Capítulo 3
RECOPILACIÓN. ALMACENES DE DATOS

Para poder comenzar a analizar y extraer algo útil de los datos es preciso, en primer lugar, disponer de ellos. Esto en algunos casos puede parecer trivial; se parte de un simple archivo de datos que analizar. En otros, la diversidad y tamaño de las fuentes hace que el proceso de recopilación de datos sea una tarea compleja, que requiere una metodología y una tecnología propias. En general, el problema de reunir un conjunto de datos que posibilite la extracción de conocimiento requiere decidir, entre otros aspectos, de qué fuentes, internas y externas, se van a obtener los datos, cómo se van a organizar, cómo se van a mantener con el tiempo y, finalmente, de qué forma se van a poder extraer parcial o totalmente, en detalle o agregados, con distintas "vistas minables " a las que podamos aplicar las herramientas concretas de minería de datos.

En este capítulo[3] nos centramos en las 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*). Los almacenes de datos no son estrictamente necesarios para realizar minería de datos, aunque sí extremadamente útiles si se va a trabajar con grandes volúmenes de datos, que varían con el tiempo y donde se desea realizar tareas de minerías de datos variadas, abiertas y cambiantes. 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. Finalmente, la selección, transformación y limpieza de datos serán tratadas en el capítulo siguiente, aunque algunas de estas operaciones puedan hacerse antes o durante el proceso de recopilación e integración.

[3] Para la comprensión de este capítulo se asumen unos conocimientos básicos de bases de datos, especialmente bases de datos relacionales. Un buen libro introductorio es [Celma et al. 2003].

3.1 Introducción

Como vimos en el capítulo anterior, el primer paso en el proceso de extracción de conocimiento a partir de datos es precisamente reconocer y reunir los datos con los que se va a trabajar. Si esta recopilación se va a realizar para una tarea puntual y no involucra muchas cantidades y variedades de datos simples, es posible que el sentido común sea suficiente para obtener un conjunto de datos con la calidad suficiente para poder empezar a trabajar. En cambio, si requerimos datos de distintas fuentes, tanto externas como internas a la organización, con datos complejos y variados, posiblemente en grandes cantidades y además cambiantes, con los que se desee realizar a medio o largo plazo diversas tareas de minería de datos, es posible que nuestro sentido común no sea suficiente para hacer una recopilación e integración en condiciones.

Al igual que la tecnología de bases de datos ha desarrollado una serie de modelos de datos (como el relacional), de lenguajes de consulta y actualización, de reglas de actividad, etc., para trabajar con la información transaccional de una organización, veremos que existe una tecnología relativamente reciente, denominada "almacenes de datos " (*data warehouses*) que pretende proporcionar metodologías y tecnología para recopilar e integrar los datos históricos de una organización, cuyo fin es el análisis, la obtención de resúmenes e informes complejos y la extracción de conocimiento. Esta tecnología está diseñada especialmente para organizar grandes volúmenes de datos de procedencia generalmente estructurada (bases de datos relacionales, por ejemplo), aunque el concepto general es útil para la organización de pequeños conjuntos de datos en aplicaciones de minería de datos más modestas.

Supóngase que en una compañía bien implantada en el ámbito europeo queremos analizar aquellos países y gamas de productos en los que las ventas vayan excepcionalmente bien (con el objetivo, por ejemplo, de premiar a las oficinas comerciales de cada gama y producto) o, dicho de una manera más técnica, averiguar si la penetración relativa (teniendo en cuenta la permeabilidad del país en cuestión) de una gama de productos es significativamente mayor que la media de penetración en el conjunto del continente. La compañía dispone, por supuesto, de una base de datos transaccional sobre la que operan todas las aplicaciones de la empresa: producción, ventas, facturación, proveedores, nóminas, etc. Lógicamente, de cada venta se registra la fecha, la cantidad y el comprador y, de éste, el país. Con toda esta información histórica nos podemos preguntar: ¿es esta información suficiente para realizar el análisis anterior? La respuesta, a primera vista, quizá de manera sorprendente, es negativa. Pero, aparentemente, si tenemos detalladas las ventas de tal manera que una consulta SQL puede calcular las ventas por países de todos los productos y gamas, ¿qué más puede hacer falta?

Sencillamente, la respuesta hay que buscarla fuera de la base de datos, en el contexto donde se motiva el análisis. La penetración de un producto depende de las ventas *por habitante*. Si no tenemos en cuenta la población de cada país la respuesta del análisis estará sesgada; será muy probable que entre los países con mayor penetración siempre esté Alemania, y entre los países con menor penetración se encuentre San Marino. Pero no sólo eso, es posible que, si deseamos hacer un análisis más perspicaz, nos interese saber la renta per cápita de cada país o incluso la distribución por edad de cada país. Dependiendo de la gama, nos puede interesar información externa verdaderamente específica. Por ejemplo, las

horas de sol anuales de cada país pueden ser una información valiosísima para una compañía de cosméticos. Lógicamente es más difícil vender bronceadores en Lituania que en Grecia o, dicho más técnicamente, Lituania tiene menos permeabilidad a la gama de bronceadores que Grecia. Pero este hecho, que nos parece tan lógico, sólo podrá ser descubierto por nuestras herramientas de minería de datos si somos capaces de incorporar información relativa a las horas de sol o, al menos, cierta información climática de cada país.

Evidentemente, cada organización deberá recoger diferente información que le pueda ser útil para la tarea de análisis, extracción de conocimiento y, en definitiva, de toma de decisiones. En la Figura 3.1 se muestran las fuentes de datos que pueden ser requeridas en el caso anterior para un proceso de extracción de conocimiento satisfactorio. Sólo conociendo el contexto de cada organización o de cada problema en particular se puede determinar qué fuentes externas van a ser necesarias. Además, este proceso es generalmente iterativo. A medida que se va profundizando en un estudio, se pueden ir determinando datos externos que podrían ayudar y se pueden ir añadiendo a nuestro "repositorio de datos". Por tanto, la tarea de mantener un "repositorio" o un "almacén" con toda la información necesaria cobra mayor relevancia y complejidad.

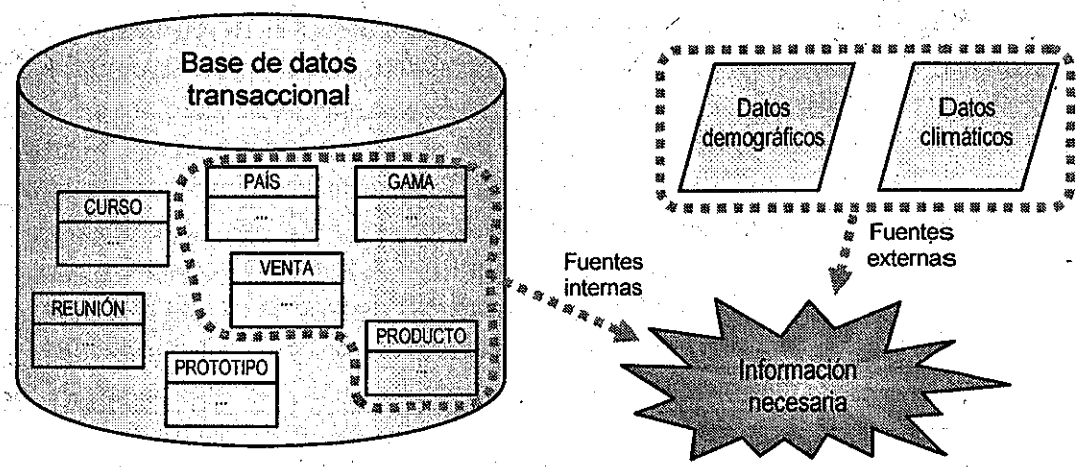


Figura 3.1. Fuentes de datos requeridas para responder "países con mayor penetración de bronceadores".

El mantenimiento de esta información plantea cuestiones técnicas. En primer lugar, se requerirá añadir, puede que frecuentemente, nueva información a nuestro repositorio, tanto proveniente de actualizaciones de la propia organización como de fuentes externas, ya sean actualizaciones como nuevas incorporaciones. En segundo lugar, y la que resulta la cuestión principal, ¿hay que almacenar toda esta información en la base de datos transaccional? Puestos en el ejemplo anterior, ¿requieren las aplicaciones diarias de la organización almacenar en una tabla de la base de datos la temperatura media de Lituania?

Estas y otras cuestiones, como veremos a continuación, han motivado el desarrollo de una tecnología nueva y específica, denominada "almacenes de datos" (*data warehouses*[4]).

[4] La traducción en castellano no es unánime. En partes de Latinoamérica se conocen también como "bodegas de datos".

3.2 Necesidad de los almacenes de datos

La proliferación de sistemas de información sustentados en bases de datos ha generalizado el uso de herramientas que permiten obtener informes complejos, resúmenes e incluso estadísticas globales sobre la información almacenada con el objetivo de asistir en la toma de decisiones. La mayoría de sistemas comerciales de gestión de bases de datos incluyen herramientas de "informes avanzados", "inteligencia de negocio" (*business intelligence*), sistemas de información ejecutivos (EIS, *Executive Information Systems*) y otras, que pese a sus nombres variados intentan realizar un procesamiento analítico de la información, más que el procesamiento transaccional habitual realizado por las aplicaciones del día a día de la organización.

Por tanto, cada día es más necesario distinguir dos usos diferentes del sistema de información: el procesamiento transaccional y el procesamiento analítico.

3.2.1 OLTP y OLAP

Con las siglas OLTP y OLAP se denominan dos tipos de procesamiento bien diferentes:

- **OLTP (*On-Line Transactional Processing*).** El procesamiento transaccional en tiempo real constituye el trabajo primario en un sistema de información. Este trabajo consiste en realizar transacciones, es decir, actualizaciones y consultas a la base de datos con un objetivo operacional: hacer funcionar las aplicaciones de la organización, proporcionar información sobre el estado del sistema de información y permitir actualizarlo conforme va variando la realidad del contexto de la organización. Muestras de este tipo de trabajo transaccional son, por ejemplo, en el caso de una empresa, la inserción de un nuevo cliente, el cambio de sueldo de un empleado, la tramitación de un pedido, el almacenamiento de una venta, la impresión de una factura, la baja un producto, etc. Es el trabajo diario y para el que inicialmente se ha diseñado la base de datos.
- **OLAP (*On-Line Analytical Processing*).** El procesamiento analítico en tiempo real engloba un conjunto de operaciones, exclusivamente de consulta, en las que se requiere agregar y cruzar gran cantidad de información. El objetivo de estas consultas es realizar informes y resúmenes, generalmente para el apoyo en la toma de decisiones. Ejemplos de este tipo de trabajo analítico pueden ser resúmenes de ventas mensuales, los consumos eléctricos por días, la espera media de los pacientes en cirugía digestiva de un hospital, el producto cuyas ventas han crecido más en el último trimestre, las llamadas por horas, etc. Este tipo de consultas suelen emanarse de los departamentos de dirección, logística o prospectiva y requieren muchos recursos.

Una característica de ambos procesamientos es que se pretende que sean "on-line", es decir, que sean relativamente "instantáneos" y se puedan realizar en cualquier momento (en tiempo real). Esto parece evidente e imprescindible para el OLTP, pero no está tan claro que esto sea posible para algunas consultas muy complejas realizadas por el OLAP.

La práctica general, hasta hace pocos años, y todavía existente en muchas organizaciones y empresas, es que ambos tipos de procesamiento (OLTP y OLAP) se realizaran sobre la misma base de datos transaccional. De hecho, una de las máximas de la tecnología de

base de datos era la eliminación de redundancia, con lo que parecía lo más lógico que ambos procesamientos trabajaran sobre una única base de datos general (aunque pudiera tener diferentes vistas para diferentes aplicaciones, procesamientos o servicios).

Esta práctica plantea dos problemas fundamentales:

- Las consultas OLAP perturban el trabajo transaccional diario de los sistemas de información originales. Al ser consultas complejas y que involucran muchas tablas y agrupaciones, suelen consumir gran parte de los recursos del sistema de gestión de base de datos. El resultado es que durante la ejecución de estas consultas, las operaciones transaccionales normales (OLTP), se resienten: las aplicaciones van más lentas, las actualizaciones se demoran muchísimo y el sistema puede incluso llegar a colapsarse. De este hecho viene el nombre familiar que se les da a las consultas OLAP: *"killer queries"* (consultas asesinas). Como consecuencia, muchas de estas consultas se deben realizar por la noche o en fines de semana, con lo que en realidad dejan de ser "on-line".
- La base de datos está diseñada para el trabajo transaccional, no para el análisis de los datos. Esto significa que, aunque tuviéramos el sistema dedicado exclusivamente para realizar una consulta OLAP, dicha consulta puede requerir mucho tiempo, pero no sólo por ser compleja intrínsecamente, sino porque el esquema de la base de datos no es el más adecuado para este tipo de consultas.

Ambos problemas implican que va a ser prácticamente imposible (a un coste de *hardware* razonable, lógicamente) realizar un análisis complejo de la información en tiempo real si ambos procesamientos se realizan sobre la misma base de datos.

Afortunadamente, debido a que los costes de almacenamiento masivo y conectividad se han reducido drásticamente en los últimos años, parece razonable recoger (copiar) los datos en un sistema unificado y diferenciado del sistema tradicional transaccional u. operacional. Aunque esto vaya contra la filosofía general de bases de datos, son muchas más las ventajas que los inconvenientes, como veremos a continuación. Desde esta perspectiva, se separa definitivamente la base de datos con fines transaccionales de la base de datos con fines analíticos. Nacen los almacenes de datos.

3.2.2 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 tiempo real (OLAP).
- No disturba el OLTP de las bases de datos originales.

A partir de ahora, por tanto, diferenciaremos claramente entre bases de datos transaccionales (u operacionales) y almacenes de datos. Dicha diferencia, además, se ha ido marcando más profundamente a medida que las tecnologías propias de ambas bases de datos (y en

especial la de almacenes de datos) se han ido especializando. De hecho, hoy en día, las diferencias son claras, como se muestra en la Tabla 3.1.

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

	BASE DE DATOS TRANSACCIONAL	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, pivot...*). Lectura.

Tabla 3.1. Diferencias entre la base de datos transaccional y el almacén de datos.

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 3.2.

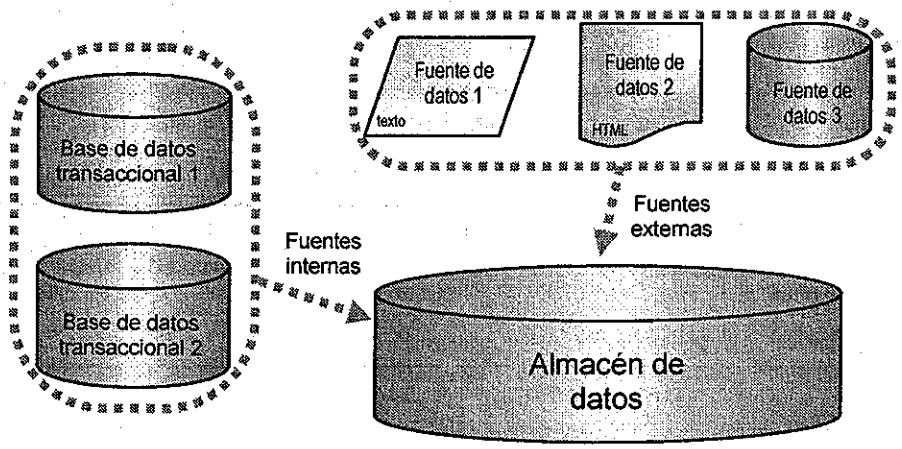


Figura 3.2. El almacén de datos como integrador de diferentes fuentes de datos.

La organización y el mantenimiento de esta información plantea cuestiones técnicas, fundamentalmente sobre cómo diseñar el almacén de datos, cómo 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.

3.3 Arquitectura de los almacenes de datos

Un almacén de datos recoge, fundamentalmente, datos históricos, es decir, *hechos*, sobre el contexto en el que se desenvuelve la organización. Los hechos son, por tanto, el aspecto central de los almacenes de datos. Esta característica determina en gran medida la manera de organizar los almacenes de datos.

3.3.1 Modelo multidimensional

El modelo *conceptual* de datos más extendido para los almacenes de datos es el **modelo multidimensional**. Los datos se organizan en torno a los *hechos*, que tienen unos atributos o *medidas* que pueden verse en mayor o menor detalle según ciertas *dimensiones*. Por ejemplo, una gran cadena de supermercados puede tener como hechos básicos las *ventas*. Cada venta tiene unas medidas: importe, cantidad, número de clientes, etc., y se puede detallar o agregar en varias dimensiones: tiempo de la venta, producto de la venta, lugar de la venta, etc. Es esclarecedor comprobar que las medidas responden generalmente a la pregunta "cuánto", mientras que las dimensiones responderán al "cuándo", "qué", "dónde", etc.

Lo realmente interesante del modelo es que ha de permitir, de una manera sencilla, obtener información sobre hechos a diferentes niveles de agregación. Por ejemplo, el hecho "El día 20 de mayo de 2003 la empresa vendió en España 12.327 unidades de productos de la categoría *insecticidas*" representa una medida (cantidad, 12.327 unidades) de una venta con granularidad día para la dimensión tiempo (20 de mayo de 2003), con granularidad país para la dimensión lugar (España) y con granularidad categoría (insecticidas) para la dimensión de productos. Del mismo modo, el hecho "El primer trimestre de 2004 la empresa vendió en Valencia por un importe de 22.000 euros del producto *Androbrío* 33 cl." representa una medida (importe, 22.000 euros) de una venta con granularidad trimestre para la dimensión tiempo (primer trimestre de 2004), con granularidad ciudad para la dimensión lugar (Valencia) y con granularidad artículo (*Androbrío* 33 cl.) para la dimensión de productos.

En la Figura 3.3 se representa parte de un almacén de datos con estructura multidimensional de donde se pueden extraer estos dos hechos.

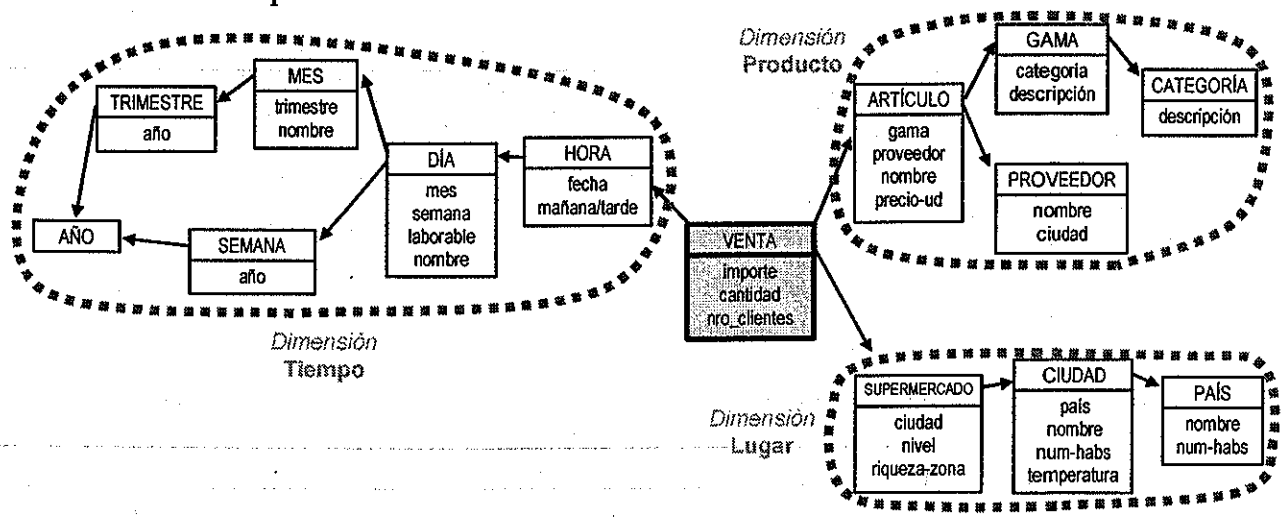


Figura 3.3. Información sobre ventas en un almacén de datos representado bajo un modelo multidimensional.

La Figura 3.3 no se basa en ningún modelo de datos en particular (por ejemplo el relacional). Nótese que no estamos hablando de que cada rectángulo de la figura sea una tabla o que las flechas sean claves ajenas. Al contrario, simplemente estamos representando datos de una manera conceptual. Mostramos los hechos "venta" y tres dimensiones con varios niveles de agregación. Las flechas se pueden leer como "se agrega en". Como se observa en la figura, cada dimensión tiene una estructura jerárquica pero no necesariamente lineal. Por ejemplo, en las dimensiones *tiempo* y *producto* hay más de un camino posible de agregación (ruta de agregación). Incluso, en el caso de los productos, el nivel de agregación mayor puede ser diferente (hacia categoría o hacia proveedor). Esto permite diferentes niveles y caminos de agregación para las diferentes dimensiones, posibilitando la definición de hechos agregados con mucha facilidad. La forma que tienen estos conjuntos de hechos y sus dimensiones hace que se llamen popularmente almacenes de datos en "**estrella simple**" (cuando no hay caminos alternativos en las dimensiones) o de "estrella jerárquica" o "**copo de nieve**" (cuando sí hay caminos alternativos en las dimensiones, como el ejemplo anterior).

Cuando el número de dimensiones no excede de tres (o se agregan completamente el resto) podemos representar cada combinación de niveles de agregación como un cubo. El cubo está formado por casillas, con una casilla para cada valor entre los posibles para cada dimensión a su correspondiente nivel de agregación. Sobre esta "vista", cada casilla representa un hecho. Por ejemplo, en la Figura 3.4 se representa un cubo tridimensional donde las dimensiones producto, lugar y tiempo se han agregado por artículo, ciudad y trimestre. La representación de un hecho como el visto anteriormente corresponde, por tanto, a una casilla en dicho cubo. El valor de la casilla es la medida observada (en este caso, importe de las ventas).

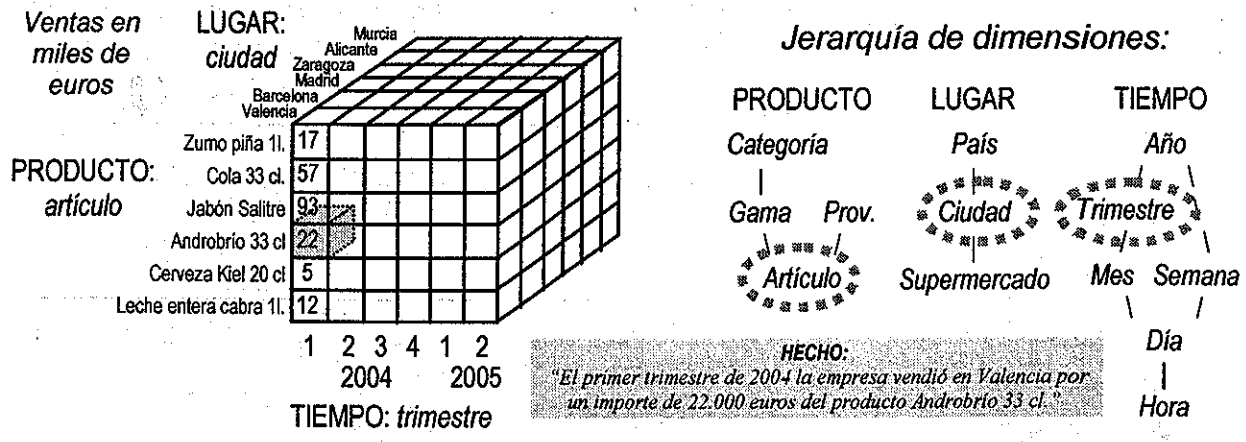


Figura 3.4. Visualización de un hecho en un modelo multidimensional.

Esta visualización hace que, incluso cuando tengamos más de tres dimensiones, se hable de un "cubo" (o más propiamente de "hipercubo") como un conjunto de niveles de agregación para todas las dimensiones.

Esta estructura permite ver de una manera intuitiva la sumarización/agregación (varias casillas se fusionan en casillas más grandes), la disgregación (las casillas se separan en casillas con mayor detalle) y la navegación según las dimensiones de la estrella.

3.3.2 Datamarts

En algunos casos puede parecer intuitivo organizar la información en dimensiones. El caso de las ventas es el ejemplo más ilustrativo. En general, cierta información es más fácilmente representable de esta forma, pero siempre se puede llegar a una estructura de este tipo. Lo que no es posible, en general, es la representación de todo el almacén de datos como una sola estrella, ni siquiera jerárquica. Por ejemplo, la información de personal de una empresa (empleados, departamentos, proyectos, etc.) es difícilmente integrable en la misma estrella que las ventas. Incluso, en ámbitos más relacionados de una organización (por ejemplo ventas y producción) esto tampoco es posible. La idea general es que para cada subámbito de la organización se va a construir una estructura de estrella. Por tanto, 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, los 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" (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 los demás. La única dimensión que suele aparecer en todos los datamarts es la dimensión *tiempo*, ya que el almacén de datos representa información histórica y, por tanto, siempre es de interés ser capaz de agregarlo por intervalos de diferente detalle.

En la Figura 3.5 se muestra un almacén de datos compuesto de varios datamarts.

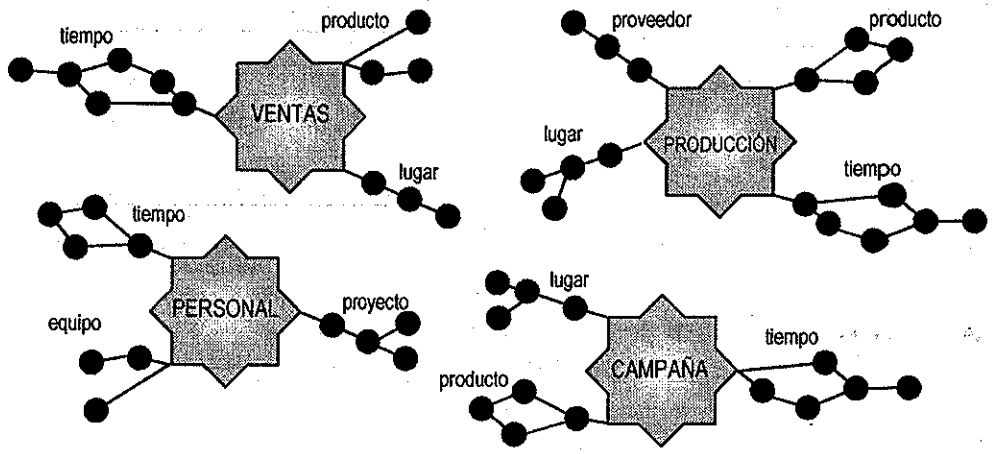


Figura 3.5. Representación icónica de un almacén de datos compuesto por varios datamarts.

Aparentemente, da la impresión de que el almacén de datos puede contener mucha información redundante, especialmente sobre las dimensiones. Aunque, en general, los almacenes de datos contienen información redundante, la estructura anterior es la estructura externa, visible o conceptual. Esta estructura no determina la manera de implementarlo ni lógica ni físicamente, como veremos.

3.3.3 Explotación de un almacén de datos. Operadores

En realidad, un modelo de datos se compone de unas estructuras y unos operadores sobre dichas estructuras. Acabamos de ver que el modelo multidimensional se basa en un conjunto de datamarts, que, generalmente, son estructuras de datos en estrella jerárquica. Para completar el modelo multidimensional debemos definir una serie de operadores sobre la estructura. Los operadores más importantes asociados a este modelo son:

- ***Drill***: se trata de disgregar los datos (mayor nivel de detalle o desglose, menos sumarización) siguiendo los caminos de una o más dimensiones.
- ***Roll***: se trata de agregar los datos (menor nivel de detalle o desglose, más sumarización o consolidación) siguiendo los caminos de una o más dimensiones.
- ***Slice & Dice***: se seleccionan y se proyectan datos.
- ***Pivot***: se reorientan las dimensiones.

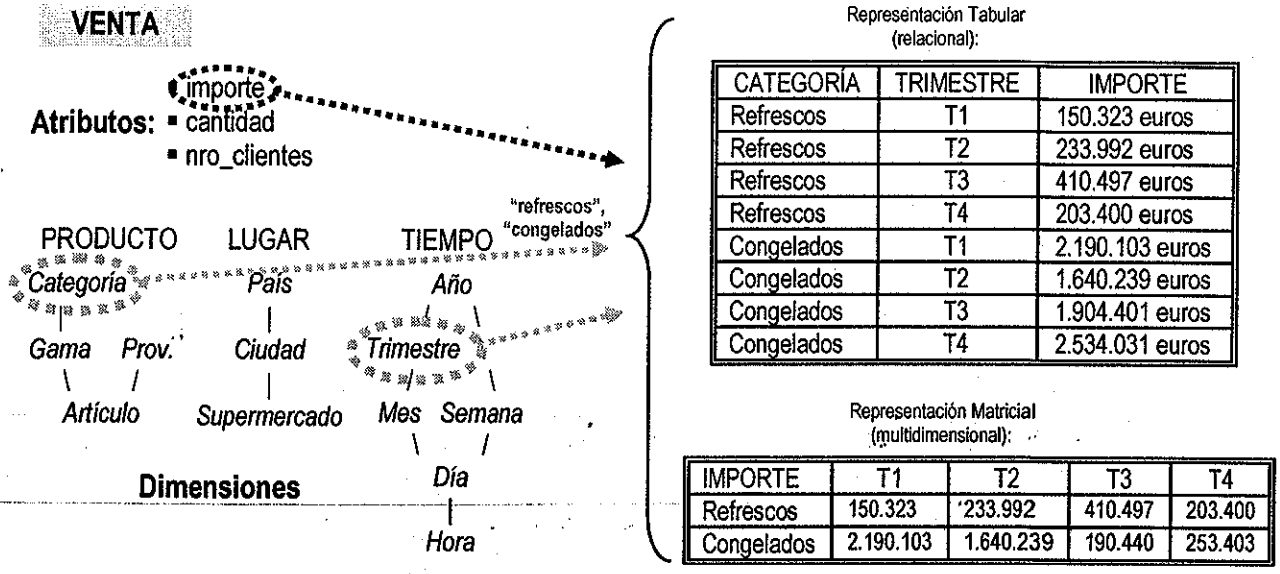
Normalmente, estos operadores se llaman operadores OLAP, operadores de análisis de datos u operadores de almacenes de datos. Para explicar estos operadores hemos de pensar que partimos, además, de unos operadores genéricos básicos, que permiten realizar consultas, vistas o informes sobre la estructura estrella, generalmente de una forma gráfica. Estos operadores básicos permiten realizar las mismas consultas de proyección, selección y agrupamiento que se pueden hacer en SQL. En muchos casos, de hecho, se puede editar la consulta SQL correspondiente, aunque ésta se haya hecho gráficamente.

Por tanto, el primer paso para poder utilizar los operadores propios del modelo multidimensional es definir una consulta. En realidad, como veremos a continuación, los operadores *drill, roll, slice & dice* y *pivot*, son modificadores o refinadores de consulta y sólo pueden aplicarse sobre una consulta realizada previamente.

Consideremos por ejemplo la consulta "obtener para cada categoría y trimestre el total de ventas" para el datamart de la Figura 3.3. En un entorno gráfico, dicha consulta se podría realizar eligiendo el nivel "categoría" para la dimensión "producto" (obteniendo además sólo dos categorías: "refrescos" y "congelados"), el nivel "trimestre" para la dimensión "tiempo" y no escogiendo la dimensión "lugar" (o considerando que se considera el nivel más agregado, es decir, todo el datamart). Además, se elegiría la propiedad que se desea ("importe").

Dependiendo del sistema o de la manera que hayamos elegido, el resultado se nos puede mostrar de manera tabular o de manera matricial, como se observa en la Figura 3.6, aunque la información mostrada es la misma. En este caso, como hay pocas dimensiones, la representación matricial parece más adecuada.

La existencia de dimensiones y atributos facilita, en gran medida, la realización de consultas y éstas se suelen hacer arrastrando con el ratón las medidas y dimensiones deseadas. No es necesario mucho más para realizar informes sencillos sobre ese datamart. No obstante, lo interesante empieza justamente cuando intentamos modificar el informe (una consulta, al fin y al cabo). A veces, querremos mayor nivel de detalle, otras veces menos, o bien desearemos añadir o quitar alguna dimensión, o modificar el informe en cualquier otro sentido.



CATEGORÍA	TRIMESTRE	IMPORTE
Refrescos	T1	150.323 euros
Refrescos	T2	233.992 euros
Refrescos	T3	410.497 euros
Refrescos	T4	203.400 euros
Congelados	T1	2.190.103 euros
Congelados	T2	1.640.239 euros
Congelados	T3	1.904.401 euros
Congelados	T4	2.534.031 euros

IMPORTE	T1	T2	T3	T4
Refrescos	150.323	233.992	410.497	203.400
Congelados	2.190.103	1.640.239	190.440	253.403

Figura 3.6. Construcción de una consulta seleccionando niveles de dimensiones.

Por ejemplo, supongamos que queremos ahora ver sólo las ventas de refrescos y desglosarlo por ciudades (en particular por dos: Valencia y León) con el objetivo de ver si los hábitos estacionales (hay más consumo de refrescos en estaciones calurosas) son generales en todas las áreas geográficas. Una forma obvia de hacer esto sería realizar un nuevo informe. Lo interesante de los nuevos operadores *drill, roll, slice & dice* y *pivot*, es que permiten modificar la consulta realizada, sin necesidad de realizar otra. En realidad son "navegadores" de informes, más que operadores por sí mismos. Por ejemplo, en el caso anterior, podemos utilizar el operador *drill*. Este operador permite entrar más al detalle en el informe. En particular, sólo es necesario que desglose la información por ciudades (en concreto, restringiéndose a sólo Valencia y León) y además seleccionando sólo la categoría "refrescos". La transformación producida tras esta operación se ilustra en la Figura 3.7, en la que se muestra cómo cambia la vista tanto en la representación relacional como el la multidimensional (la información mostrada en ambas representaciones siempre es la misma). En el resultado se puede observar que la distribución de ventas en Valencia (claramente estacional) difiere claramente de la de León (prácticamente no estacional).

CATEGORÍA	TRIMESTRE	IMPORTE
Refrescos	T1	150.323 euros
Refrescos	T2	233.992 euros
Refrescos	T3	410.497 euros
Refrescos	T4	203.400 euros
Congelados	T1	2.190.103 euros
Congelados	T2	1.640.239 euros
Congelados	T3	1.904.401 euros
Congelados	T4	2.534.031 euros

Representación Tabular (relacional):

drill-across

categoría= "refrescos"

ciudad= {"Valencia", "León"}

CATEGORÍA	TRIMESTRE	CIUDAD	IMPORTE
Refrescos	T1	Valencia	13.267
Refrescos	T1	León	3.589
Refrescos	T2	Valencia	27.392
Refrescos	T2	León	4.278
Refrescos	T3	Valencia	73.042
Refrescos	T3	León	3.780
Refrescos	T4	Valencia	18.391
Refrescos	T4	León	3.629

IMPORTE	T1	T2	T3	T4
Refrescos	150.323	233.992	410.497	203.400
Congelados	2.190.103	1.640.239	190.440	253.403

Representación Matricial (multidimensional):

Refrescos	T1	T2	T3	T4
Valencia	13.267	27.392	73.042	18.391
León	3.589	4.278	3.780	3.629

Figura 3.7. Ejemplo del operador "drill".

Lo importante de estos operadores es que modifican el informe en tiempo real y no generan uno nuevo. Lógicamente, para que esto sea eficiente el almacén de datos ha de estar diseñado e implementado para que este tipo de operaciones utilicen ciertas estructuras intermedias que permitan agregar y disgregar con facilidad.

Veamos ahora un ejemplo de la operación *roll*. Simplemente la operación *roll* es la inversa del *drill* y el objetivo es obtener información más agregada.

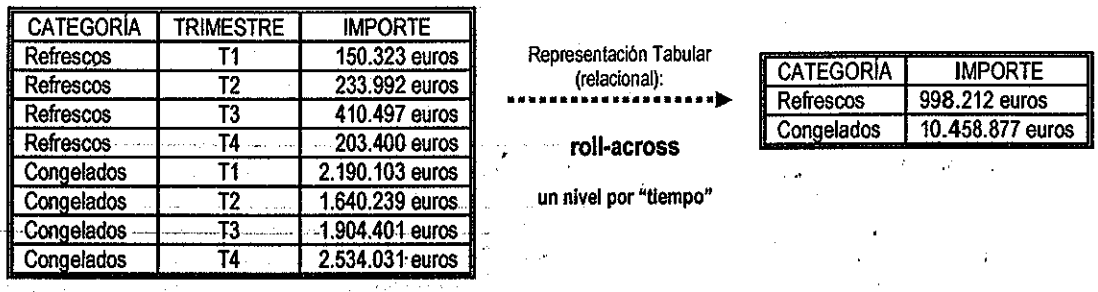


Figura 3.8. Ejemplo del operador roll.

Por ejemplo, si quisiéramos obtener los totales de las categorías "refrescos" y "congelados", simplemente sería necesario aplicar el operador *roll-across* a la consulta original, sin necesidad de crear una nueva, como se observa en la Figura 3.8.

Vistos los operadores *drill* y *roll*, cabe preguntarse por qué a veces se utiliza la notación *"-across"* (como hemos hecho nosotros) y a veces la notación *"-up"* (que incluso es más frecuente). Aunque en realidad es una cuestión meramente terminológica y no universalmente respetada, las correspondencias son las siguientes:

- *Drill-down* y *roll-up*: representan agregaciones o disgregaciones dentro de una dimensión ya definida inicialmente en la consulta.
- *Drill-across* y *roll-across*: representan agregaciones o disgregaciones en otras dimensiones de las definidas inicialmente en la consulta o hacen desaparecer alguna de las dimensiones.

Finalmente, veamos los otros dos operadores: *pivot* y *slice & dice*. Estos dos operadores se utilizan exclusivamente cuando se hace una representación matricial, o al menos una representación mixta.

Veamos en primer lugar el operador *pivot*. Supongamos que tenemos la consulta en la situación en la que estamos mostrando el importe para las categorías "refrescos" y "congelados", las ciudades "Valencia" y "León", y todos los trimestres. La posible representación (mixta, entre tabular y multidimensional) es la que se muestra en la parte izquierda de la Figura 3.9.

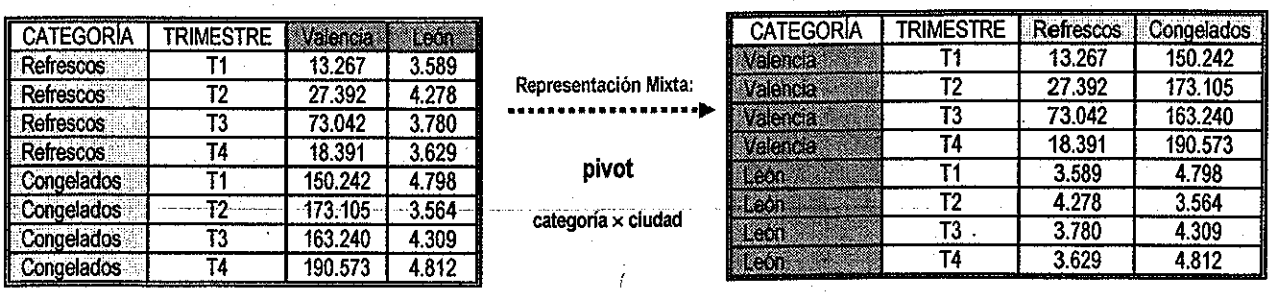


Figura 3.9. Ejemplo del operador pivot.

El operador *pivot* permite cambiar algunas filas por columnas. Esta operación, aparentemente sencilla, no está generalizada en muchos sistemas de bases de datos (en SQL-92 no existía, por ejemplo). No obstante, su inclusión es prácticamente imprescindible para poder realizar análisis de datos y, muy en particular, minería de datos. Como veremos, este cambio permite que valores de columnas pasen a ser nombre de nuevas columnas y viceversa. Como se ve en la parte derecha de la Figura 3.9 esto supone que ciertos métodos de aprendizaje proposicionales sean capaces de extraer patrones sobre la consulta de la izquierda y no sean capaces de hacerlo sobre la segunda y viceversa.

CATEGORÍA	TRIMESTRE	Valencia	León
Refrescos	T1	13.267	3.589
Refrescos	T2	27.392	4.278
Refrescos	T3	73.042	3.780
Refrescos	T4	18.391	3.629
Congelados	T1	150.242	4.798
Congelados	T2	173.105	-3.564
Congelados	T3	163.240	4.309
Congelados	T4	190.573	4.812

Representación Mixta

slice & dice

trimestre = {T1, T4}
ciudad = Valencia

CATEGORÍA	Trimestre	Valencia
Refrescos	T1	13.267
Refrescos	T4	18.391
Congelados	T1	150.242
Congelados	T4	190.573

Figura 3.10. Ejemplo del operador slice & dice.

Veamos finalmente el operador *slice & dice*. En realidad este operador permite escoger parte de la información mostrada, no por agregación sino por selección. En la Figura 3.10 se muestra un ejemplo de este operador.

Los operadores vistos son los básicos para refinar una consulta o informe, aunque distintos sistemas propietarios pueden añadir más operadores, maneras diferentes de representar los datos, de interpretar la petición de aplicación de operadores (mediante arrastre de dimensiones utilizando el ratón), etc. En los dos capítulos siguientes veremos cómo estos operadores pueden facilitar en gran medida la transformación y adecuación de datos de cara a obtener una "vista minable" que sea idónea para aplicar técnicas de minería de datos.

3.3.4 Implementación del almacén de datos. Diseño

Recordemos que una de las razones para crear un almacén de datos separado de la base de datos operacional era conseguir que el análisis se pudiera realizar de una manera eficiente. El hecho de que la estructura anterior y los operadores vistos permitan trabajar sencillamente y combinar dimensiones, detallar o agregar informes, etc., y todo ello de manera gráfica, no asegura que esto sea eficiente.

Con el objetivo de obtener la eficiencia deseada, los sistemas de almacenes de bases de datos pueden implementarse utilizando dos tipos de esquemas físicos[5]:

- **ROLAP (*Relational OLAP*)**: físicamente, el almacén de datos se construye sobre una base de datos relacional.
- **MOLAP (*Multidimensional OLAP*)**: físicamente, el almacén de datos se construye sobre estructuras basadas en matrices multidimensionales.

Las ventajas del ROLAP son, en primer lugar, que se pueden utilizar directamente sistemas de gestión de bases de datos genéricos y herramientas asociadas: SQL, restricciones,

[5] Existen sistemas mixtos, denominados HOLAP (*Hybrid OLAP*).

disparadores, etc. En segundo lugar, la formación y el coste necesario para su implementación es generalmente menor. Las ventajas del MOLAP son su especialización, la correspondencia entre el nivel lógico y el nivel físico. Esto hace que el MOLAP sea generalmente más eficiente, incluso aunque en el caso de ROLAP se utilicen ciertas técnicas de optimización, como comentaremos más abajo.

No todos los sistemas, libros y manuales son consistentes respecto a si la diferencia ROLAP/MOLAP se produce a nivel físico o a nivel lógico. En algunos textos se habla de que si el sistema representa los resultados de los informes/consultas como tablas, el sistema es ROLAP y si los representa como matrices el sistema es MOLAP. Según nuestra definición (y la de muchos otros autores) tanto ROLAP como MOLAP se refieren a la implementación y son independientes de la manera en la que, externamente, se vean las herramientas del sistema de almacenes de datos o el sistema OLAP. Por tanto un sistema puede tener una representación de las consultas relacional y estar basado en un MOLAP o puede tener una representación completamente multidimensional y estar basado en un ROLAP. Algunos ejemplos de sistemas ROLAP son *Microstrategy, Informix Metacube* u *Oracle Discoverer*. El primero, por ejemplo, tiene una interfaz completamente multidimensional mientras que por debajo existe un sistema relacional. Ejemplos de sistemas MOLAP son el *Oracle Express* o el *Hyperion Enterprise*.

Como hemos dicho, la ventaja de los ROLAP es que pueden utilizar tecnología y nomenclatura de los sistemas de bases de datos relacionales. Esto tiene el riesgo de que en algunos casos se pueda decidir mantener parte de la base de datos transaccional o inspirarse en su organización (manteniendo claves ajenas, claves primarias, conservando parte de la normalización, etc.). En general, aunque esto pueda ser cómodo inicialmente, no es conveniente a largo plazo. De hecho, una de las maneras más eficientes de implementar un datamart multidimensional mediante bases de datos relacionales se basa en ignorar casi completamente la estructura de los datos en las fuentes de origen y utiliza una estructura nueva denominada *starflake* [Kimball 1996]. Esta estructura combina los esquemas en estrella, *star* y en estrella jerárquica o copo de nieve, *snowflake*.

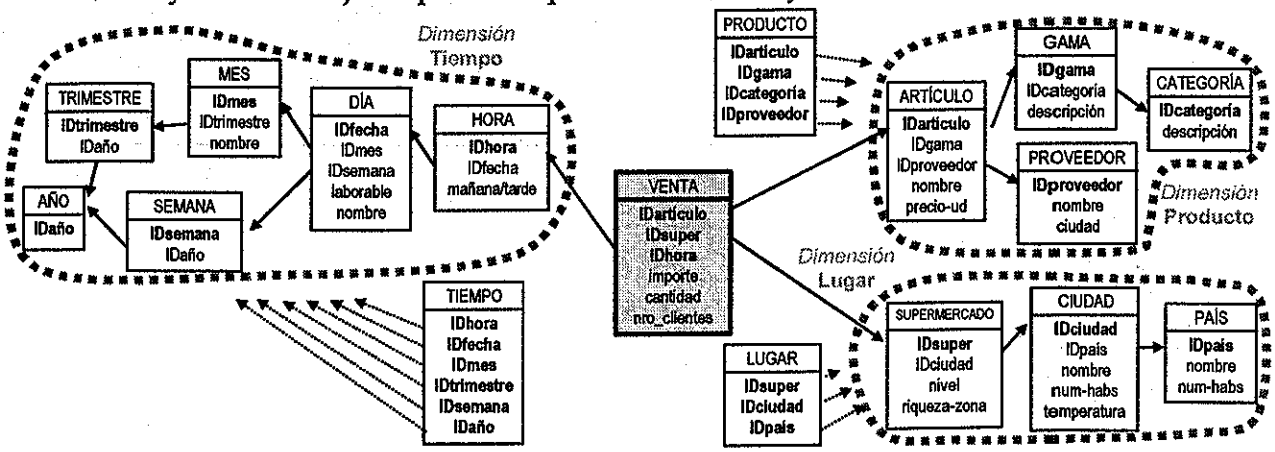


Figura 3.11. Implementación de un datamart utilizando tecnología relacional (ROLAP).

Para construir esta estructura se construyen tres tipos de tablas:

- Tablas copo de nieve (*snowflake tables*): para cada nivel de agregación de una dimensión se crea una tabla. Cada una de estas tablas tiene una clave primaria (señalada en la Figura 3.11 en negrita) y tantas claves ajenas como sean necesarias pa-

ra conectar con los niveles de agregación superiores. En la Figura 3.11, las tablas "mes", "día", "artículo", "ciudad" y "país" (entre otras) son tablas copo de nieve.

- Tabla de hechos (*fact tables*): se crea una única tabla de hechos por datamart. En esta tabla se incluye un atributo para cada dimensión, que será clave ajena (*foreign key*[6]) a cada una de las tablas copo de nieve de mayor detalle de cada dimensión. Además, todos estos atributos forman la clave primaria. Adicionalmente, pueden existir atributos que representen información de cada hecho, denominados generalmente *medidas*. En la Figura 3.11, la tabla "VENTA" es la tabla de hechos.
- Tablas estrella (*star tables*): para cada dimensión se crea una tabla que tiene un atributo para cada nivel de agregación diferente en la dimensión. Cada uno de estos atributos es una clave ajena que hace referencia a tablas copo de nieve. Todos los atributos de la tabla forman la clave primaria (señalados en negrita). En la Figura 3.11, las tablas "TIEMPO", "PRODUCTO" y "LUGAR" son tablas estrella.

Las tablas estrella son, en realidad, tablas de apoyo, ya que no representan ninguna información que no esté en las demás.

Este diseño proporciona la realización de consultas OLAP de una manera eficiente, así como la aplicación de los operadores específicos:

- Las tablas copo de nieve permiten realizar vistas o informes utilizando diferentes grados de detalle sobre varias dimensiones. Al estar normalizadas permiten seleccionar datos dimensionales de manera no redundante. Esto es especialmente útil para los operadores *drill, slice & dice* y *pivot*.
- Las tablas estrella son, como hemos dicho, tablas de apoyo, que representan "preconcatenaciones" o "pre-junciones" (*pre-joins*) entre las tablas copo de nieve. El propósito de las tablas estrella es evitar concatenaciones costosas cuando se realizan operaciones de *roll-up*.

Además de la estructura anterior, los sistemas ROLAP se pueden acompañar de estructuras especiales: índices de mapa de bits, índices de JOIN, optimizadores de consultas, extensiones de SQL (por ejemplo "CUBE"), etc., así como técnicas tan variadas como el precálculo y almacenamiento de valores agregados que vayan a utilizarse frecuentemente (totales por año, por producto, etc.). Además, se pueden desactivar los *locks* de lectura/escritura concurrente (ya que sólo hay lecturas), muchos índices dinámicos se pueden sustituir por estáticos o por *hashing* (ya que las tablas no van a crecer frecuentemente), etc. Todas estas extensiones y ajustes hacen que el sistema de gestión de bases de datos subyacente se adapte mejor a su nuevo cometido que ya no es una base de datos operacional sino un almacén de datos y proporcione la eficiencia necesaria.

Por el contrario los sistemas MOLAP almacenan físicamente los datos en estructuras multidimensionales de forma que la representación externa e interna coincidan. Las estructuras de datos utilizadas para ello son bastante específicas, lo que permite rendimientos mayores que los ROLAP. En cambio, los sistemas MOLAP tienen algunos inconvenientes:

---

[6] También llamada clave/llave externa o foránea o secundaria.

- Se necesitan sistemas específicos. Esto supone un coste de *software* mayor y generalmente compromete la portabilidad, al no existir estándares sobre MOLAP tan extendidos como los estándares del modelo relacional.
- Al existir un gran acoplamiento entre la visión externa y la implementación, los cambios en el diseño del almacén de datos obligan a una reestructuración profunda del esquema físico y viceversa.
- Existe más desnormalización que en las ROLAP. En muchos casos un almacén de datos MOLAP ocupa más espacio que su correspondiente ROLAP.

Tanto para los sistemas ROLAP y MOLAP existen numerosos aspectos que influyen en el diseño físico. Además, existen metodologías y modelos conceptuales para asistir en el diseño conceptual y lógico y, de ahí, al diseño físico. Existen extensiones del modelo entidad-relación (ER) [Sapia et al. 1999; Tryfona et al. 1999] o de modelos orientados a objetos [Trujillo et al. 2001], así como modelos específicos [Golfarelli et al. 1998]. Respecto a la representación, en especial si se utiliza una metodología orientada a objetos o se es familiar con el UML (*Unified Modeling Language*), existe un estándar *Common Warehouse Metadata* (CWM) del OMG (*Object Management Group*, http://www.omg.org). Se trata de una extensión del UML para modelar almacenes de datos.

Quizá la parte de diseño de almacenes de datos es una de las áreas más abiertas y donde existe menos convergencia. Las razones son múltiples pero, fundamentalmente, se resumen en que los almacenes de datos se han originado principalmente desde el ámbito industrial y no académico, que el fin inicial del almacén de datos era realizar OLAP eficiente, con lo que el énfasis recaía fundamentalmente en los niveles lógico y físico. A pesar de todo esto, podemos identificar cuatro pasos principales a la hora de diseñar un almacén de datos (en realidad estos pasos se han de seguir para cada datamart):

1. Elegir para modelar un "proceso" o "dominio" de la organización sobre el que se deseen realizar informes complejos frecuentemente, análisis o minería de datos. Por ejemplo, se puede hacer un datamart sobre pedidos, ventas, facturación, etc.
2. Decidir el hecho central y el "gránulo" (nivel de detalle) máximo que se va a necesitar sobre él. Por ejemplo, ¿se necesita información horaria para el tiempo?, ¿se necesita saber las cajas del supermercado o es suficiente el supermercado como unidad mínima?, etc. En general, siempre hay que considerar gránulos finos por si más adelante se fueran a necesitar, a no ser que haya restricciones de tamaño importantes. Precisamente, el almacén de datos se realiza, entre otras cosas, para poder agregar eficientemente, por lo que un almacén de datos demasiado detallado no compromete, en principio, la eficacia.
3. Identificar las dimensiones que caracterizan el "dominio" y su grafo o jerarquía de agregación, así como los atributos básicos de cada nivel. No se deben incluir atributos descriptivos más que lo imprescindible para ayudar en la visualización. En cambio, atributos informativos del estilo "es festivo", "es fin de semana", "es estival", etc., son especialmente interesantes de cara a agregaciones y selecciones que detecten patrones. Las dimensiones varían mucho de un dominio a otro, aunque respondan a preguntas como "qué", "quién", "dónde", "de dónde", "cuándo", "cómo", etc. El tiempo siempre es una (o más de una) de las dimensiones presentes.

4. Determinar y refinar las medidas y atributos necesarios para los hechos y las dimensiones. Generalmente las medidas de los hechos son valores numéricos agregables (totales, cuentas, medias...) y suelen responder a la pregunta "cuánto". Revisar si toda la información que se requiere sobre los hechos está representada en el almacén de datos.

Existen muchas otras consideraciones que hay que tener en cuenta durante el diseño (véase, por ejemplo [Inmon 2002; Kimball 1996]). Por ejemplo, no hay que obsesionarse por el espacio (algunas normalizaciones no van a mejorar la eficiencia y el ahorro en espacio no es considerable). Tampoco hay que orientarse demasiado en la estructura de la base de datos transaccional. Por ejemplo, no se debe utilizar la misma codificación de claves primarias que en la base de datos transaccional.

3.4 Carga y mantenimiento del almacén de datos

Finalmente, si se ha decidido diseñar un almacén de datos, y ya esté implementado mediante tecnología ROLAP o MOLAP, el siguiente paso es cargar los datos. El proceso tradicional de base de datos más parecido a la carga de un almacén de datos es el proceso de "migración", aunque a diferencia de él, existe un "mantenimiento" posterior.

En realidad, la carga y mantenimiento de un almacén de datos es uno de los aspectos más delicados y que más esfuerzo requiere (alrededor de la mitad del esfuerzo necesario para implantar un almacén de datos), y, de hecho, suele existir un sistema especializado para realizar estas tareas, denominado **sistema ETL** (*Extraction, Transformation, Load*)[7]. Dicho sistema no se compra en el supermercado ni se descarga de Internet, sino que:

- La construcción del ETL es responsabilidad del equipo de desarrollo del almacén de datos y se realiza específicamente para cada almacén de datos.

Afortunadamente, aunque un ETL se puede construir realizando programas específicos, también se puede realizar adaptando herramientas genéricas (por ejemplo *triggers*), herramientas de migración o utilizando herramientas más específicas que van apareciendo cada vez más frecuentemente.

El sistema ETL se encarga de realizar muchas tareas:

- Lectura de datos transaccionales: se trata generalmente de obtener los datos mediante consultas SQL sobre la base de datos transaccional. Generalmente se intenta que esta lectura sea en horarios de poca carga transaccional (fines de semana o noches). Para la primera carga los datos pueden encontrarse en históricos y es posible que en distintos formatos. Este hecho condiciona muchas veces el número de años que se puede incluir en el almacén de datos.
- Incorporación de datos externos: generalmente aquí se deben incorporar otro tipo de herramientas, como *wrappers*, para convertir texto, hojas de cálculo o HTML en XML o en tablas de base de datos que se puedan integrar en el almacén de datos.

[7] Existen traducciones diversas en castellano, como ETC (Extracción, Transformación, Carga) o ETT (Extracción, Transformación, Transporte).

- Creación de claves: en general se recomienda crear claves primarias nuevas para todas las tablas que se vayan creando en el almacenamiento intermedio o en el almacén de datos.
- Integración de datos: consiste en muchos casos en la fusión de datos de distintas fuentes, detectar cuándo representan los mismos objetos y generar las referencias y restricciones adecuadas para conectar la información y proporcionar integridad referencial.
- Obtención de agregaciones: si se sabe que cierto nivel de detalle no es necesario en ningún caso, una primera fase de agregación se puede realizar aquí.
- Limpieza y transformación de datos: aunque de estas dos tareas nos dedicaremos en el capítulo siguiente, parte de la limpieza y la transformación necesaria para organizar el almacén se realiza por el ETL. Se trata, como veremos, de evitar datos redundantes, inconsistentes, estandarizar medidas, formatos, fechas, tratar valores nulos, etc.
- Creación y mantenimiento de metadatos: para que todo el ETL pueda funcionar es necesario crear y mantener metadatos sobre el propio proceso ETL y los pasos realizados y por realizar.
- Identificación de cambios: esto se puede realizar de muy distintas maneras: mediante una carga total cada vez que haya un cambio, mediante comparación de instancias (uso de archivos delta), mediante marcas de tiempo (*time stamping*) en los registros, mediante disparadores, mediante el archivo de *log* o mediante técnicas mixtas. Algunas son muy ineficiente (carga total o uso de disparadores) y otras son muy complejas de implementar (archivo de *log*). Generalmente, por tanto, se utilizan técnicas mixtas.
- Planificación de la carga y mantenimiento: consiste en definir las fases de carga, el orden, para evitar violar restricciones de integridad, del mismo modo que se realizan las migraciones, y las ventanas de carga, con el objetivo de poder hacer la carga sin saturar ni la base de datos transaccional, así como el mantenimiento sin paralizar el almacén de datos.
- Indización: finalmente se han de crear índices sobre las claves y atributos del almacén de datos que se consideren relevantes (niveles de dimensiones, tablas de hechos, etc.).
- Pruebas de calidad: en realidad se trata de definir métricas de calidad de datos del almacén de datos, así como implantar un programa de calidad de datos, con un responsable de calidad que realice un seguimiento, especialmente si el almacén de datos se desea utilizar para el apoyo en decisiones estratégicas o especialmente sensibles.

Generalmente, para realizar todas estas tareas, los sistemas ETL se basan en un repositorio de datos intermedio, como se muestra en la Figura 3.12. Esto puede parecer que ya es abusar de recursos, al tener además de la base de datos transaccional y el almacén de datos un tercer repositorio de datos de similar magnitud. Sin embargo, este almacenamiento intermedio es extremadamente útil, ya que hay tareas que no se pueden realizar en el sistema transaccional ni en el almacén de datos. Por ejemplo, la limpieza y transformación de datos se pueden realizar tranquilamente en este repositorio intermedio, ciertos

metadatos pueden almacenarse ahí y valores agregados intermedios también pueden residir ahí, así como los valores integrados de fuentes externas. Con ello, muchos procesos del ETL, incluidos el mantenimiento, se pueden realizar en gran medida sin paralizar ni la base de datos transaccional ni el almacén de datos.

Esta estructura basada en un "almacenamiento intermedio" se muestra en la Figura 3.12 y sitúa más claramente las siglas del acrónimo ETL.

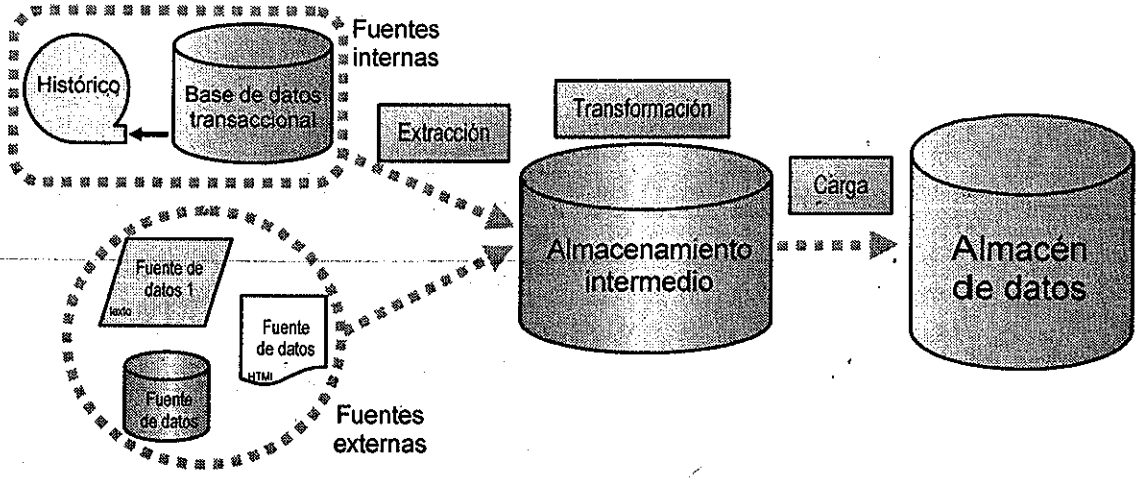


Figura 3.12. El sistema ETL basado en un repositorio intermedio.

La organización con almacenamiento intermedio es especialmente indicada para integrar la información externa. Como dijimos en la introducción, dicha información externa es especialmente importante para encontrar patrones o aspectos significativos en muchos casos, por lo que no nos podemos limitar a la información de la base de datos transaccional. Por ejemplo, la Figura 3.13 muestra que se pueden malinterpretar los datos si no se compara (o se integra) con información externa; una tendencia que parece atisbar una recuperación puede verse como una pérdida de mercado si se compara con la competencia. Del mismo modo, sin esta información externa, puede ser prácticamente imposible extraer patrones.

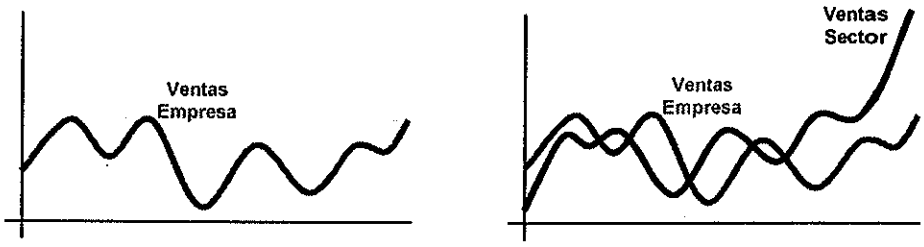


Figura 3.13. La importancia de usar fuentes externas.

En general, existe información que suele ser apropiada para muchos almacenes de datos: demografías (censo), datos resumidos de áreas geográficas, distribución de la competencia, evolución de la economía, información de calendarios y climatológicas, programaciones televisivas-deportivas, catástrofes, páginas amarillas, psicografías sociales, información de otras organizaciones, datos compartidos en una industria o área de negocio, organizaciones y colegios profesionales, catálogos, etc.

El valor de estas fuentes externas ha propiciado la existencia de un mercado de este tipo de base de datos. En aquellos casos en los que no se puede obtener de una manera

gratuita (por ejemplo, desde algún organismo público), algunas bases de datos se pueden comprar a compañías especializadas. Gran parte de estas bases de datos externas no tienen información personal y por tanto no infringen leyes de protección de datos. En el caso de que esto pudiera ser así, es muy importante estar al tanto de la legalidad al respecto (de este tema trataremos en el Capítulo 23).

3.5 Almacenes de datos y minería de datos

El concepto de almacenes de datos nace hace más de una década [Inmon 1992] ligado al concepto de EIS (*Executive Information System*), el sistema de información ejecutivo de una organización. En realidad, cuando están cubiertas las necesidades operacionales de las organizaciones se plantean herramientas informáticas para asistir o cubrir en las necesidades estratégicas.

La definición original de almacén de datos es una "colección de datos, orientada a un dominio, integrada, no volátil y variante en el tiempo para ayudar en las decisiones de dirección" [Inmon 1992; Inmon 2002]. A raíz de esta definición, parecería que los almacenes de datos son sólo útiles en empresas o instituciones donde altos cargos directivos tengan que tomar decisiones. A partir de ahí, y de la difusión cada vez mayor de las herramientas de *business intelligence* y OLAP, podríamos pensar que los almacenes de datos no se aplican en otros ámbitos: científicos, médicos, ingenieriles, académicos, donde no se tratan con las variables y problemáticas típicas de las organizaciones y empresas.

Al contrario, en realidad, los almacenes de datos pueden utilizarse de muy diferentes maneras, y pueden agilizar muchos procesos diferentes de análisis. En la Figura 3.14 se pueden observar las distintas aplicaciones y usos que se puede dar a un almacén de datos: herramientas de consultas e informes, herramientas EIS, herramientas OLAP y herramientas de minería de datos.

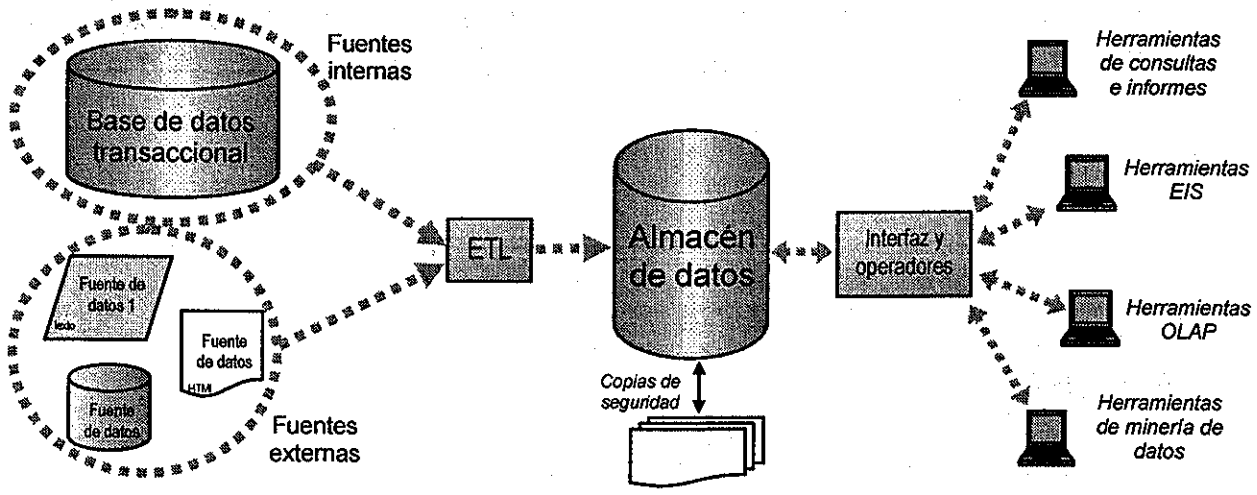


Figura 3.14. Perspectiva general y usos de un almacén de datos.

La variedad de usos que se muestran en la figura anterior sugiere también la existencia de diferentes grupos de usuarios: analistas, ejecutivos, investigadores, etc. Según el carácter de estos usuarios se les puede catalogar en dos grandes grupos:

- "picapedreros" (o "granjeros"): se dedican fundamentalmente a realizar informes periódicos, ver la evolución de indicadores, controlar valores anómalos, etc.

- "exploradores": encargados de encontrar nuevos patrones significativos utilizando técnicas OLAP o de minería de datos. La estructura del almacén de datos y sus operadores facilita la obtención de diferentes vistas de análisis o "vistas minables".

Esta diferencia, y el hecho de que se catalogue como "exploradores" a aquellos que utilizan técnicas OLAP o minería de datos, no nos debe hacer confundir las grandes diferencias de un análisis clásico, básicamente basado en la agregación, la visualización y las técnicas descriptivas estadísticas con un uso genuino de minería de datos que no transforma los datos en otros datos (más o menos agregados) sino que transforma los datos en conocimiento (o más humildemente, en reglas o modelos).

Un aspecto a destacar es que el nivel de agregación para los requerimientos de análisis OLAP puede ser mucho más grueso que el necesario para la minería de datos. Por ejemplo, para el análisis OLAP puede ser suficiente usar como unidad mínima de lugar el supermercado. En cambio, para la minería de datos puede ser interesante tener un nivel más fino (por caja o por cajera).

Los almacenes de datos no son imprescindibles para hacer extracción de conocimiento a partir de datos. En realidad, se puede hacer minería de datos sobre un simple archivo de datos. Sin embargo, las ventajas de organizar un almacén de datos se amortizan sobradamente a medio y largo plazo. Esto es especialmente patente cuando nos enfrentamos a grandes volúmenes de datos, o éstos aumentan con el tiempo, o provienen de fuentes heterogéneas o se van a querer combinar de maneras arbitrarias y no predefinidas. Tampoco es cierto que un almacén de datos sólo tenga sentido si tenemos una base de datos transaccional inicial. Incluso si todos los datos originalmente no provienen de bases de datos puede ser conveniente la realización de un almacén de datos.

En gran medida, un almacén de datos también facilita la limpieza y la transformación de datos (en especial para generar "vistas minables" en tiempo real). La limpieza y transformación de datos se tratan precisamente en el capítulo siguiente.