles de especialización .

Una dimensión tendrá definido un único camino de jerarquía de especialización, ya que existe un grafo de jerarquía de especialización por dimensión y éste contiene solo un camino. Se puede ver que dicho camino también finaliza en la clase especial <nombre.dimensión>.all por los mismos motivos expuestos anteriormente

Definición 4.2.7. Sea $SP_D = (c_1, sp_n, sp_{n-1}, \ldots, sp_1, c_1 .all)$ un camino de jerarquía de especialización definido en una dimensión D, definimos la función $dom_{spD}\ sp_i = \{c_1, c_2, \ldots, c_n\}$ como la función dominio que devuelve las n clases que pertenecen al nivel de jerarquía de especialización sp_i

4.2.2 Hechos

Una vez definidas todas las dimensiones del modelo junto con las n clases que forman cada dimensión, pasamos a representar los hechos. Un hecho se definirá como una clase compuesta que se relacionará mediante una relación de agregación compartida con una clase por cada una de las dimensiones definidas. Esta clase de cada dimensión puede representar cualquier nivel de jerarquía de especialización o clasificación. Ello nos permitirá la flexibilidad de definir hechos como por ejemplo una clase denominada compras de productos que para la dimensión almacén se relacione con la clase ciudad; lo que significará que se desean observar las compras de productos, no por el almacén donde han sido compradas sino por la ciudad en la que se compraron.

Definición 4.2.8 . Sean c_1 , c_2, . . ., c_n, n clases que pertenecen a las n dimensiones definidas, definimos un **hecho (F)** como una clase compuesta (FC) en una relación de agregación compartida donde las n clases son las componentes. Dicho de otra forma:

F={(FC, c_1; card_origen, card_destino), (FC, c_2; card_origen, card_destino), . . ., (FC, c_n; card_origen, card_destino)}, donde,

- FC es la clase de hechos. El hecho (F) adopta el nombre de esta clase de hechos,
- c_i es la i-ésima clase que pertenece a la i-ésima dimensión, de tal forma que dadas dos clases cualesquiera c_i y c_j, no pertenecen a la misma dimensión; es decir, $c_i \in d_i$ y $c_j \in d_j$, con $d_i \neq d_j$,
- card_origen y card_destino describen las cardinalidades mínimas y máximas de la relación de agregación compartida entre la clase de hechos y cada una de clases de cada dimensión con la que se relaciona, con card_destino≥1.

La clase de hechos (FC) se relaciona con las dimensiones a través de las clases de dimensión, de tal forma que una clase de hechos no puede estar relacionada con una dimensión a través de dos clases distintas.

Considerar las cardinalidades en la relación de agregación compartida entre la clase de hechos y las clases de las dimensiones, nos permite representar las relaciones "muchos a muchos" entre un hecho y una dimensión en particular. Desde un punto de vista conceptual, esta situación puede requerir que el modelador defina un atributo identificador en la clase de hechos que ayude a identificar unívocamente a las instancias de dicha clase. En analogía con el modelo ER, este atributo sería como el atributo multivaluado definido en una relación "muchos a muchos" entre dos entidades. Este atributo identificador formará parte de la clave primaria de la tabla de hechos y permitirá así identificar unívocamente a las instancias de dicha tabla.

Definición 4.2.9 (Aditividad). Sea a_i un atributo de dimensión o hecho, se dice que a_i es:

- **aditivo**, si el operador SUM se puede aplicar para agregar valores de a_i a lo largo de todas las dimensiones,
- **semi-aditivo**, si el operador SUM no se puede aplicar a lo largo de alguna dimensión para agregar valores de a_i,
- **no-aditivo**, si el operador SUM no se puede aplicar a lo largo de ninguna dimensión para agregar valores de a_i.

Con la definición anterior establecemos que si un atributo es aditivo, cualquier operador de agregación se podrá aplicar para agregar sus valores a lo largo de todas las dimensiones. Sin embargo, si es semi-aditivo o no-aditivo, indica que el operador SUM no se podrá aplicar a lo largo de alguna o ninguna dimensión respectivamente. Sin embargo, sobre estos dos últimos tipos de atributos se puede desear aplicar algún otro operador de agregación relacional como por ejemplo AVG, MIN o MAX. Por ello, necesitamos capturar el concepto de aditividad de tal forma que nos permita especificar los operadores de agregación concretos que se pueden aplicar sobre los atributos. Para tal efecto, definimos a continuación los **patrones de agregación**.

Definición 4.2.10. Sea a_i un atributo de hecho o dimensión, se define un **Patrón de agregación** sobre dicho atributo como una tupía AP(ai , dj, agt), donde,

- a_i es un atributo,
- si $a_i \in FC$, es decir, si a_i es un atributo de hecho, entonces $d_j \in D$ / {ALL}, donde d_j; es una dimensión (D) y la etiqueta ALL hace referencia a cualquier dimensión definida en el modelo. Pero si el

atributo $a_i \in D$, es decir, si a_i es un atributo de una dimensión, entonces $d_i \in D / F / \{ALL\}$, con F un hecho.

- agt puede a su vez ser:
  - una lista de los operadores de agregación que se pueden aplicar[15] a lo largo de la dimensión d_j,
  - la etiqueta ALL, para indicar que cualquier operador de agregación se puede aplicar a lo largo de la dimensión d_j,
  - la etiqueta c, para indicar que ningún operador de agregación se puede utilizar a lo largo de la dimensión d_j.

4.3 Parte Dinámica

La parte dinámica del modelo multidimensional consta fundamentalmente de dos aspectos: la definición de requisitos iniciales de usuario y, un conjunto de operaciones OLAP que permiten al usuario cambiar la forma de analizar los datos devueltos por los requisitos iniciales. Además de representar estos dos aspectos, modelamos la evolución del comportamiento que experimentan los requisitos de usuario en función de

[15] Podemos tomar como punto de partida los clásicos operadores de agregación relacionales SUM, AVG, MAX, MIN y COUNT

las operaciones OLAP que se apliquen. Este hecho proporciona calidad al modelo obtenido ya que en el momento de definir el requisito, el usuario conoce no sólo la parte estructural del modelo multidimensional, sino también el tipo de operaciones OLAP que podrá aplicar sobre cada requisito definido. Ello permite una mayor seguridad sobre la idoneidad del modelo multidimensional definido para satisfacer los requisitos básicos de usuario.

4.3 .1 Modelado de los requisitos de usuario

En las aplicaciones OLAP, la definición de requisitos se lleva a cabo a partir de la estructura del modelo multidimensional que se interroga. Una vez analizados los datos devueltos por tales requisitos (normalmente en forma de tabla o cubo multidimensional), el usuario aplica operaciones OLAP sobre la definición del requisito inicial` para analizar los datos desde distintas perspectivas.

En esta sección definimos las clases cubo (CC) como una forma sencilla e intuitiva de modelar los requisitos de usuario. La definición de estas CC están basadas en un hecho, que proporciona los atributos objeto de

análisis y n dimensiones, que proporcionan las distintas perspectivas para analizar los atributos de hecho. Con referencia a los métodos permitidos sobre las clases cubo, definiremos el conjunto de operaciones OLAP que se podrán aplicar sobre estas clases cubo y que nos permitirán definir nuevos requisitos. A continuación pasamos a definir las clases cubo.

Definición 4.3.1. Sean d_1, d_2 , . . ., d_n n dimensiones y F un hecho, definimos una clase cubo (CC) como una tupla (H,M,S,D_C,CO), donde,

- H (Head:) es el nombre de la clase
- M (Measures:) M={a_{d1}, / p_1 , a_{d2} / P_2, . . .,a_{dn},/P_n} es el conjunto de atributos de hecho a analizar, donde $a_{di} \in$ FC es un atributo de hecho que contiene un operador de agregación relacional en su definición y p_i es una expresión lógica correcta que contiene algún operador de agregación relacional si y solo si los atributos que contiene dicha expresión no son atributos derivados que a su vez están definidos utilizando algún operador de agregación relacional en su regla derivación.
- S (Slice :) S=s_1 <op> s2 <op> . . . <op> s_n es una expresión lógica compuesta de predicados atómicos (s_i) que representan restricciones que han de cumplir los valores devueltos, donde < op> representa un

operador lógico (ejm. $\wedge$(y), $\vee$(o)) y s_i son expresiones lógicas de la forma:

d_k . (c_i[.a_i] <o> v l sp_i '=' c_j), donde,

- dk es la k-ésima dimensión,
- c_i es una clase que representa a un nivel de jerarquía de clasificación de d_k ,
- a_i es un atributo de la clase c_i , si a_i es omitido, entonces se tomará el atributo definido como descriptor,
- <o> es un operador de comparación lógico,
- sp_i es un nivel de jerarquía de especialización de d_k,
- c_j es una clase que pertenece al nivel de especialización spi, es decir, $c_j \in$ dom sp_{dk} sp_i.
- Tomamos la lógica bivaluada para evaluar estas expresiones lógicas. En dicha lógica existen dos valores {cierto, falso} y, por tanto, las expresiones lógicas descritas anteriormente se pueden evaluar a cierto o falso.

- D_C (Dice:) DC={dc_1 , dc_2, . . .dc_n,} es el conjunto de las n condiciones de agrupamiento requeridas para el análisis de los datos con dc_i expresiones de la forma:

  d_k . (c_i / sp_i), donde,

  - d_k es la k-ésima dimensión,

- ci es una clase que representa a un nivel de jerarquía de clasificación de d_k y,
- sp_i es un nivel de jerarquía de especialización de d_k

Para cada dimensión considerada en la sección Dice ($d_k \in D_c$) se ha de cumplir que cada operador de agregación o_j aplicado a cada atributo objeto de análisis ($a_{di}, \in M$) sea un operador permitido y definido en los patrones de agregación definidos para dicho atributo a_{di} , es decir, $o_j \in agt$ con $AP(a_{di}, d_k, agt)$.

- CO (Cube Operations:) es el conjunto de operaciones OLAP.

Es obligatorio que los atributos de hecho (a_i) incluidos en la sección Measures (M) tengan definido un operador de agregación relacional ya que las clases cubo obtendrán datos agregados y "agrupados" por las condiciones de agrupamiento definidas en la sección Dice (D_c) . Esto se puede conseguir de dos formas:

- incluyendo una medida derivada a_{di} en cuya regla de derivación se utilice un operador de agregación. De esta forma la regla de derivación es fija para dicha medida o,

- definiendo predicados (p_i) que contengan operadores de agregación sobre medidas ya definidas en la clase de hechos

Esto proporciona la flexibilidad de definir las reglas de derivación con los operadores de agregación en la definición de los atributos en las clases de hechos (ver definición del plantilla de clase) o, definir los operadores de agregación al incluir el atributo en la definición de la clase cubo. La diferencia radica en que si el operador se incluye en la definición del atributo en la clase de hechos, el operador utilizado será siempre el mismo durante toda la fase de análisis. Si por el contrario, el operador de agregación se define en el momento de definir la clase cubo, éste puede variar al definir otras clases cubo y al aplicar ciertas operaciones OLAP.

4.3.2 Patrones de navegación para operaciones OLAP

Las clásicas operaciones OLAP roll-up y drill-down permiten navegar por la estructura del modelo multidimensional y llevar a cabo una agregación y desagregación de datos respectivamente a lo largo de los niveles de jerarquía de clasificación definidos en la

dimensiones sobre las que se aplican. Por tanto, la posibilidad de aplicar dichas operaciones sobre una clase cubo depende de la estructura de dicha clase cubo y de la estructura del modelo multidimensional.

Por esta razón, mostrará las distintas relaciones que pueden tener los niveles de jerarquía y atributos que pueden ser incluidos como condiciones de agrupamiento sobre los que se desea aplicar una operación de navegación OLAP. Clasificamos estas relaciones en patrones de navegación; Trujillo (2001), que contendrán información sobre qué operaciones OLAP se pueden aplicar sobre qué elementos . Como se verá más adelante, además de las clásicas operaciones roll-up y drill-down, existe dos nuevas operaciones de navegación: Combine y Divide que permiten navegar por la estructura del modelo cuando no existe ninguna jerarquía explícita definida entre los elementos sobre los que se aplican.

En principio distinguimos tres situaciones (patrones) posibles:

- cuando dos niveles de jerarquía pertenecen a un mismo camino de jerarquía,

- cuando dos niveles de jerarquía no pertenecen a un mismo camino de jerarquía,
- cuando un atributo pertenece a un nivel de jerarquía de especialización o clasificación.

Definición 4.3.2. Se define un **patrón de jerarquía** como una tupla P_{HP} = (c_i c_j ,Roll-up, Drill-down), donde,

- $(c_i, c_j) \in HP_{dk}$ / $(c_i,c_j) \in SP_{dk}$, es decir, c_i , c_j son dos niveles que pertenecen a un camino de jerarquía de clasificación (HP) o a uno de especialización (SP) para una determinada dimensión d_k
- Roll-up , Drill-down son las operaciones a aplicar a lo largo de c_i, c_j para agrupar y separar datos respectivamente

Este patrón define que cuando dos niveles de jerarquía cualesquiera (c_i, c_j) pertenezcan o bien a un mismo camino de jerarquía de clasificación o bien a uno de especialización, se podrán aplicar las operaciones Roll-up y Drill-down para agrupar y separar datos respectivamente .

Definición 4.3.3 . Se define un **patrón de no jerarquía** como una tupla P_{NHP}=(c_i, c_j, Combine, Divide), donde:

- $(c_i,c_j) \notin HP_{dk}$, / $(c_i,c_j) \notin SP_{dk}$, es decir, que c_i y c_j son dos niveles de jerarquía de una misma dimensión que no pertenecen ni a un mismo camino de jerarquía de clasificación (HP) ni a un mismo camino de jerarquía de especialización (SP) para una determinada dimensión d_k .
- Combine, Divide son las operaciones a aplicar sobre c_i, c_j para aumentar y disminuir el nivel de detalle respectivamente.

Con este patrón de navegación resolvemos la situación de navegar a través de niveles de jerarquía que no pertenecen a un mismo camino de jerarquía. El efecto de las operaciones **Combine** y **Divide** es el mismo que el de **Drill-down** y **Roll-up** en el sentido de que incrementan y disminuyen el nivel de detalle respectivamente.

Definición 4.3.4. Se define un **patrón de atributos** como una tupla $P_{AP}=(c_i, a_j, \text{Combine, Divide})$, donde,

- $c_i \in HP_{dk}$, / $c_i \in SP_{dk}$, es decir, c_i pertenece a un camino de jerarquía de clasificación o a uno de especialización para una determinada dimensión d_k. Si $c_i \in HP$, entonces a_j es un atributo de dicha clase, pero si $c_i \in SP$, entonces a_j es un atributo de cualquiera de las clases padre de de c_i.

- Combine, Divide son las operaciones a aplicar sobre c_i para incrementar y disminuir el nivel detalle respectivamente mostrando o escondiendo el atributo a_j.

4.3.3 Operaciones OLAP

Presentamos el conjunto de operaciones OLAP que se puede aplicar sobre una clase cubo, que clasificamos en dos grupos:

- las que permiten navegar por las jerarquías de clasificación y especialización (Roll-up, Drill-down, Combine, Divide) y,
- Las que permiten variar las condiciones especificadas en una clase cubo (Slice, Rotate).

A continuación mostramos una breve definición intuitiva de todas las operaciones OLAP definidas:

- **Roll-up, Drill-down** agrupan y separan datos respectivamente a lo largo de niveles de jerarquía que pertenecen a un mismo camino

de jerarquía de clasificación o a uno de especialización (Patrón P_{HP}).

- **Combine, Divide** (caso 1) introducen y eliminan condiciones de agrupamiento respectivamente cuando no existe un mismo camino de jerarquía de clasificación o de especialización entre las condiciones de agrupamiento consideradas para cada dimensión (Patrón P_{NHP}).
- **Combine, Divide** (caso 2) introducen y eliminan condiciones de agrupamiento respectivamente cuando una de ellas es un atributo que pertenece a una clase que representa a un nivel de jerarquía de clasificación o de especialización (Patrón P_{AP}).
- **Slice** (caso 1) selecciona un subconjunto de los datos devueltos por una clase cubo al introducir una restricción más restrictiva que las actuales definidas en la clase cubo, esto es, restringe aún más las condiciones de la sección slice de la clase cubo.
- **Slice** (caso 2) introduce una restricción más general y, por tanto, no tiene por qué seleccionar un subconjunto de datos.
- **Rotate** Permite añadir y eliminar condiciones de agrupamiento en una clase cubo.

ROLL-UP

La operación Roll-up lleva a cabo una agregación sobre los valores de las medidas devueltos por una clase cubo y los agrupa a lo largo de los niveles de jerarquía definidos en las dimensiones. Dicha operación se aplica bajo el patrón de agregación PHP ya que se agregan datos a lo largo de niveles de jerarquía definidos en las dimensiones.

Definición 4.3.5 (Roll-up). La operación Roll-up lleva a cabo una agregación sobre los valores de las medidas devueltos por una clase cubo a lo largo de dos niveles de jerarquía de clasificación o de especialización de una o más dimensiones. Su signatura es:

Roll- $up_{[agg,]\{d1(ci,cj),\ldots,dn(ci,cj)\}}$ CC = CC' donde,

- CC=(H,M,S,D_c,CO) es la clase cubo de entrada,
- agg es el operador de agregación a aplicar sobre la medida $a_i \in M$, de tal forma que agg $\in$ agt con AP(a_i, d_k, agt) y a_i no es un atributo derivado que contiene algún operador de agregación relacional en su definición en la clase de hechos ,

- d_k es la k-ésima dimensión sobre la que se desea agrupar los datos, con $k \leq n$,
- (c_i, c_j) son los niveles de jerarquía de la k-ésima dimensión, de tal forma que se ha de cumplir que, $d_k.c_i \in D_c \wedge (c_i,c_j) \in HP_{dk} / (c_i, c_j) \in SP_{dk} \ \forall \ d_k \ (c_i, c_j)$, con HP_{dk} y SP_{dk} caminos de jerarquía de clasificación y especialización respectivamente. Se exige que el nivel inicial $d_k .c_i$ esté definido como condición de agrupamiento en la sección Dice (D_C),
- $CC'=(H,M',S,D_C',CO)$, donde $D_C'=(D_C - d_k .c_i) \ U \ d_k .c_j$ si $d_k .c_j \notin D_C$, $M'=\{agg(a_1), \ldots, agg(a_n)\} \ \forall \ a_i \in M$.

La operación Roll-up aplica un operador de agregación (agg) sobre los valores devueltos por las medidas (ai) definidas en una clase cubo (CC) . Para poder aplicar el operador de agregación se tienen que dar dos condiciones:

- que dicho operador sea un operador permitido sobre la medida a aplicar, es decir, que esté definido en el patrón de agregación ($AP(a_i, d_k, agt)$) que define los posibles operadores de agregación a aplicar sobre la medida.
- que la medida no esté definida en la clase de hechos como una medida derivada utilizando algún operador de agregación, ya que

esto obligaría a que dicho operador fuera el único permitido para obtener su cálculo.

DRILL-DOWN

La operación Drill-down lleva a cabo una desagregación implícita sobre los valores de las medidas devueltos por una clase cubo y los separa a lo largo de los niveles de jerarquía definidos en las dimensiones. Por ello, dicha operación se aplica bajo el patrón de agregación P_{HP}.

Definición 4.3.6 (Drill-down). La operación Drill-down lleva a cabo una desagregación implícita sobre los valores de las medidas devueltos por una clase cubo a lo largo de dos niveles de jerarquía de clasificación o de especialización de una o más dimensiones.

Drill-down$_{\{d1(cj, ci),.. .,dn(cj, ci)\}}$ CC = CC' donde,

- CC=(H,M,S,D_C,CO) es la clase cubo de entrada,
- d_k es la k-ésima dimensión sobre la que se desea separar los datos, con $k \leq n$,

- (c_j, c_i) son los niveles de jerarquía de la k-ésima dimensión, de tal forma que se ha de cumplir que, $d_k.c_i \in D_c \wedge (c_i,c_j) \in HP_{dk}$ / $(c_i, c_j) \in SP_{dk}$ $\forall$ d_k (c_j, c_i), con HP_{dk} y SP_{dk} caminos de jerarquía de clasificación y especialización respectivamente .

  Se exige que el nivel inicial d_k .c_j ha de estar definido como condición de agrupamiento en la sección Díce (D_C),
- CC'=(H,M,S,D_C',C0), donde D_c'=(D_c) U d_k .c_i .

La operación Drill-down lleva a cabo una desagregación implícita de los valores de las medidas (a_i) devueltos por una clase cubo (CC) . El operador de agregación que se aplicará sobre las medidas puede ser:

- el utilizado en la definición de la medida derivada si lo hubiese o,
- el aplicado en la clase cubo inicial (CC) sobre la que se aplica la operación Drill-down

Esta desagregación implícita se lleva a cabo porque las nuevas condiciones de agrupamiento especificadas en la sección Dice (D_c) de la nueva clase cubo (CC') corresponden a niveles de jerarquía inferiores que los de la clase cubo inicial (CC). Estos nuevos niveles de jerarquía inferiores provocan, además, que los valores de las dimensiones sean separados.

Definición 4.3.7 (Combine). La operación Combine aumenta las condiciones de agrupamiento de una clase cubo bajo dos situaciones:

- Patrón de no jerarquía (P_{NHP}): entre niveles de jerarquía que no pertenecen al mismo camino de jerarquía.
- Patrón de atributos (P_{AP}): entre un nivel de jerarquía y uno de sus atributos.

$$Combine_{\{d_1(c_j,(a_i/c_i)),\ldots,d_n(c_j,(a_i/c_i))\}}CC = CC' \text{ , donde,}$$

- CC=(H,M,S,Dc,C0) es la clase cubo de entrada,
- d_k es la k-ésima dimensión sobre la que se desea incluir una nueva condición de agrupamiento, con $k \leq n$,
- $(c_j , (a_i / c_i))$ son las nuevas condiciones de agrupamiento.
- $CC' = (H,M,S,D_C', CO)$, donde $D_C' = D_C \cup d_k.c_i [.a_i]$.

Elementos Multidimensionales

El DW contiene una gran cantidad de "Medidas", los analistas quieren entender y comparar. Estudiar todos juntos sería casi imposible vamos a ver cómo estos datos pueden ser, sucesivamente, agrupados en diferentes niveles de detalle para facilitar su gestión vamos a tener "medidas" agrupados en celdas de diferentes clases (que puede ser visto como n dimensiones Cubos), que se agrupan sobre la base del tipo de hecho que representan.

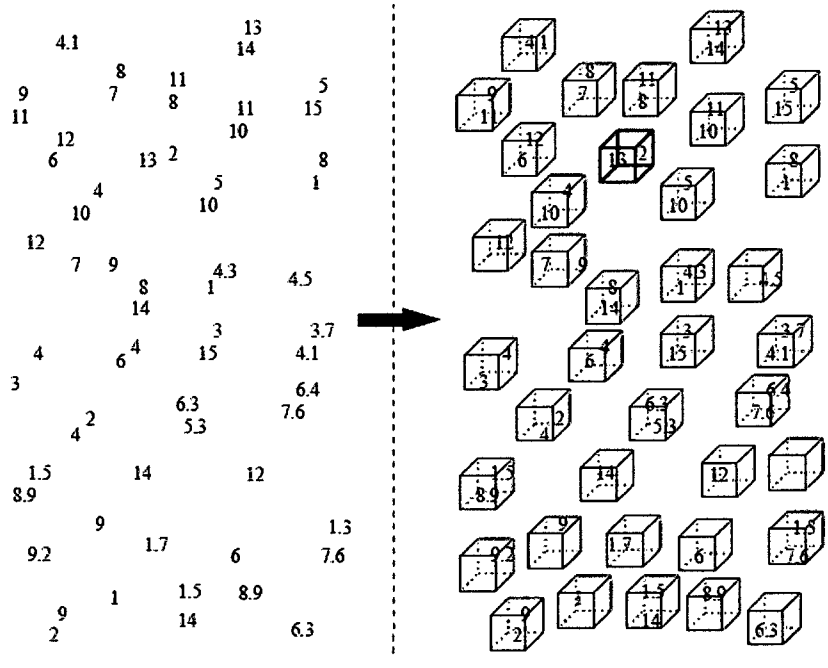


Figura 4.1: Medidas agrupadas en las casillas correspondientes a los hechos.

Fuente: GUNDERLOY MIKE – SNEATH TIM.

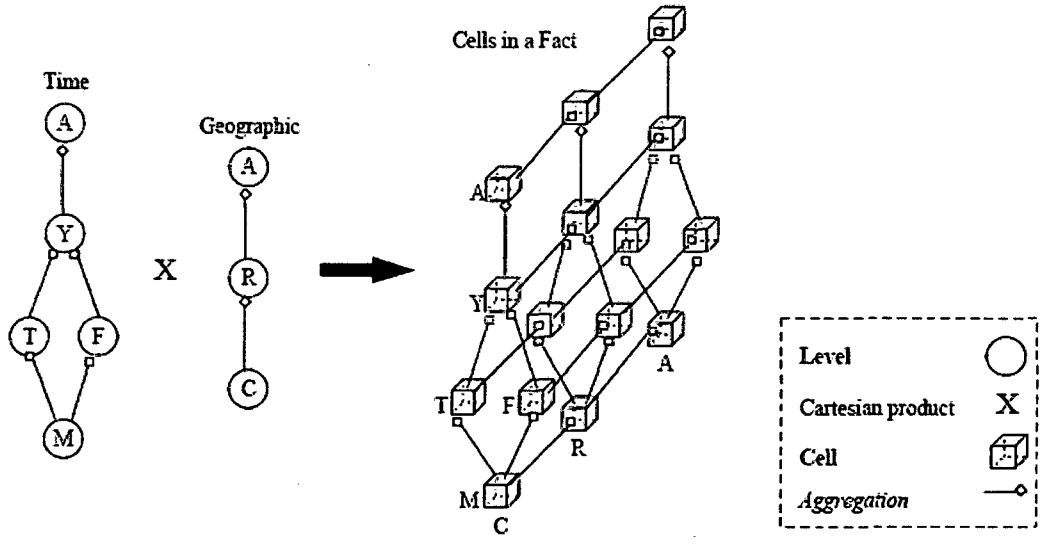


Figura 4.2: Celdas en un hecho con dos dimensiones.

Fuente: GUNDERLOY MIKE – SNEATH TIM.

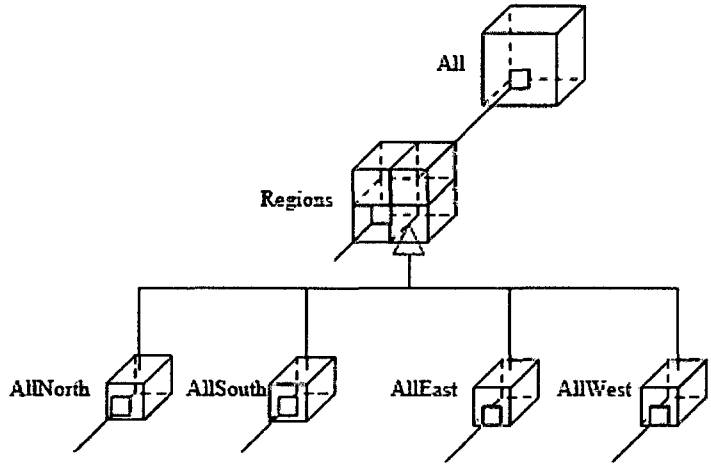


Figura 4.3: Especialización de un hecho sobre la base de una celda.

Fuente: GUNDERLOY MIKE – SNEATH TIM.

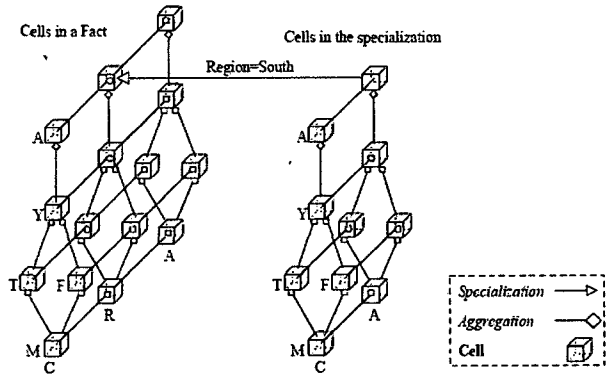


Figura 4.4: de un hecho de especialización por región.

Fuente: GUNDERLOY MIKE – SNEATH TIM.

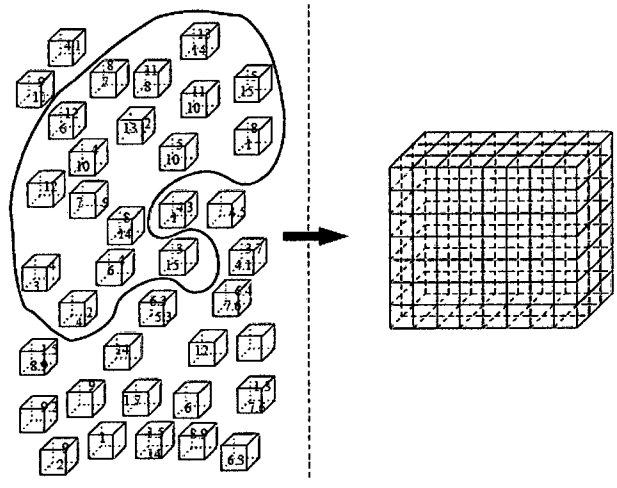


Figura 4.5: Esquema de una Celda de análisis independiente con tres dimensiones

Fuente: GUNDERLOY MIKE – SNEATH TIM.

CONCLUSIONES

- Es importante el modelo de datos multidimensionales para un mejor diseño de una Base de datos, esto debido a que la creación de un Data Warehouse previa a el desarrollo de los Data Marts, según la arquitectura planteada por Inmon, ayuda a que una institución tenga toda su información consolidada y ordenada en un solo lugar, lo cual es muy importante en este tipo de organizaciones debido a la sensibilidad e importancia de la información.
- Tener todos los datos consistentes y ordenados en el Data Warehouse brinda una fuente confiable y estandarizada para el desarrollo de futuros Data Marts o para la ampliación del alcance de los existentes, facilitando el desarrollo de estos.
- Se caracterizó el uso de Base de Datos con visión multidimensional utilizando modelo orientado a objetos.
- Se desarrollo mecanismos de abstracción proporcionados para definir requisitos iniciales de usuario y operaciones OLAP así como la posibilidad de modelar el comportamiento de dichos requisitos en función de las operaciones OLAP aplicadas.

- La aplicación eficientemente de modelo de datos multidimensionales en el proceso de diseño Base de Datos es muy necesario para el manejo de gran cantidad de datos.
- Se puede definir formalmente el uso de Base de Datos con visión multidimensional.

RECOMENDACIONES

- Algunos procesos, tanto en el ETL como la construcción de reportes, presentan casos particulares que son muy difíciles de solucionar y en los cuales la documentación no presenta una solución. Para tratar estos problemas, resulta muy útil consultar los foros de las herramientas mismas u otras páginas de Internet dedicadas a resolver consultas de este tipo.

- Resulta frecuente estas situaciones, y al ser una tecnología relativamente nueva, se crean foros de ayuda comunitaria en donde los usuarios comparten sus diversas experiencias. Se puede considerar a estos lugares como una fuente muy útil de información.

BIBLIOGRAFÍA

1. BORRUEL, FERNANDO Y MUÑOZ, MONICA (2001). Data Warehouse con Business Objects y Webintelligence. Editorial Anaya. España.
2. BUSSINESSOBJECTS. http://www.businessobjects.com. Ultimo acceso: agosto 2007
3. CATIBUSIC S, HADZAGIC-CATIBUSIC F, ZUBCEVIC S. (2004). Data warehousing as a generator of system component for decision support in health care. Esatdos unidos.
4. CHAKRABARTI K. Y MEHROTRA S. (1998). Dynamic granular locking approach to phantom protection in R-Trees. En ICDE'98: Proceedings of the Fourteenth International Conference on Data Engineering. USA.
5. CHAUDHURI S. Y DAYAL U. (1997). An overview of data warehousing and OLAP technology. México.
6. CHEN M.-S., HAN J. Y YU P. S. (1996). Data mining: An overview from a database perspective. IEEE Trans. Knowl. Estados unidos.
7. COGNOS & BUSINESS INTELLIGENCE. http://www.cognos.com. Ultimo acceso: agosto 2007
8. CONNOLLY, T., BEGG, C. E. (2005). Sistemas de Bases de Datos: Un enfoque practico para diseño, implementación y gestión. 4ª Ed., Pearson Eddison Wesley. España.

9. CONNOLLY, THOMAS (2002). Sistema de base de datos. Editorial Addison-Wesley Iberoamericana. España.
10. COREY MICHAEL – ABBEY MICHAEL (1997). Oracle, Data Warehousing. Editorial Osborne/McGraw-Hill. Madrid España.
11. DATASTAGE WEB PAGE. http://www-304.ibm.com/jct03002c/software/data/integration/datastage/. Ultimo acceso: agosto 2007.
12. DATE C. J. (2001). Introducción al Sistema de base de datos. Editorial Pearson educación. México.
13. DE MIGUEL CASTAÑO ADORACIÓN – MARTÍNEZ FERNÁNDEZ PALOMA (2001). Diseño de Base de Datos, Editorial AlfaOmega ra-ma, Universidad Carlos III de Madrid, España.
14. DE MIGUEL CASTAÑO ADORACIÓN – PIATINI Mario (1999). Fundamentos y modelos de Base de Datos, Editorial Alfaomega ra-ma 2da. Edición, Universidad Carlos III de Madrid, España.
15. DE MIGUEL CASTAÑO ADORACIÓN – PIATINI Mario (2000). Diseño de base de datos relacionales, Editorial Alfaomega ra-ma, Universidad Carlos III de Madrid, España.
16. FEGARAS, L., ELMASRI, R. (2001). Query Engines for Web-Accesible XML Data. Proceedings of the 27th VLDB Conference. Roma Italia.
17. FRANCO JEAN MICHEL (1997). El Data Warehouse. Editorial Eyrolles. Paris – Francia.

18. FRANCO, JEAN (1997). El Data Warehouse. Editorial: Ediciones Gestión 200s.a. España.
19. GOLFARELLI, M. MAIO, D. RIZZI, S. (1998). The Dimensional Fact Model: A Conceptual Model for Data Warehouses. International Journal of Cooperative Information Systems. Esatdos Unidos. Pág. 215 – 247.
20. GOLFARELLI, M. MAIO, D. RIZZI, S. (1999). Designing the Data Warehouse: Key Steps and Crucial Issues. Journal of Computer Science and Information Management, Maximilian Press Publisher. USA.
21. GUNDERLOY MIKE – SNEATH TIM (2001). SQL Server Developer´s Guide to OLAP with Analysis Services. Editorial Sybex. Estados Unidos.
22. GUZMAN JIMÉNES ROSARIO (2001). Base de datos Relacional. Fondo de desarrollo Editorial, Universidad de Lima. Perú.
23. GYSSENS, M. LAKSHMANAN, L. (1997). A Foundation for Multi-Dimensional Databases, VLDB, Athens, Greece. Estados Unidos.
24. HARJINDER S. GILL Y PRAKASH C. RAO (1996). Data Warehousing. Editorial Hall Hispano americano. México
25. HERNANDEZ ORALLO JOSE – RAMIREZ QUINTANA JOSE (2004). Introducción a minería de datos. Editorial Pearson. Madrid, España.
26. HIDALGO ESPINAQUE, MAURICIO (2001). Data Warehouse & Olap Developer. Editorial Cibertec. Perú.
27. IBM (1999). Fundamentals of Data Warehouse and Business Intelligence for Knowledge Management – Instructor Guide. Estados Unidos. Pág. 48

28. INMON W. H. (1999). Building the operational data store. Edition wiley. Estados Unidos.

29. LAUDON KENNETH C.; LAUDON JANE PRICE (2001). Sistemas de información gerencial: administración de la empresa digital. Ediciones Pearson Educación. España.

30. LI, C., WANG (1996). Data mining and machine learning for test, diagnosis, validation and verification. Estados Unidos.

31. LUQUE RUIZ IRENE, GOMEZ MIGUEL ANGEL (2002). Base de datos desde Chen hasta Codd con Oracle. Editorial Alfaomega Ra-ma. Córdova España.

32. LUQUE RUIZ MIGUEL - GOMEZ MIGUEL ANGEL (2002). Base de datos, Editorial Alfaomega ra-ma, Universidad de Córdova. España.

33. MANNINO MICHAEL V. (2007). Administración de base de datos. Diseño y desarrollo de aplicaciones. Editorial McGraw Hill. Tercera edición. México. Pag. 553 – 603

34. MICROSOFT (2000). Data Warehousing with SQL Server 7.0 Technical Reference. Editorial Dennis Peterson. Estados Unidos.

35. MICROSOFT (2001). Designing and Implementing OLAP Solutions with Microsoft SQL Server 2000. Ediciones Cargraphics. Estados Unidos.

36. MICROSTRATEGY BUSINESS INTELLIGENCE SOLUTIONS. http://www.microstrategy.com. Ultimo acceso: agosto 2007.

37. MOSS LARISSA, ATRE SHAKU (2003). Business Intelligence Roadmap.

The Complete Project Lifecycle for Decision-Support Applications. EE.UU.

38. PALMA CLAUDIO Y PEREZ RICHARD (2009). Data Mining. Editorial Ril Editores. Chile.
39. PIATTINI M. G., MARCOS, E., CALERO, C., VELA, B. (2006). Tecnología y Diseño de Bases de Datos. Ra-Ma, España.
40. PIATTINI MARIO – CALVO JOSE (2000). Análisis y Diseño detallado de Aplicaciones Informáticas de Gestión, Editorial Alfaomega ra-ma. España.
41. RAMAKRISHNAN RAGHU – GEHRKE JOHANNES (2007). Sistema de Gestión de Bases de Datos. Editorial McGraw-Hill, España. Pág. 513.
42. RAMEZ A. ELMASRI, SHAMKANT B. NAVATHE (2002). Fundamentos de Sistemas de Base de datos. Editorial Addison Wesley. España.
43. ROB PETER, CORONEL CARLOS (2004). Sistema de Base de Datos, Diseño, implementación y administración. Ediciones Thomson. México.
44. RUMBAUGH, J., BLAHA, M., PREMERLANI, W., EDDY, F., & LORENSEN, F. (1991). Object-Oriented Modeling and Design. Editorial Prentice – Hall. Estados Unidos.
45. SILBERSCHATZ, ABRAHAM (2006). Fundamentos de Base de datos. Ediciones McGraw-Hill / Interamericana, quinta edición. España.
46. SOLID QUALITY LEARNING (2007). Bases de datos con SQL Server 2005. **Editores** ANAYA MULTIMEDIA. España.

47. SQL SERVER INTEGRATION SERVICES (SSIS) Web Page. http://www.microsoft.com/sql/technologies/integration/default.mspx Último acceso: agosto 2007.

48. SUNOPSIS Web Page. http://www.sunopsis.com/corporate/index.htm Último acceso: agosto 2007.

49. THOMSON PARANINFO (2005). Introducción a las bases de datos: EL MODELO RELACIONAL. Editorial Provisional. España.

50. TRUJILLO, J., PALOMAR, M. (2001). Designing Data Warehouses with OO Conceptual Models. Estados Unidos.

51. VARA. J. M., VELA. B., MARCOS, E. (2000). Transformaciones de modelos para el desarrollo de base de datos. Grupo Kybele. España.

52. VERCELLIS CARLO (2008). Business Intelligence. Editorial Wiley. España.

53. VINCENT RAINARDI (2000). Building a Data Warehouse With Examples in SQL Server. Editorial Apress. Estados Unidos.

54. VITT ELIZABETH, LUCKEVICH MICHAEL, MISNER STACIA (2002). Business Intelligence. Editorial McGraw-Hill. España.

55. WU, CH., BUCHMANN, P. (1997). Research issues in data warehousing Datebanksysteme in Büro, Technik und Wissenchaft (BTW), Informatik Aktuell, Pág. 61- 62. Estados Unidos.

ANEXO

Desarrollo de un caso de estudio

Para la comparación y el análisis del rendimiento de los tipos de almacenamiento ROLAP y MOLAP se realizó un caso de estudio en el SGBD SQL Server con su componente Microsoft Analysis Services.

Caso de estudio: ventas

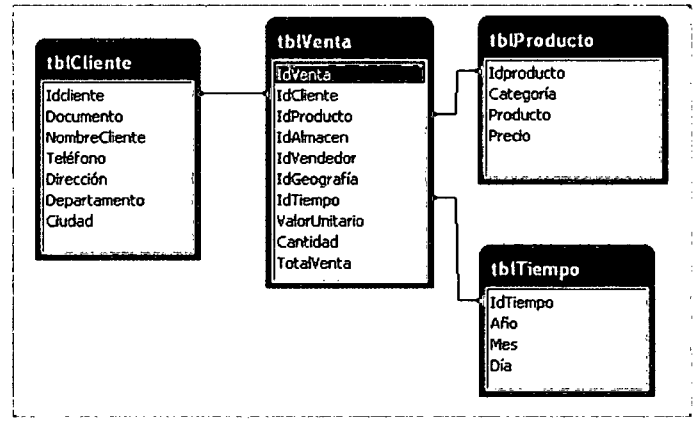
El caso de estudio realizado consiste en las ventas de productos de una cadena de almacenes que se encuentran en varias ciudades del país.

Antes de diseñar los cubos es necesario establecer una BD OLAP. Esta es similar a una BD SQL Server relacional, sin embargo la última contiene tablas relacionales y vistas, mientras que una BD OLAP contiene cubos multidimensionales, dimensiones, orígenes de datos y otros objetos.

En la BD OLAP seleccionada se crearon seis dimensiones: almacén, cliente, geografía, producto, vendedor y tiempo. Se crearon ocho cubos, cuatro con tipo de almacenamiento MOLAP y cuatro con tipo de almacenamiento ROLAP en forma correspondiente.

Cubo comportamiento *clientes*

La tabla de hechos es la de ventas, y las dimensiones son clientes, producto y tiempo, como se muestra en la Figura A.1. Este cubo muestra la información sobre el comportamiento de los clientes a través del tiempo en cuanto a la compra de productos.

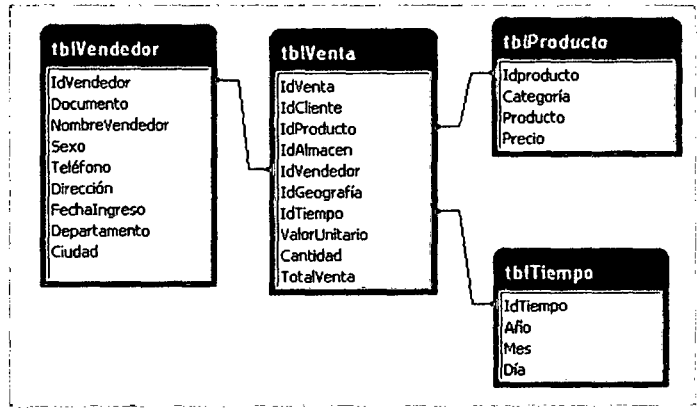


Las dimensiones almacén, vendedor y geografía no se tiene en cuenta para la construcción de este cubo.

Figura A.1. Cubo comportamiento clientes

Cubo desempeño *vendedores*

La tabla de hechos es la de ventas y las dimensiones son vendedor, producto y tiempo, como se muestra en la Figura A.2. Este cubo permite estudiar y evaluar cómo ha sido el desempeño de cada vendedor a través del tiempo en cuanto a la venta de productos.

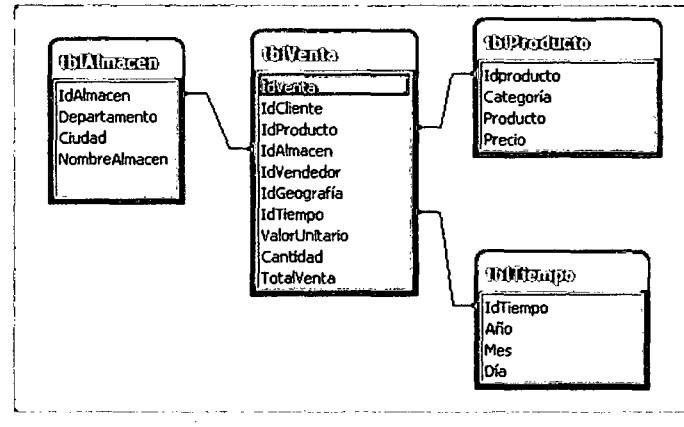


Las dimensiones cliente, almacén, y geografía no se tiene en cuenta para la construcción de este cubo.

Figura A.2. Cubo desempeño vendedores

Cubo ventas *por almacén*

La tabla de hechos es la de ventas y las dimensiones son almacén, producto y tiempo, como se muestra en la Figura A.3. Este cubo se creó con el fin de analizar las ventas de productos de cada almacén a través del tiempo.



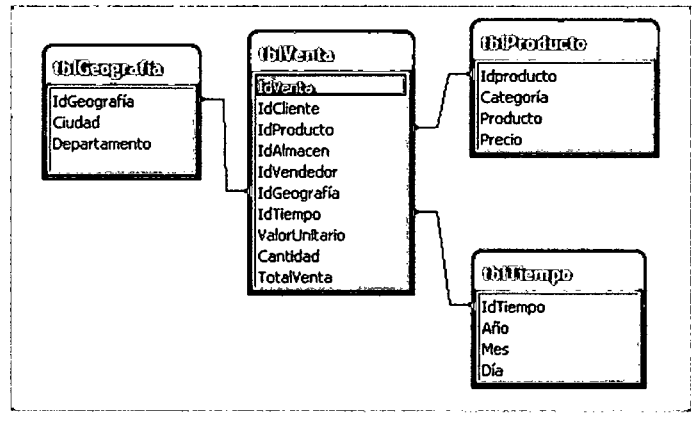
Las dimensiones cliente, vendedor, y geografía no se tiene en cuenta para la construcción de este cubo.

Figura A.3. Cubo Ventas por almacén

Cubo ventas por geografía

La tabla de hechos es la de ventas y las dimensiones son geografía, producto y tiempo, como se muestra en la

Figura A.4. Este cubo permite un análisis a través del tiempo acerca de las variaciones registradas en las ventas de productos en los distintos niveles geográficos (ciudad, departamento, país) donde hay almacenes.



Las dimensiones cliente, vendedor, y Almacen no se tiene en cuenta para la construcción de este cubo.

Figura A.4. Cubo Ventas por geografía

Generación de los informes y análisis de los datos

Para la presentación y análisis de los datos se utilizaron tres herramientas, las cuales permiten filtrar los datos

según las dimensiones, aplicar operaciones para ver los datos más agregados (*Drill Up*) o para ver los datos más detallados (*Drill Down*):

- Cube Browser incluido en Analysis Services

- Excel 2003, junto con un servidor SQL Server OLAP Server y un cliente motor de cálculo y almacenamiento en caché denominado Microsoft PivotTable Service se logra el análisis de los cubos (Microsoft Corporation, 2003)

- MDX (Multidimensional Expressions), el cual es un lenguaje usado para consultar una BD OLAP en SQL Server 2000 Analysis Services (Microsoft Corporation 2, 2004), de la misma forma en que se usa SQL para consultar una BD SQL Server. Comparte con SQL la misma estructura básica, pero tiene algunas características adicionales creadas especialmente para manipular este tipo de BD.

Tamaño de la BD OLAP utilizada

La tabla de hechos y las dimensiones de la BD OLAP utilizada tienen la siguiente cantidad de registros. Tabla de hechos Venta: 124.049, Dimensión Cliente: 65.000, Dimensión Producto: 28, Dimensión Vendedor: 39, Dimensión Almacén: 20, Dimensión Geografía: 11 y Dimensión Tiempo: 365.

Informes

Se generó un informe por cada cubo. A cada informe se le aplicaron las diferentes operaciones como: *Slice & Dice, Drill Down, Drill Up*, entre otros, para observar el comportamiento de los datos y analizar el tiempo de respuesta de cada informe.

Informe 1 de comportamiento de clientes

Se filtraron los clientes (dimensión cliente) de un departamento específico agrupados por categoría (dimensión producto) y por día (dimensión tiempo).

Informe 2 de comportamiento de clientes

Se filtraron los clientes (dimensión cliente) de dos departamentos específicos agrupados por categoría (dimensión producto) y por año (dimensión tiempo).

Informe de desempeño de vendedores

Informe de todos los vendedores (dimensión vendedor) agrupados por producto (dimensión producto) y por día (dimensión tiempo).

Informe de ventas por almacén

Informe de todos los almacenes (dimensión almacén) agrupados por producto (dimensión producto) y por día (dimensión tiempo).

Informe de ventas por geografía

Informe de todas las ciudades (dimensión geografía) agrupadas por producto (dimensión producto) y por día (dimensión tiempo).

La Tabla resume los resultados obtenidos de los cinco informes.

Tabla Tiempos de respuesta de los informes

Nombre del Informe	**Tiempo de respuesta MOLAP**	**Tiempo de respuesta ROLAP**
Informe 1 de comportamiento de clientes	14 s	20 s
Informe 2 de comportamiento de clientes	7 s	11 s
Informe de desempeño de vendedores	2 s	6 s
Informe de ventas por almacén	5 s	10 s