CAPÍTULO 2

DATA WAREHOUSES

Un *Data Warehouse* (DW) es un gran repositorio lógico de datos que permite el acceso y la manipulación flexible de grandes volúmenes de información provenientes tanto de transacciones detalladas como datos agregados de fuentes de distinta naturaleza [19]. Los sistemas de administración de DW integran información procedente de diversos sistemas operacionales, la seleccionan, la historizan y la almacenan para proporcionar la base para la planeación, control y toma de decisiones a un alto nivel. En este capítulo se expondrá la arquitectura *Data Warehouse*, su funcionamiento y componentes en general. En la primera sección se expone la arquitectura básica de un *Data Warehouse* y la forma en que los componentes funcionan dentro del sistema. La sección 2 da una descripción de los modelos multidimensionales a través de los cuales se representa la información que contiene un *warehouse*. La sección 3 se refiere a la construcción de los *warehouses* y detalla cada uno de los componentes principales de la arquitectura que intervienen en su construcción, tanto en la carga de los datos como en su mantenimiento. Por último la sección 4 presenta una descripción general de las técnicas de análisis OLAP (*On Line Analytical Processing*) utilizadas generalmente para la explotación de los *Data Warehouses*.

2.1 Arquitectura general

La arquitectura general de un DW es la que se muestra en la Figura 1. Este diagrama muestra como primer componente dentro de la arquitectura DW a las fuentes desde las

cuales se extrae la información necesaria para poblar la base de datos. Conectada a cada una de las fuentes se encuentran los siguientes componentes básicos de la arquitectura, los *wrappers* o extractores, los cuales extraen y transforman la información de las fuentes. Posteriormente través de un integrador dicha información se carga a la base de datos, la cual constituye el siguiente componente básico de la arquitectura. Este proceso de cargado de la información ejecuta las tareas siguientes:

- Transforma los datos de acuerdo al modelo de datos del *warehouse*.
- Limpia dichos datos para corregir y depurar errores que pueden contener las fuentes (por lo general se generan en la captura de los datos en los sistemas de transacción diaria).
- Integra todos los datos para formar la base de datos en la cual se encontrará la información.

De igual manera, los meta datos deben ser refrescados dentro de este proceso. Dicho proceso es crítico para asegurar la calidad de la información y soportar una adecuada toma de decisiones con datos correctos y previamente verificados. Una vez que los datos han sido cargados se encuentran disponibles para un sistema que soporte decisiones. Sin embargo, las aplicaciones no accesan directamente el *warehouse* debido a que es demasiado grande, además de poseer un esquema genérico no óptimo para el usuario final. Por consiguiente, vistas especializadas más pequeñas del DW son cargadas en los *data marts*, éstos son repositorios más pequeños con vistas materializadas para facilitar la consulta de los datos (ver Figura 2.1). Esta carga se realiza a través de un segundo proceso más simple debido a que los datos ya se encuentran ordenados y verificados dentro del DW. Únicamente se seleccionan las vistas requeridas y a través de una serie

de transformaciones necesarias quedan establecidas para facilitar y acelerar el proceso de consulta del usuario. Finalmente los *data marts* son accesados a través de las herramientas para el usuario final (OLAP o ambientes de consultas analíticas, generalmente), las cuales permiten analizar la información disponible en el *warehouse* para la generación de consultas especializadas, reportes, nuevas clasificaciones y tendencias que sirvan de apoyo a la toma de decisiones.

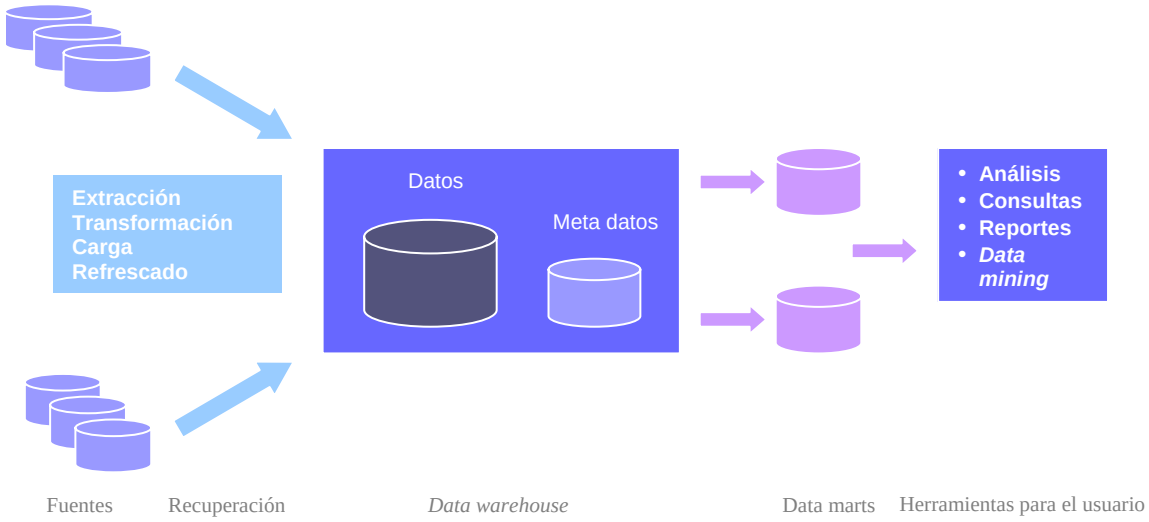


Figura 2.1: Arquitectura básica de un DW.

2.2 Modelo multidimensional

Para facilitar el análisis de los datos, un D*ata Warehouse* representa los datos que contiene usando modelos multidimensionales. De manera general, un modelo multidimensional provee dos conceptos principales: *medida* y *dimensión*. Una *medida* es un valor en un espacio multidimensional definido por *dimensiones* ortogonales. Así, el cubo es el concepto central del modelado de datos multidimensional como se muestra en

la Figura 2.2, donde se muestra una instancia del modelo multidimensional: un esquema del mismo tipo.

Dentro del modelo de datos multidimensional las medidas o atributos numéricos describen un cierto proceso del mundo real el cual va a ser objeto de un análisis. Estos atributos dependen de ciertas dimensiones las cuales proveen el contexto a través del cual van a ser interpretadas las medidas. Dichas dimensiones regularmente se encuentran en orden jerárquico (ejemplo: tiempo: día → mes → año). Las medidas pueden ser agregadas a lo largo de las dimensiones lo cual resulta en un cubo el cual es la base para el uso de las operaciones OLAP, estas operaciones serán explicadas más adelante en otra sección.

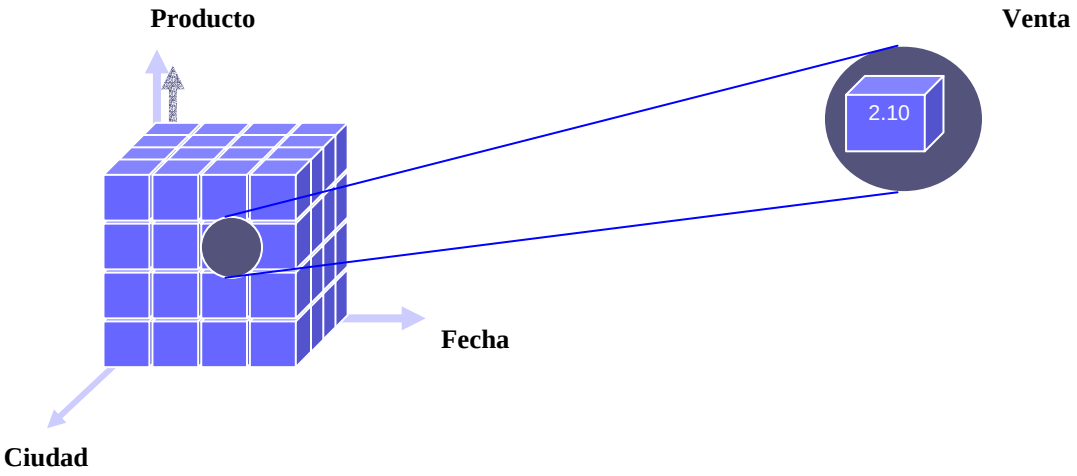


Figura 2.2: Esquema multidimensional de bases de datos

El esquema multidimensional presentado en la Figura 2.2 puede ser implementado directamente a través de un servidor MOLAP (Multidimensional OLAP). Dichos servidores soportan vistas multidimensionales de los datos a través de un repositorio multidimensional. Esto hace posible implementar consultas multidimensionales a la base a través de un mapeo directo. Otra alternativa para implementar el modelo

multidimensional es a través de la tecnología ROLAP (*Relational* OLAP). Esta tecnología utiliza un esquema de bases de datos relacional para representar la información (datos y medidas) del esquema multidimensional. Ambas tecnologías son útiles y tienen sus méritos. El esquema relacional puede manejar grandes cantidades de datos y los nuevos avances en esta tecnología han mejorado para el manejo de *Data Warehouses*. Los sistemas MOLAP debido a la representación de los datos pueden responder rápidamente a consultas muy complejas y permitir así un análisis rápido de la información. Sin embargo siguen teniendo problemas para bases con grandes cantidades de datos.

La tecnología mayormente usada para representar los esquemas multidimensionales en el manejo de DW es la ROLAP [10]. Cuando un servidor relacional es utilizado, el modelo multidimensional y sus operaciones deben ser mapeadas a relaciones y las consultas basadas en SQL (*Structured Query Language*). La mayoría de los DW utilizan el esquema en estrella para representar el modelo multidimensional de bases de datos. En la Figura 2.3 se muestra un esquema de este tipo. La base consiste de una tabla simple de hechos que contiene un apuntador a cada una de las dimensiones que proveen las coordenadas del esquema multidimensional y guarda las medidas numéricas para esas coordenadas. Cada tabla de dimensión consiste de columnas que corresponden a los atributos de cada dimensión.

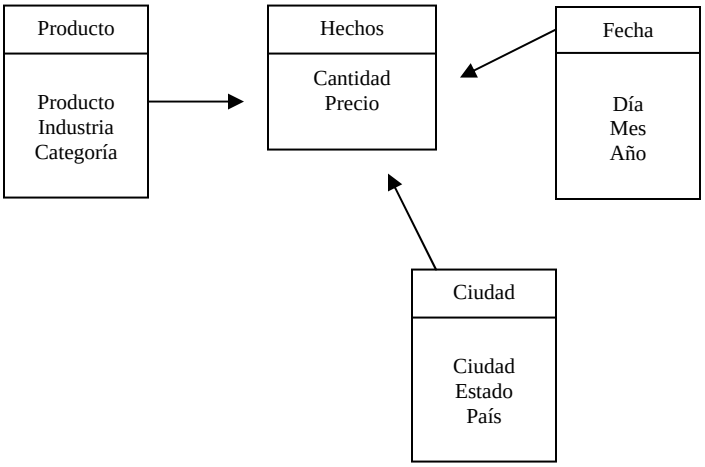


Figura 2.3: Esquema en estrella

Los esquemas en estrella generalmente no proveen un soporte explícito para la jerarquía de cada una de las dimensiones. Por lo cual generalmente después de generar un esquema en estrella se realiza una normalización del mismo generando un esquema copo de nieve [6]. Los esquemas copo de nieve como se muestra en la Figura 2.4 se basan en el esquema en estrella para realizar una normalización del mismo y obtener un esquema que representa de mejor manera el modelo multidimensional del DW.

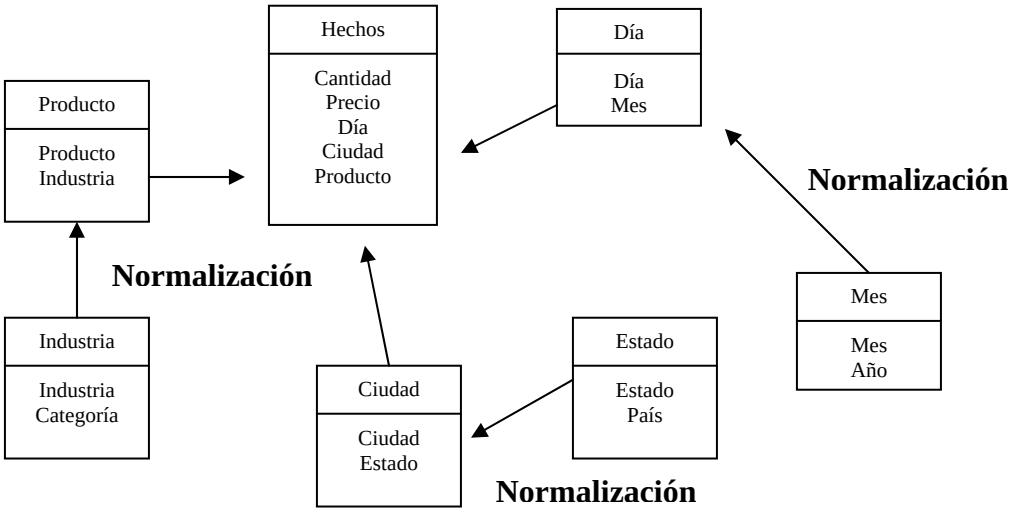


Figura 2.4: Esquema copo de nieve

2.3 Construcción

La Figura 2.5 presenta la arquitectura de construcción del DW. En la etapa de construcción, es importante resaltar algunos de los elementos representados en su arquitectura los cuales interactúan directamente para construir la base que integrará el *warehouse*. Las fuentes de datos se encuentran en lo más bajo de la construcción ya que es de donde se extrae la información a través de la utilización de los *wrappers*, los cuales son programas que extraen cierto tipo de información seleccionada y la transforman al modelo de datos que se utilice en el DW. El siguiente componente es el integrador, el cual toma esta información previamente homogeneizada y la integra según el esquema del *warehouse* para construir una base consistente, completa y sin errores. A continuación se describirán más a detalle cada uno de los componentes mencionados que participan en la construcción del DW.

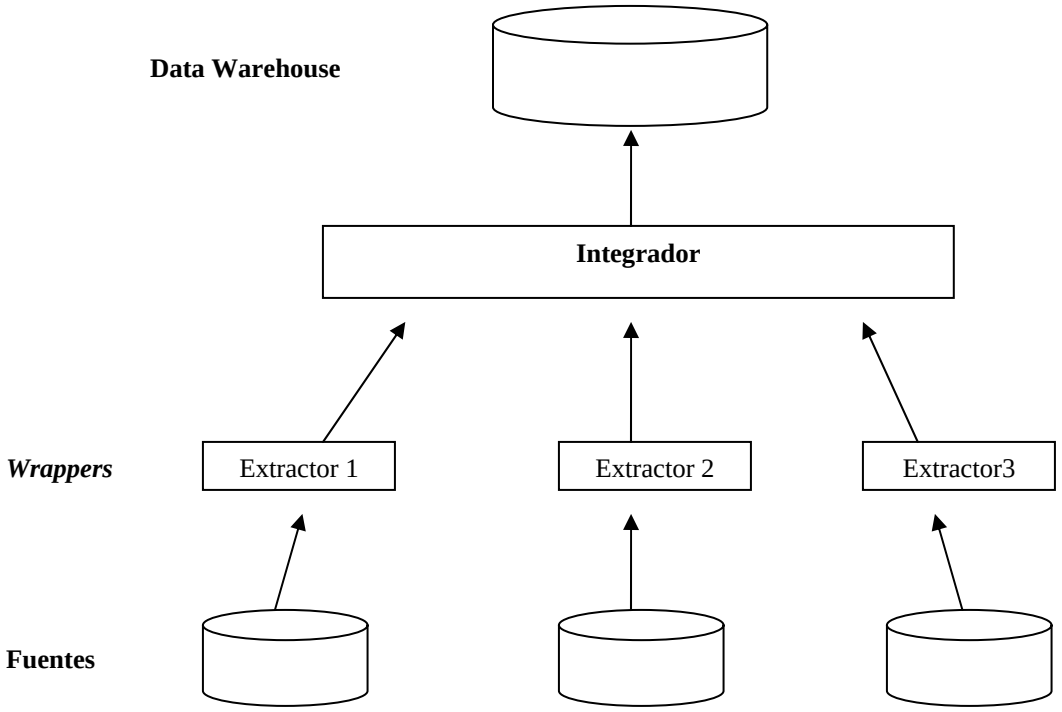


Figura 2.5: Ilustración de carga y refrescado del DW

2.3.1 Fuentes

El primer componente de la arquitectura de un DW consiste en las fuentes de las cuales se extrae la información. En la mayoría de los casos estas fuentes son sistemas OLTP. Estos sistemas fueron diseñados para trabajar "*stand-alone*" con los archivos que utilizan en sus aplicaciones. En grandes instalaciones hay miles de estos archivos que coexisten para un sin número de aplicaciones con una gran redundancia en sus datos y con distintos formatos dependiendo de la aplicación bajo la cual trabajen. Estos sistemas son los de procesamiento diario, realizan todas las transacciones necesarias dentro de una empresa o institución, manejando una gran cantidad de datos, por consiguiente dichos datos debidamente seleccionados e integrados proporcionan la base para soportar un adecuado análisis de la información a un nivel más alto de dirección. Otras fuentes de las cuales se puede obtener información son otros sistemas organizacionales como sistemas de oficina así como bases de datos no convencionales, pero que también pueden contener información relevante (ejemplo: archivos generados a mano, documentos HTML y SGML, sistemas legales, bases de conocimiento y cualquier información electrónica de la cual se puedan obtener datos importantes para el análisis que se busca).

Dichas fuentes no son modificadas dentro de sus sistemas nativos, únicamente son utilizadas para extraer la información necesaria, a la cual se le realizan las transformaciones pertinentes para ser utilizada. Por consiguiente, el único software que es introducido a este nivel son los *wrappers* que representan la extracción y transformación de la información necesaria para poblar la base del DW. El objetivo principal de la implementación de esta arquitectura es que los sistemas transaccionales y las fuentes sigan funcionando de manera habitual y sólo se utilicen para extraer información que

necesite el DW sin generar ningún cambio en el lugar donde se encuentran y a la vez proporcionen información que ayude al análisis de los datos.

Una característica importante de la arquitectura de un *warehouse* es la separación que se establece entre los sistemas operacionales y el procesamiento de la información para la toma de decisiones. Debido a esta separación la arquitectura del *warehouse* permite la coexistencia de ambos sistemas permitiendo así la optimización del procesamiento de datos ya que mientras los sistemas transaccionales continúan sus funciones de procesamiento diario, es posible a través del DW realizar el análisis de la información necesario [1].

2.3.2 Extractores

Los extractores son los programas que extraen la información necesaria de las fuentes para realizar la carga del *warehouse*. Conectado a cada fuente de información se encuentra un extractor el cual tiene dos responsabilidades interrelacionadas:

- **Traducción:** Realiza un proceso de traducción de la información seleccionada que extrae de las fuentes para que sea representada usando el mismo esquema de datos del sistema de *warehouse*. Por ejemplo, si la fuente de información consiste de un conjunto de archivos con un esquema local y el esquema del *warehouse* es un esquema relacional entonces el *wrapper* debe soportar una interfaz que presente los datos como si la fuente original de donde provienen fuera relacional. El problema de la traducción es inherente a casi todas las aplicaciones de integración de datos y no es sólo específico del *data warehousing*. Típicamente

un componente que traduce la fuente de información a un modelo integral es llamado *traductor* o *wrapper*.

- **Detección de cambios:** Monitoreo de las fuentes de información para la detección de cambios importantes que se puedan reflejar dentro del DW y propagarlos al integrador, el cual especificaremos más adelante. Una ventaja generalmente utilizada es simplemente propagar copias enteras de información relevante desde las fuentes de datos al integrador del DW periódicamente. Este caso de detección de cambios es pertinente realizarlo en sistemas en los que el DW debe estar siempre conectado a las fuentes para continuos accesos a las fuentes. En caso contrario cuando el sistema puede estar desconectado a las fuentes, no es necesario realizar una función de detección de cambios en el *wrapper*.

Finalmente, es importante hacer notar que es necesario implementar un *wrapper* distinto para cada fuente de información ya que su funcionalidad depende directamente del tipo de la fuente así como de los tipos de datos que provee dicha fuente.

2.3.3 Integrador

El integrador es una parte importante de la arquitectura del *warehouse* y se encuentra en estrecha relación con el trabajo que realizan los *wrappers*. Una vez que la información es extraída de las fuentes y se encuentra homogeneizada, se integra para realizar la carga de la base del DW. Es decir, el integrador se encarga de integrar la información de acuerdo a la estructura definida en el *warehouse* así como de cargarlo con dicha información. Asumiendo que el *warehouse* ha sido cargado con los datos inicialmente extraídos de las

fuentes la tarea constante del integrador es recibir los cambios notificados por el *wrapper* de las fuentes de información y reflejar estos cambios en el DW. En un nivel muy abstracto los datos dentro del DW pueden ser vistos como una vista materializada (un conjunto de estas vistas) donde la base de los datos reside en las fuentes de información.

Viendo el problema de esta manera el trabajo del integrador es esencialmente representar las vistas materializadas constantemente actualizadas.

2.3.4 Mantenimiento

El mantenimiento del DW se refiere básicamente a mantener actualizado el DW con información correcta, precisa y actualizada en cualquier momento que se utilice. Los *warehouses* como ya sabemos contienen información histórica seleccionada y almacenada la cual es necesaria para la correcta toma de decisiones. Debido a que esta información es seleccionada de distintas fuentes, muchas de las cuales, son sistemas transaccionales diarios y su información se está actualizando y cambiando a cada momento, resulta necesario contar con un mecanismo de actualización o refrescado de la base. Dicho mecanismo debe asegurar que la información que se analice sea correcta según los cambios que se presenten al día. Este mantenimiento o refrescado de la información depende de los datos que posee el *warehouse* o de las necesidades de análisis para las cuales fue creado. Generalmente las funciones de refrescado se llevan a cabo cada determinado tiempo o durante las noches mientras los sistemas transaccionales no se encuentran funcionando o trabajan menos que durante el día cuando todas las actividades que se procesan están siendo realizadas. El encargado de mantener actualizado el DW es

el integrador a través de las notificaciones de cambios realizadas por el *wrapper*, el cual se encuentra en interacción directa con las fuentes de datos.

En muchos sistemas de DW la base de datos del *warehouse* se encuentra separada de las fuentes de datos lo cual refleja que la información que está siendo analizada por el sistema no necesita de una constante actualización de los datos y ésta se puede conectar en ciertos periodos de tiempo que se establezcan como necesarios para realizar las actividades de refrescado. En otros sistemas donde la información fluye constantemente y se encuentra en constante cambio es necesario que el DW se encuentre todo el tiempo conectado y que los *wrappers* le notifiquen al integrador de los cambios efectuados para que se vean reflejados en el *warehouse* inmediatamente.

Cuando el integrador recibe notificaciones de cambios en orden de integrar los cambios en el *warehouse* necesita adicionalmente datos relacionados de las demás fuentes para asegurar que la consulta que va a realizar para refrescar la información afecta todas las relaciones involucradas. A esta alternativa de refrescado se denomina incremental e implica actualizar únicamente los datos que registraron algún cambio así como refrescar todas las relaciones que involucran dichos datos. En casos extremos en los que no se actualicen sólo las relaciones afectadas, se utiliza la segunda alternativa de refrescado, en la cual se realiza una copia de toda la información relevante de las fuentes. Las vistas del *warehouse* son recalculadas en su totalidad si es necesario, asegurando así que la información que fue modificada ha sido actualizada de manera correcta en todas las partes en la que se encuentra relacionada, ya sea dentro de las vistas materializadas así como dentro de la base central del *warehouse*. Esta alternativa es un poco cara ya que

implica obtener de nuevo toda la información pero muchas veces es la forma más usual de asegurar la legalidad de la información y asegurar que es correcta [4].

El mantenimiento del *warehouse* es una parte importante dentro de la arquitectura. Es importante que se analice detalladamente qué necesidades de refrescado se tienen según el análisis requerido para que la información siempre se encuentre óptima para la toma de decisiones.

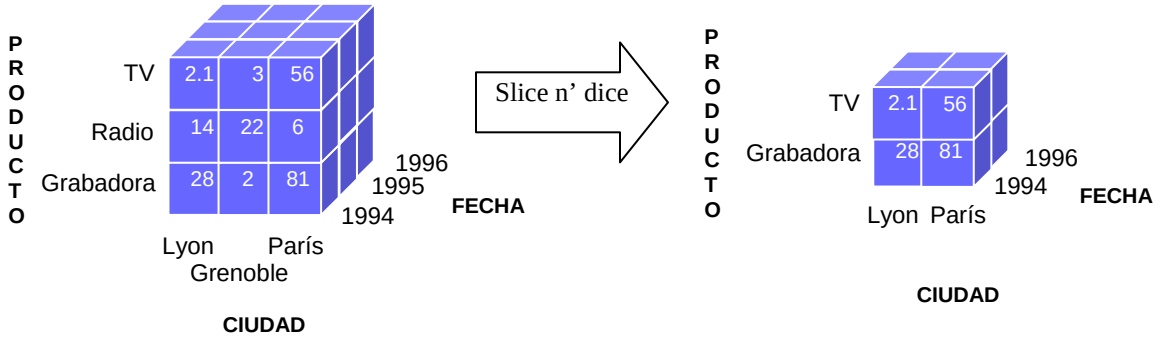
2.4 Análisis OLAP

El análisis de un DW se refiere a la manipulación o explotación de la información almacenada dentro del mismo, a la forma en que el usuario consultará el sistema y a la parte de información a la que éste tendrá acceso. El análisis es el conjunto de información que se selecciona dentro del DW mediante una consulta específica. Dicho análisis depende directamente de las necesidades de los usuarios y del tipo de decisiones que quieran que se soporten a través del uso del DW construido.

Las técnicas OLAP son ampliamente utilizadas para este tipo de tareas, a través del uso de sus operadores se lleva a cabo la explotación de la información almacenada. Entre los operadores con los que se cuenta para realizar estos procesos de análisis podemos citar los siguientes:

- **Slice'n dice**: Éste nos permite hacer una selección de los valores de las dimensiones que uno requiere. En la Figura 2.6 se puede observar un ejemplo de este tipo en el cual el usuario solo necesita ciertas regiones y productos en un tiempo específico.

- **Roll-up**: Este operador nos permite agregar los datos en sus distintos niveles de agrupación definidos previamente en el esquema multidimensional. Un ejemplo de dicho operador se encuentra en la Figura 2.7 donde de toda la información que se tiene únicamente se quieren conocer los productos que se vendieron en una región (compuesta por varias ciudades) determinada. En otras palabras es subir las consultas de un nivel de agregación específico a otro más amplio.
- **Drill-down**: Este operador OLAP es el que nos permite bajar a los niveles más atómicos de nuestro esquema multidimensional, en sentido inverso al roll- up. En la Figura 2.7 se ilustra su funcionamiento.



Slice'n dice Producto != Radio, Tienda != ¨Grenoble¨ Tiempo != 1995 (SALES)

Figura 2.6: Slice n'dice

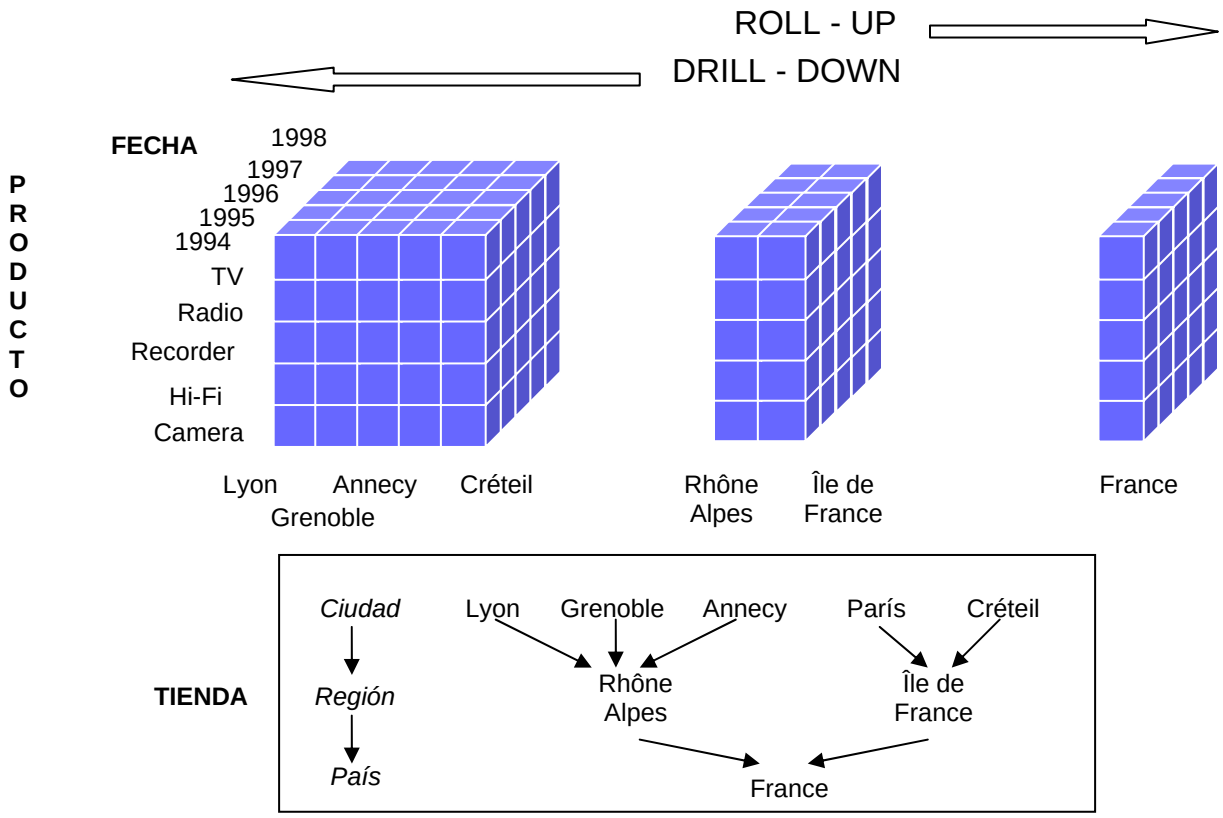


Figura 7: Roll up y drill down

2.5 Discusión

La tecnología DW provee a las empresas, grandes ventajas en el manejo de la información como son: el manejo de información con rapidez, información concisa y confiable. Por ello muchas empresas están tratando de implementar sistemas con estas características. Algunos de los beneficios se mencionan a continuación:

- **Múltiples fuentes.** Es posible acceder a información de diversas bases de datos teniendo un DW, ya que permite consolidar los datos y poder tener acceso a la información de todas las bases en una sola.

- **Información histórica.** Una de las posibilidades que nos ofrece el DW es el manejo de la información histórica, mediante el uso de versiones, las cuales se van almacenando cada vez que se ejecuten actualizaciones sobre los datos.

- **Resumen de la información.** Mediante la agregación se reduce significativamente el tamaño de los archivos; esto permite el manejo de mayores volúmenes de información en archivos más pequeños.

- **Menor tiempo de respuesta.** Dado que la información esta ya agregada, el tiempo de respuesta se reduce considerablemente en lo referente a la generación de reportes y consultas.

- **Capacidad de análisis.** Es posible mostrar la información mediante gráficas o reportes globalizados, dado que los datos se encuentran ya resumidos. Esto genera que se evalúen las situaciones por las que la empresa está atravesando.

Implementar un DW a través del modelo multidimensional tiene un número importante de ventajas [6]:

- Primero, un modelo multidimensional es una estructura homogénea y predecible. Reportes escritos, herramientas de consulta e interfaces de usuario, todas pueden hacer suposiciones acerca del modelo multidimensional para que las interfaces del usuario sean más entendibles y los procesos, más eficientes.

- Una segunda característica del modelo multidimensional reside en que es una estructura predecible. Cada dimensión es equivalente. Todas las dimensiones

pueden ser vistas como un conjunto de puntos igualmente simétricos dentro de una tabla de hechos.

- Una tercera característica del modelo multidimensional, es que es extensible para acomodar nuevos elementos de datos inesperados y nuevas decisiones de diseño. Esto se lleva a acabo añadiendo nuevos hechos de forma inesperada, añadiendo nuevas dimensiones, añadiendo nuevos atributos dimensionales y cambiando datos de una cierta granularidad para pasarlos a otra.

- Una última característica del modelo multidimensional es la creciente cantidad de herramientas administrativas y de software que manejan y usan la granularidad.

Una vez mostrada la técnica DW en el siguiente capítulo se muestra la estrategia para aprovecharla en una tecnología de apoyo a la toma de decisiones.