HEFESTO

DATA WAREHOUSING: Investigación y Sistematización de Conceptos

HEFESTO: Metodología para la Construcción de un Data Warehouse

Ing. Bernabeu Ricardo Dario

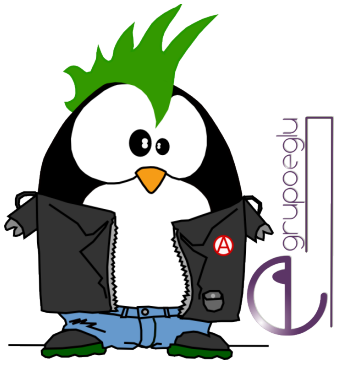
Córdoba, Argentina – Lunes 19 de Julio de 2010

Fecha	Versión	Autor/a	Detalle del cambio
Lunes 19 de Julio de 2010	2.0	Ing. Bernabeu Ricardo Dario	Actualización.
Lunes 31 de Agosto de 2009	1.2	Ing. Fernandez Carlos	Sección: Area de Datos.
Martes 21 de Abril de 2009	1.1	Ing. Bernabeu Ricardo Dario	Actualización.
Sábado 17 de Enero de 2009	1.0	Ing. Bernabeu Ricardo Dario	Actualización.
Miércoles 07 de Noviembre de 2007	0.1	Ing. Bernabeu Ricardo Dario	Versión Inicial.

...si supiese qué es lo que estoy haciendo,
no lo llamaría INVESTIGACIÓN...

Albert Einstein

Contacto:



- Blogs:
  - Personal: HEFESTO [http://tgx-hefesto.blogspot.com].
  - Dataprix: Dataprix [http://www.dataprix.com/blogs/bernabeu-dario].
- Mail: darioSistemas@gmail.com (poner en asunto [HEFESTO]).
- Red Social:
  - LinkedIn [http://www.linkedin.com/in/bernabeudario].
  - XING [https://www.xing.com/profile/Dario_Bernabeu].
  - Open Business Intelligence [http://www.redopenbi.com/profile/BernabeuRDario].
- Soluciones OSBI:
  - eGlu BI [http://www.eglubi.com.ar].
  - Mail: dbernabeu@grupoeglu.com.ar

A partir de la versión 2.0 de esta publicación, se han dejado de lado todos los términos que tienden a ”masculinizar” el lenguaje y en su lugar se ha optado por otra forma de expresión que es inclusiva para todos los géneros.

Por ejemplo, en vez de escribir ”los usuarios”, se utiliza ”l@s usuari@s”.

Índice general

I DATA WAREHOUSING: Investigación y Sistematización de Conceptos 1

RESUMEN 3

1. BUSINESS INTELLIGENCE 5
1.1. Introducción . . . 5
1.2. Definición . . . 6
1.3. Proceso de BI . . . 6
1.4. Beneficios . . . 7

2. DATA WAREHOUSING & DATA WAREHOUSE 9
2.1. Introducción . . . 9
2.2. Definición . . . 9
2.3. Características . . . 10
2.3.1. Orientada al negocio . . . 10
2.3.2. Integrada . . . 11
2.3.3. Variante en el tiempo . . . 12
2.3.4. No volátil . . . 13
2.4. Cualidades . . . 13
2.5. Ventajas . . . 14
2.6. Desventajas . . . 15
2.7. Redundancia . . . 16
2.8. Estructura . . . 16
2.9. Flujo de Datos . . . 18

3. ARQUITECTURA DEL DATA WAREHOUSING 19
3.1. Introducción . . . 19
3.2. OLTP . . . 20
3.3. Load Manager . . . 21
3.3.1. Extracción . . . 22
3.3.2. Transformación . . . 22
3.3.2.1. Codificación . . . 22
3.3.2.2. Medida de atributos . . . 23
3.3.2.3. Convenciones de nombramiento . . . 24
3.3.2.4. Fuentes múltiples . . . 24
3.3.2.5. Limpieza de datos . . . 24
3.3.3. Carga . . . 25
3.3.4. Proceso ETL . . . 26
3.4. Data Warehouse Manager . . . 27
3.4.1. Base de datos multidimensional . . . 28
3.4.2. Tablas de Dimensiones . . . 28
3.4.2.1. Tabla de Dimensión Tiempo . . . 29

3.4.3. Tablas de Hechos . . . 30
3.4.3.1. Tablas de hechos agregadas y preagregadas . . . 32
3.4.4. Cubo Multidimensional: introducción . . . 33
3.4.4.1. Indicadores . . . 34
3.4.4.2. Atributos . . . 35
3.4.4.3. Jerarquías . . . 35
3.4.4.4. a) Relación . . . 36
3.4.4.5. b) Granularidad . . . 37
3.4.5. Tipos de modelamiento de un DW . . . 37
3.4.5.1. Esquema en Estrella . . . 37
3.4.5.2. Esquema Copo de Nieve . . . 39
3.4.5.3. Esquema Constelación . . . 40
3.4.6. OLTP vs DW . . . 41
3.4.7. Tipos de implementación de un DW . . . 42
3.4.7.1. ROLAP . . . 42
3.4.7.2. MOLAP . . . 43
3.4.7.3. HOLAP . . . 44
3.4.7.4. ROLAP vs MOLAP . . . 44
3.4.8. Cubo Multidimensional: profundización . . . 45
3.4.9. Metadatos . . . 49
3.4.9.1. Mapping . . . 50
3.5. Query Manager . . . 51
3.5.1. Drill-down . . . 53
3.5.2. Drill-up . . . 55
3.5.3. Drill-across . . . 57
3.5.4. Roll-across . . . 58
3.5.5. Pivot . . . 59
3.5.6. Page . . . 60
3.5.7. Drill-through . . . 63
3.6. Herramientas de Consulta y Análisis . . . 64
3.6.1. Reportes y Consultas . . . 66
3.6.2. OLAP . . . 66
3.6.3. Dashboards . . . 67
3.6.4. Data Mining . . . 68
3.6.4.1. Redes Neuronales . . . 69
3.6.4.2. Sistemas Expertos . . . 69
3.6.4.3. Programación Genética . . . 69
3.6.4.4. Árboles de Decisión . . . 70
3.6.4.5. Detección de Desviación . . . 70
3.6.5. EIS . . . 70
3.7. Usuari@s . . . 71

4. CONCEPTOS COMPLEMENTARIOS 73
4.1. Sistema de Misión Crítica . . . 73
4.2. Data Mart . . . 73
4.3. SGBD . . . 75
4.4. Particionamiento . . . 76
4.5. Business Models . . . 76
4.6. Áreas de Datos . . . 77
4.6.1. Staging Area . . . 77
4.6.2. Operational Data Store . . . 78
4.6.3. Almacén de Datos Corporativo . . . 78
4.6.4. Data Mart . . . 79

II HEFESTO: Metodología para la Construcción de un Data Warehouse 81

RESUMEN 83

5. METODOLOGÍA HEFESTO 85
5.1. Introducción . . . 85
5.2. Descripción . . . 86
5.3. Características . . . 88
5.4. Empresa analizada . . . 88
5.5. Pasos y aplicación metodológica . . . 89
5.5.1. PASO 1) ANÁLISIS DE REQUERIMIENTOS . . . 89
5.5.1.1. a) Identificar preguntas . . . 89
5.5.1.2. b) Identificar indicadores y perspectivas . . . 90
5.5.1.3. c) Modelo Conceptual . . . 91
5.5.2. PASO 2) ANÁLISIS DE LOS OLTP . . . 93
5.5.2.1. a) Conformar indicadores . . . 93
5.5.2.2. b) Establecer correspondencias . . . 93
5.5.2.3. c) Nivel de granularidad . . . 95
5.5.2.4. d) Modelo Conceptual ampliado . . . 98
5.5.3. PASO 3) MODELO LÓGICO DEL DW . . . 99
5.5.3.1. a) Tipo de Modelo Lógico del DW . . . 99
5.5.3.2. b) Tablas de dimensiones . . . 99
5.5.3.3. c) Tablas de hechos . . . 101
5.5.3.4. d) Uniones . . . 104
5.5.4. PASO 4) INTEGRACIÓN DE DATOS . . . 105
5.5.4.1. a) Carga Inicial . . . 105
5.5.4.2. b) Actualización . . . 110
5.6. Creación de Cubos Multidimensionales . . . 112
5.6.1. Creación de Indicadores . . . 112
5.6.2. Creación de Atributos . . . 113
5.6.3. Creación de Jerarquías . . . 114
5.6.4. Otros ejemplos de cubos multidimensionales . . . 115

6. CONSIDERACIONES DE DISEÑO 117
6.1. Tamaño del DW . . . 117
6.2. Tiempo de construcción . . . 118
6.3. Implementación . . . 118
6.4. Performance . . . 118
6.5. Mantenimiento . . . 119
6.6. Impactos . . . 119
6.7. DM como sub proyectos . . . 119
6.8. Teoría de grafos . . . 119
6.9. Elección de columnas . . . 120
6.10. Claves primarias en tablas de Dimensiones . . . 122
6.11. Balance de diseño . . . 122
6.12. Relación muchos a muchos . . . 123
6.13. Claves Subrogadas . . . 124
6.14. Dimensiones lentamente cambiantes . . . 125
6.14.1. SCD Tipo 1: Sobreescribir . . . 126
6.14.2. SCD Tipo 2: Añadir fila . . . 127
6.14.3. SCD Tipo 3: Añadir columna . . . 128
6.14.4. SCD Tipo 4: Tabla de Historia separada . . . 129
6.14.5. SCD Tipo 6: Híbrido . . . 129

6.15 Dimensiones Degeneradas . . . 129
6.16 Dimensiones Clustering . . . 130

Apéndice A **133**

A. Descripción de la empresa **133**
A.1. Identificación de la empresa . . . 133
A.2. Objetivos . . . 133
A.3. Políticas . . . 133
A.4. Estrategias . . . 134
A.5. Organigrama . . . 134
A.6. Datos del entorno específico . . . 134
A.7. Relación de las metas de la organización con las del DWH . . . 135
A.8. Procesos . . . 135

Apéndice B **137**

B. Licencia de Documentación Libre de GNU **137**
B.1. Preámbulo . . . 137
B.2. Aplicabilidad y definiciones . . . 138
B.3. Copia literal . . . 139
B.4. Copiado en cantidad . . . 140
B.5. Modificaciones . . . 140
B.6. Combinación de documentos . . . 142
B.7. Colecciones de documentos . . . 142
B.8. Agregación con trabajos independientes . . . 142
B.9. Traducción . . . 143
B.10. Terminación . . . 143
B.11. Revisiones futuras de esta licencia . . . 143
B.12. Adenda: cómo usar esta Licencia en sus documentos . . . 144

Bibliografía **145**

Parte I

DATA WAREHOUSING: Investigación y Sistematización de Conceptos

RESUMEN

En esta primera parte de la publicación, se sistematizarán todos los conceptos inherentes al Data Warehousing, haciendo referencia a cada uno de ellos en forma ordenada, en un marco conceptual claro, en el que se desplegarán sus características y cualidades, y teniendo siempre en cuenta su relación o interrelación con los demás componentes del ambiente.

Inicialmente, se definirá el concepto de Business Intelligence y sus respectivas características. Seguidamente, se introducirá al Data Warehousing y se expondrán sus aspectos más relevantes y significativos. Luego, se precisarán y detallarán todos los componentes que intervienen en su arquitectura, de manera organizada e intuitiva, atendiendo su interrelación. Finalmente, se describirán algunos conceptos complementarios que deben tenerse en cuenta.

El principal objetivo de esta investigación, es ayudar a comprender el complejo ambiente del Data Warehousing, sus respectivos componentes y la interrelación entre los mismos, así como también cuales son sus ventajas, desventajas y características propias. Es por ello, que se hará énfasis en la sistematización de todos los conceptos de la estructura del Data Warehousing, debido a que la documentación existente se enfoca en tratar temas independientes sin tener en cuenta su vinculación y referencias a otros componentes del mismo.

Capítulo 1

BUSINESS INTELLIGENCE

1.1. Introducción

Actualmente, en las actividades diarias de cualquier organización, se generan datos como producto secundario, que son el resultado de todas las transacciones que se realizan. Es muy común, que los mismos se almacenen y administren a través de sistemas transaccionales en bases de datos relacionales.

Pero, la idea central de esta publicación, es que estos dejen de solo ser simples datos, para convertirse en información que enriquezca las decisiones de l@s usuari@s.

Precisamente, la inteligencia de negocios (Business Intelligence - BI), permite que el proceso de toma de decisiones esté fundamentado sobre un amplio conocimiento de sí mismo y del entorno, minimizando de esta manera el riesgo y la incertidumbre.

Además, propicia que las organizaciones puedan traducir sus objetivos en indicadores de estudio, y que estos puedan ser analizados desde diferentes perspectivas, con el fin de encontrar información que no solo se encargue de responder a preguntas de lo que está sucediendo o ya sucedió, sino también, que posibilite la construcción de modelos, mediante los cuales se podrán predecir eventos futuros.

Cuando se nombra el término inteligencia, se refiere a la aplicación combinada de información, habilidad, experiencia y razonamientos, para resolver un problema de negocio.

Cabe destacar, que la aplicación de soluciones BI no es solo para grandes-medianas empresas, sino para quien desee tomar decisiones a través del análisis de sus datos. Es por ello que las soluciones BI no solo se enfocarán a resolver temas relacionados a: aumentar la rentabilidad, disminuir costos y obtener la famosa ventaja competitiva.

De acuerdo a lo planteado anteriormente se presentarán dos grandes ejemplos de la aplicación de BI, una en una empresa de ventas de productos, la otra en una biblioteca vecinal:

1. Empresa de venta de productos: en este caso la aplicación de BI podrá resolver las siguientes preguntas.
    - ¿Quiénes son l@s mejores client@s?.
    - ¿Cómo minimizar costos y maximizar las prestaciones?.

- ¿Cuál será el pronóstico de ventas del próximo mes?.

2. Biblioteca vecinal: en este caso la aplicación de BI podrá resolver las siguientes preguntas.
   - ¿Cuál es la temática más consultada?.
   - ¿Qué días hay mayor concurrencia, y por qué?.
   - ¿Qué libros deben ser adquiridos?.

1.2. Definición

Se puede describir BI, como un concepto que integra por un lado el almacenamiento y por el otro el procesamiento de grandes cantidades de datos, con el principal objetivo de transformarlos en conocimiento y en decisiones en tiempo real, a través de un sencillo análisis y exploración.

La definición antes expuesta puede representarse a través de la siguiente fórmula:

Datos + Análisis = **Conocimiento**

Este conocimiento debe ser oportuno, relevante, útil y debe estar adaptado al contexto de la organización.

Existe una frase muy popular acerca de BI, que dice: “Inteligencia de Negocios es el proceso de convertir datos en conocimiento y el conocimiento en acción, para la toma de decisiones”.

BI hace hincapié en los procesos de recolectar y utilizar efectivamente la información, con el fin de mejorar la forma de operar de una organización, brindando a sus usuari@s, el acceso a la información clave que necesitan para llevar a cabo sus tareas habituales y más precisamente, para poder tomar decisiones oportunas basadas en datos correctos y certeros.

Al contar con la información exacta y en tiempo real, es posible, aparte de lo ya mencionado, identificar y corregir situaciones antes de que se conviertan en problemas y en potenciales pérdidas de control de la empresa, pudiendo conseguir nuevas oportunidades o readaptarse frente a la ocurrencia de sucesos inesperados.

La Inteligencia de Negocios tiene sus raíces en los Sistemas de Información Ejecutiva[1] (Executive Information Systems – EIS) y en los Sistemas para la Toma de Decisiones[2] (Decision Support Systems – DSS), pero ha evolucionado y se ha transformado en todo un conjunto de tecnologías capaces de satisfacer a una gran gama de usuari@s junto a sus necesidades específicas en cuanto al análisis de información.

1.3. Proceso de BI

A fin de comprender cómo una organización puede crear inteligencia de sus datos, para, como ya se ha mencionado, proveer a l@s usuari@s finales oportuna y acertadamente acceso a esta información, se describirá a continuación el proceso de BI. El mismo

[1] Ver sección 3.6.5, en la página 70.

[2] Los DSS son una clase especial de sistemas de información cuyo objetivo es analizar datos de diferentes procedencias y brindar soporte para la toma de decisiones.

esta dividido en cinco fases, las cuales serán explicadas teniendo como referencia el siguiente gráfico, que sintetiza todo el proceso:

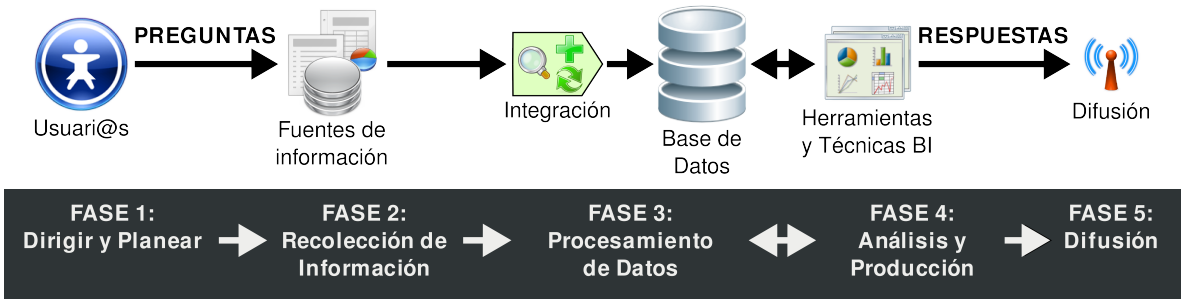


Figura 1.1: Fases del proceso BI.

- FASE 1: Dirigir y Planear. En esta fase inicial es donde se deberán recolectar los requerimientos de información específicos de l@s diferentes usuari@s, así como entender sus diversas necesidades, para que luego en conjunto con ell@s se generen las preguntas que les ayudarán a alcanzar sus objetivos.
- FASE 2: Recolección de Información. Es aquí en donde se realiza el proceso de extraer desde las diferentes fuentes de información de la empresa, tanto internas como externas, los datos que serán necesarios para encontrar las respuestas a las preguntas planteadas en el paso anterior.
- FASE 3: Procesamiento de Datos. En esta fase es donde se integran y cargan los datos en crudo en un formato utilizable para el análisis. Esta actividad puede realizarse mediante la creación de una nueva base de datos, agregando datos a una base de datos ya existente o bien consolidando la información.
- FASE 4: Análisis y Producción. Ahora, se procederá a trabajar sobre los datos extraídos e integrados, utilizando herramientas y técnicas propias de la tecnología BI, para crear inteligencia. Como resultado final de esta fase se obtendrán las respuestas a las preguntas, mediante la creación de reportes, indicadores de rendimiento, cuadros de mando, gráficos estadísticos, etc.
- FASE 5: Difusión. Finalmente, se les entregará a l@s usuari@s que lo requieran las herramientas necesarias, que les permitirán explorar los datos de manera sencilla e intuitiva.

1.4. Beneficios

Entre los beneficios más importantes que BI proporciona a las organizaciones, vale la pena destacar los siguientes:

- Reduce el tiempo mínimo que se requiere para recoger toda la información relevante de un tema en particular, ya que la misma se encontrará integrada en una fuente única de fácil acceso.
- Automatiza la asimilación de la información, debido a que la extracción y carga de los datos necesarios se realizará a través de procesos predefinidos.

- Proporciona herramientas de análisis para establecer comparaciones y tomar decisiones.
- Cierra el círculo que hace pasar de la decisión a la acción.
- Permite a l@s usuari@s no depender de reportes o informes programados, porque los mismos serán generados de manera dinámica.
- Posibilita la formulación y respuesta de preguntas que son claves para el desempeño de la organización.
- Permite acceder y analizar directamente los indicadores de éxito.
- Se pueden identificar cuáles son los factores que inciden en el buen o mal funcionamiento de la organización.
- Se podrán detectar situaciones fuera de lo normal.
- Permitirá predecir el comportamiento futuro con un alto porcentaje de certeza, basado en el entendimiento del pasado.
- L@s usuari@s podrán consultar y analizar los datos de manera sencilla e intuitiva.

Capítulo 2

DATA WAREHOUSING & DATA WAREHOUSE

2.1. Introducción

Debido a que para llevar a cabo BI, es necesario gestionar datos guardados en diversos formatos, fuentes y tipos, para luego depurarlos e integrarlos, además de almacenarlos en un solo destino o base de datos que permita su posterior análisis y exploración, es imperativo y de vital importancia contar con un proceso que satisfaga todas estas necesidades. Este proceso se denomina Data Warehousing.

El Data Warehousing (DWH), es el encargado de extraer, transformar, consolidar, integrar y centralizar los datos que una organización genera en todos los ámbitos de su actividad diaria (compras, ventas, producción, etc) y/o información externa relacionada. Permitiendo de esta manera el acceso y exploración de la información requerida, a través de una amplia gama de posibilidades de análisis multivariables, con el objetivo final de dar soporte al proceso de toma de desiciones estratégico y táctico.

2.2. Definición

El Data Warehousing posibilita la extracción de datos de sistemas operacionales y fuentes externas, permite la integración y homogeneización de los datos de toda la empresa, provee información que ha sido transformada y sumarizada, para que ayude en el proceso de toma de decisiones estratégicas y tácticas.

El Data Warehousing, convertirá entonces los datos operacionales de la empresa en una herramienta competitiva, debido a que pondrá a disposición de l@s usuari@s indicad@s la información pertinente, correcta e integrada, en el momento que se necesita.

Pero para que el Data Warehousing pueda cumplir con sus objetivos, es necesario que la información que se extrae, transforma y consolida, sea almacenada de manera centralizada en una base de datos con estructura multidimensional denominada Data Warehouse (DW).

Una de las definiciones más famosas sobre DW, es la de William Harvey Inmon, quien define: “Un Data Warehouse es una colección de datos orientada al negocio, integrada, variante en el tiempo y no volátil para el soporte del proceso de toma de decisiones de

la gerencia".

Debido a que W. H. Inmon, es reconocido mundialmente como el padre del DW, la explicación de las características más sobresalientes de este concepto se basó en su definición.

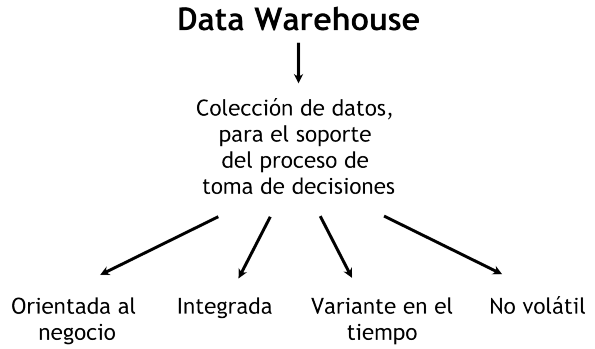


Figura 2.1: Data Warehouse, características.

Cabe aclarar que los términos almacén de datos y depósito de datos, son análogos a DW, y se utilizarán de aquí en adelante para referirse al mismo.

2.3. Características

2.3.1. Orientada al negocio

La primera característica del DW, es que la información se clasifica en base a los aspectos que son de interés para la organización. Esta clasificación afecta el diseño y la implementación de los datos encontrados en el almacén de datos, debido a que la estructura del mismo difiere considerablemente a la de los clásicos procesos operacionales orientados a las aplicaciones.

A continuación, y con el fin de obtener una mejor comprensión de las diferencias existentes entre estos dos tipos de orientación, se realizará un análisis comparativo:

- Con respecto al nivel de detalle de los datos, el DW excluye la información que no será utilizada exclusivamente en el proceso de toma de decisiones; mientras que en los procesos orientados a las aplicaciones, se incluyen todos aquellos datos que son necesarios para satisfacer de manera inmediata los requerimientos funcionales de la actividad que soporten. Por ejemplo, los datos comunes referidos a l@s client@s, como su dirección de correo electrónico, fax, teléfono, D.N.I., código postal, etc, que son tan importantes de almacenar en cualquier sistema operacional, no son tenidos en cuenta en el depósito de datos por carecer de valor para la toma de decisiones, pero sí lo serán aquellos que indiquen el tipo de cliente, su clasificación, ubicación geográfica, edad, etc.

- En lo que concierne a la interacción de la información, los datos operacionales mantienen una relación continua entre dos o más tablas, basadas en alguna regla comercial vigente; en cambio las relaciones encontradas en los datos residentes del

DW son muchas, debido a que por lo general cada tabla del mismo estará conformada por la integración de varias tablas u otras fuentes del ambiente operacional, cada una con sus propias reglas de negocio inherentes.

El origen de este contraste es totalmente lógico, ya que el ambiente operacional se diseña alrededor de las aplicaciones u programas que necesite la organización para llevar a cabo sus actividades diarias y funciones específicas. Por ejemplo, una aplicación de una empresa minorista manejará: stock, lista de precios, cuentas corrientes, pagos diferidos, impuestos, retenciones, ventas, notas de crédito, compras, etc. De esta manera, la base de datos combinará estos elementos en una estructura que se adapte a sus necesidades.

En contraposición, siguiendo con el ejemplo anterior, en una empresa minorista el ambiente DW se organizará alrededor de entidades de alto nivel tales como: clientes, productos, rubros, proveedores, vendedores, zonas, etc. Que son precisamente aquellos sujetos mediante los cuales se desea analizar la información. Esto se debe a que el depósito de datos se diseña para realizar consultas e investigaciones sobre las actividades de la organización y no para soportar los procesos que se realizan en ella.

En síntesis, la ventaja de contar con procesos orientados a la aplicación, esta fundamentada en la alta accesibilidad de los datos, lo que implica un elevado desempeño y velocidad en la ejecución de consultas, ya que las mismas están predeterminadas; mientras que en el DW para satisfacer esta ventaja se requiere que la información este desnormalizada, es decir, con redundancia[1] y que la misma esté dimensionada, para evitar tener que recorrer toda la base de datos cuando se necesite realizar algún análisis determinado, sino que simplemente la consulta sea enfocada por variables de análisis que permitan localizar los datos de manera rápida y eficaz, para poder de esta manera satisfacer una alta demanda de complejos exámenes en un mínimo tiempo de respuesta.

2.3.2. Integrada

La integración implica que todos los datos de diversas fuentes que son producidos por distintos departamentos, secciones y aplicaciones, tanto internos como externos, deben ser consolidados en una instancia antes de ser agregados al DW, y deben por lo tanto ser analizados para asegurar su calidad y limpieza, entre otras cosas. A este proceso se lo conoce como Integración de Datos, y cuenta con diversas técnicas y subprocesos para llevar a cabo sus tareas. Una de estas técnicas son los procesos ETL: Extracción, Transformación y Carga de Datos[2] (Extraction, Transformation and Load).

Si bien el proceso ETL es solo una de las muchas técnicas de la Integración de Datos, el resto de estas técnicas puede agruparse muy bien en sus diferentes etapas. Es decir, en el proceso de Extracción tendremos un grupo de técnicas enfocadas por ejemplo en tomar solo los datos indicados y mantenerlos en un almacenamiento intermedio; en el proceso de Transformación por ejemplo estarán aquellas técnicas que analizarán los datos para verificar que sean correctos y válidos; en el proceso de Carga de Datos se agruparán por ejemplo técnicas propias de la carga y actualización del DW.

La integración de datos, resuelve diferentes tipos de problemas relacionados con las convenciones de nombres, unidades de medidas, codificaciones, fuentes múltiples, etc., cada uno de los cuales será correctamente detallado y ejemplificado más adelante.

La causa de dichos problemas, se debe principalmente a que a través de los años l@s diseñador@s y programador@s no se han basado en ningún estándar concreto para

[1] Ver sección 2.7, en la página 16.
[2] Ver sección 3.3, en la página 21.

definir nombres de variables, tipos de datos, etc., ya sea por carecer de ellos o por no creer que sean necesarios. Por lo cual, cada uno por su parte ha dejado en cada aplicación, módulo, tabla, etc., su propio estilo personalizado, confluyendo de esta manera en la creación de modelos muy inconsistentes e incompatibles entre sí.

Los puntos de integración afectan casi todos los aspectos de diseño, y cualquiera sea su forma, el resultado es el mismo, ya que la información será almacenada en el DW en un modelo globalmente aceptable y singular, aún cuando los sistemas operacionales y demás fuentes almacenen los datos de maneras disímiles, para que de esta manera l@s usuari@s finales estén enfocad@s en la utilización de los datos del depósito y no deban cuestionarse sobre la confiabilidad o solidez de los mismos.

2.3.3. Variante en el tiempo

Debido al gran volumen de información que se manejará en el DW, cuando se le realiza una consulta, los resultados deseados demorarán en originarse. Este espacio de tiempo que se produce desde la búsqueda de datos hasta su consecución es del todo normal en este ambiente y es, precisamente por ello, que la información que se encuentra dentro del depósito de datos se denomina de tiempo variable.

Esta característica básica, es muy diferente de la información encontrada en el ambiente operacional, en el cual, los datos se requieren en el momento de acceder, es decir, que se espera que los valores procurados se obtengan a partir del momento mismo de acceso.

Además, toda la información en el DW posee su propio sello de tiempo:

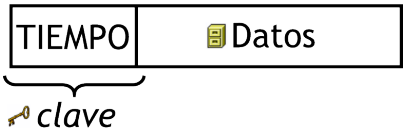


Figura 2.2: Data Warehouse, variante en el tiempo.

Esto contribuye a una de las principales ventajas del almacén de datos: los datos son almacenados junto a sus respectivos históricos. Esta cualidad que no se encuentra en fuentes de datos operacionales, garantiza poder desarrollar análisis de la dinámica de la información, pues ella es procesada como una serie de instantáneas, cada una representando un periodo de tiempo. Es decir, que gracias al sello de tiempo se podrá tener acceso a diferentes versiones de la misma información.

Es importante tener en cuenta la granularidad[3] de los datos, así como también la intensidad de cambio natural del comportamiento de los fenómenos de la actividad que se desarrolle, para evitar crecimientos incontrolables y desbordamientos de la base de datos.

El intervalo de tiempo y periodicidad de los datos debe definirse de acuerdo a la necesidad y requisitos de l@s usuari@s.

[3] Ver sección 3.4.4.5, en la página 37.

Es elemental aclarar, que el almacenamiento de datos históricos, es lo que permite al DW desarrollar pronósticos y análisis de tendencias y patrones, a partir de una base estadística de información.

2.3.4. No volátil

La información es útil para el análisis y la toma de decisiones solo cuando es estable. Los datos operacionales varían momento a momento, en cambio, los datos una vez que entran en el DW no cambian.

La actualización, o sea, insertar, eliminar y modificar, se hace de forma muy habitual en el ambiente operacional sobre una base, registro por registro, en cambio en el depósito de datos la manipulación básica de los datos es mucho más simple, debido a que solo existen dos tipos de operaciones: la carga de datos y el acceso a los mismos.

Por esta razón es que en el DW no se requieren mecanismos de control de concurrencia y recuperación.

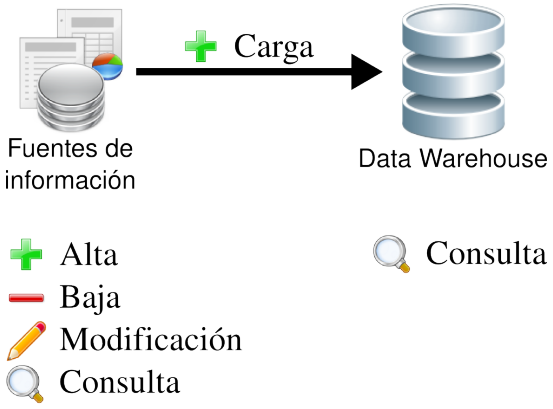


Figura 2.3: Data Warehouse, no volátil.

2.4. Cualidades

Una de las primeras cualidades que se puede mencionar del DW, es que maneja un gran volumen de datos, debido a que consolida en su estructura la información recolectada durante años, proveniente de diversas fuentes y áreas, en un solo lugar centralizado. Es por esta razón que el depósito puede ser soportado y mantenido sobre diversos medios de almacenamiento.

Además, como ya se ha mencionado, el almacén de datos presenta la información sumarizada y agregada desde múltiples versiones, y maneja información histórica.

Organiza y almacena los datos que se necesitan para realizar consultas y procesos analíticos, con el propósito de responder a preguntas complejas y brindarles a l@s usuari@s finales la posibilidad de que mediante una interface amigable, intuitiva y fácil de utilizar, puedan tomar decisiones sobre los datos sin tener que poseer demasiados conocimientos informáticos. El DW permite un acceso más directo, es decir, la información

gira en torno al negocio, y es por ello que también l@s usuari@s pueden sentirse cómod@s al explorar los datos y encontrar relaciones complejas entre los mismos.

Cabe aclarar que el Data Warehousing no se compone solo de datos, ni tampoco solo se trata de un depósito de datos aislado. El Data Warehousing hace referencia a un conjunto de herramientas para consultar, analizar y presentar información, que permiten obtener o realizar análisis, reporting, extracción y explotación de los datos, con alta performance, para transformar dichos datos en información valiosa para la organización.

Con respecto a las tecnologías que son empleadas, se pueden encontrar las siguientes:

- Arquitectura cliente/servidor.
- Técnicas avanzadas para replicar, refrescar y actualizar datos.
- Software front-end, para acceso y análisis de datos.
- Herramientas para extraer, transformar y cargar datos en el depósito, desde múltiples fuentes muy heterogéneas.
- Sistema de Gestión de Base de Datos[4] (SGBD).

  Todas las cualidades expuestas anteriormente, son imposibles de saldar en un típico ambiente operacional, y esto es una de las razones de ser del Data Warehousing.

2.5. Ventajas

A continuación se enumerarán algunas de las ventajas más sobresalientes que trae aparejada la implementación de un Data Warehousing y que ejemplifican de mejor modo sus características y cualidades:

- Transforma datos orientados a las aplicaciones en información orientada a la toma de decisiones.
- Integra y consolida diferentes fuentes de datos (internas y/o externas) y departamentos empresariales, que anteriormente formaban islas, en una única plataforma sólida y centralizada.
- Provee la capacidad de analizar y explotar las diferentes áreas de trabajo y de realizar un análisis inmediato de las mismas.
- Permite reaccionar rápidamente a los cambios del mercado.
- Aumenta la competitividad en el mercado.
- Elimina la producción y el procesamiento de datos que no son utilizados ni necesarios, producto de aplicaciones mal diseñadas o ya no utilizadas.
- Mejora la entrega de información, es decir, información completa, correcta, consistente, oportuna y accesible. Información que l@s usuari@s necesitan, en el momento adecuado y en el formato apropiado.

[4] Ver sección 4.3, en la página 75.

- Logra un impacto positivo sobre los procesos de toma de decisiones. Cuando l@s usuari@s tienen acceso a una mejor calidad de información, la empresa puede lograr por sí misma: aprovechar el enorme valor potencial de sus recursos de información y transformarlo en valor verdadero; eliminar los retardos de los procesos que resultan de información incorrecta, inconsistente y/o inexistente; integrar y optimizar procesos a través del uso compartido e integrado de las fuentes de información; permitir a l@s usuari@s adquirir mayor confianza acerca de sus propias decisiones y de las del resto, y lograr así, un mayor entendimiento de los impactos ocasionados.

- Aumento de la eficiencia de l@s encargad@s de tomar decisiones.

- L@s usuari@s pueden acceder directamente a la información en línea, lo que contribuye a su capacidad para operar con mayor efectividad en las tareas rutinarias o no. Además, pueden tener a su disposición una gran cantidad de valiosa información multidimensional, presentada coherentemente como fuente única, confiable y disponible en sus estaciones de trabajo. Así mismo, l@s usuari@s tienen la facilidad de contar con herramientas que les son familiares para manipular y evaluar la información obtenida en el DW, tales como: hojas de cálculo, procesadores de texto, software de análisis de datos, software de análisis estadístico, reportes, tableros, etc.

- Permite la toma de decisiones estratégicas y tácticas.

2.6. Desventajas

A continuación se enumerarán algunas de las desventajas más comunes que se pueden presentar en la implementación de un Data Warehousing:

- Requiere una gran inversión, debido a que su correcta construcción no es tarea sencilla y consume muchos recursos, además, su misma implementación implica desde la adquisición de herramientas de consulta y análisis, hasta la capacitación de l@s usuari@s.

- Existe resistencia al cambio por parte de l@s usuari@s.

- Los beneficios del almacén de datos son apreciados en el mediano y largo plazo. Este punto deriva del anterior, y básicamente se refiere a que no tod@s l@s usuari@s confiarán en el DW en una primera instancia, pero sí lo harán una vez que comprueben su efectividad y ventajas. Además, su correcta utilización surge de la propia experiencia.

- Si se incluyen datos propios y confidenciales de clientes, proveedores, etc, el depósito de datos atentará contra la privacidad de los mismos, ya que cualquier usuari@ podrá tener acceso a ellos.

- Infravaloración de los recursos necesarios para la captura, carga y almacenamiento de los datos.

- Infravaloración del esfuerzo necesario para su diseño y creación.

- Incremento continuo de los requerimientos de l@s usuari@s.

- Subestimación de las capacidades que puede brindar la correcta utilización del DWH y de las herramientas de BI en general.

2.7. Redundancia

Debido a que el DW recibe información histórica de diferentes fuentes, sencillamente se podría suponer que existe una repetición de datos masiva entre el ambiente DW y el operacional. Por supuesto, este razonamiento es superficial y erróneo, de hecho, hay una mínima redundancia de datos entre ambos ambientes.

Para entender claramente lo antes expuesto, se debe considerar lo siguiente:

- Los datos del ambiente operacional se filtran antes de pertenecer al DW. Existen muchos datos que nunca ingresarán, ya que no conforman información necesaria o suficientemente relevante para la toma de decisiones.

- El horizonte de tiempo es muy diferente entre los dos ambientes.

- El almacén de datos contiene un resumen de la información que no se encuentra en el ambiente operacional.

- Los datos experimentan una considerable transformación, antes de ser cargados al DW. La mayor parte de los datos se alteran significativamente al ser seleccionados, consolidados y movidos al depósito.

En vista de estos factores, se puede afirmar que, la redundancia encontrada al cotejar los datos de ambos ambientes es mínima, ya que generalmente resulta en un porcentaje menor del 1 %.

2.8. Estructura

Los DW estructuran los datos de manera muy particular y existen diferentes niveles de esquematización y detalle que los delimitan.

En la siguiente figura se puede apreciar mejor su respectiva estructura.

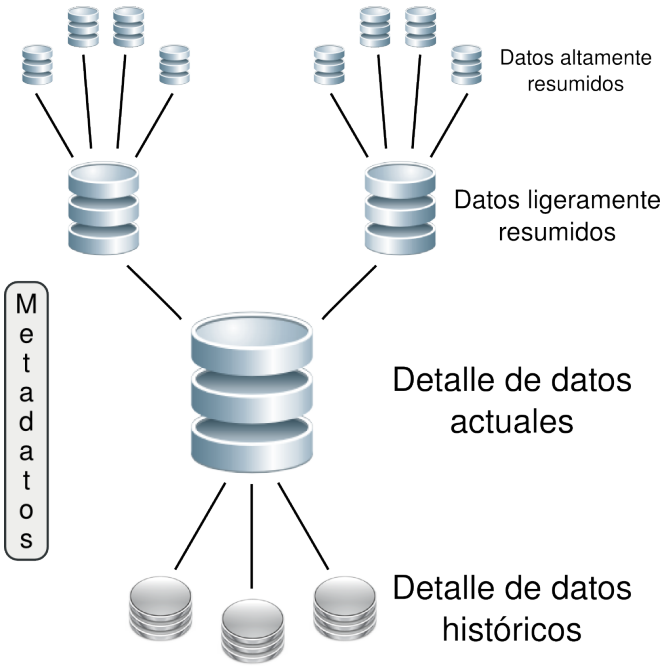


Figura 2.4: Data Warehouse, estructura.

Como se puede observar, los almacenes de datos están compuestos por diversos tipos de datos, que se organizan y dividen de acuerdo al nivel de detalle o granularidad que posean.

A continuación se explicarán cada uno de estos tipos de datos:

- Detalle de datos actuales: son aquellos que reflejan las ocurrencias más recientes. Generalmente se almacenan en disco, aunque su administración sea costosa y compleja, con el fin de conseguir que el acceso a la información sea sencillo y veloz, ya que son bastante voluminosos. Su gran tamaño se debe a que los datos residentes poseen el más bajo nivel de granularidad, o sea, se almacenan a nivel de detalle. Por ejemplo, aquí es donde se guardaría el detalle de una venta realizada en tal fecha.
- Detalle de datos históricos: representan aquellos datos antiguos, que no son frecuentemente consultados. También se almacenan a nivel de detalle, normalmente sobre alguna forma de almacenamiento externa, ya que son muy pesados y en adición a esto, no son requeridos con mucha periodicidad. Este tipo de datos son consistentes con los de Detalle de datos actuales. Por ejemplo, en este nivel, al igual que en el anterior, se encontraría el detalle de una venta realizada en tal fecha, pero con la particularidad de que el día en que se registró la venta debe ser lo suficientemente antigua, para que se considere como histórica.
- Datos ligeramente resumidos: son los que provienen desde un bajo nivel de detalle y sumarizan o agrupan los datos bajo algún criterio o condición de análisis. Habitualmente son almacenados en disco. Por ejemplo, en este caso se almacenaría la sumarización del detalle de las ventas realizadas en cada mes.
- Datos altamente resumidos: son aquellos que compactan aún más a los datos ligeramente resumidos. Se guardan en disco y son muy fáciles de acceder. Por ejemplo,

aquí se encontraría la sumarización de las ventas realizadas en cada año.

- Metadatos[5]: representan la información acerca de los datos. De muchas maneras se sitúa en una dimensión diferente al de otros datos del DW, ya que su contenido no es tomado directamente desde el ambiente operacional.

Estos diferentes niveles de detalle o granularidad, se obtienen a través de tablas de hechos agregadas y/o preagregadas[6].

2.9. Flujo de Datos

El DW posee un flujo de datos estándar y generalizado, el cual puede apreciarse mejor en la siguiente figura.

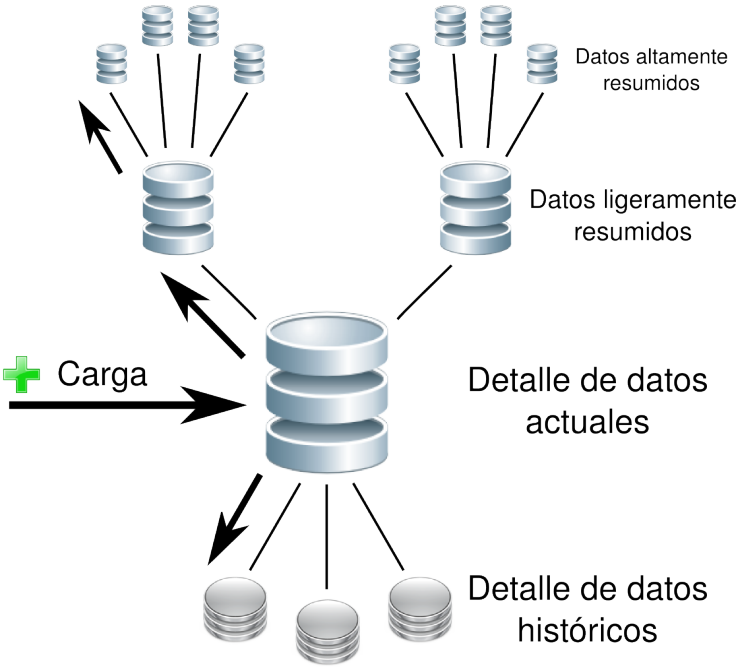


Figura 2.5: Data Warehouse, flujo de datos.

Cuando la información ingresa al depósito de datos se almacena a nivel de Detalle de datos actuales. Los datos permanecerán allí hasta que ocurra alguno de los tres eventos siguientes:

- Sean borrados del depósito de datos.
- Sean resumidos, ya sea a nivel de Datos ligeramente resumidos o a nivel de Datos altamente resumidos.
- Sean archivados a nivel de Detalle de datos históricos.

[5]Ver sección 3.4.9, en la página 49.
[6]Ver sección 3.4.3.1, en la página 32.

Capítulo 3

ARQUITECTURA DEL DATA WAREHOUSING

3.1. Introducción

En este punto y teniendo en cuenta que ya se han detallado claramente las características generales del Data Warehousing, se definirán y describirán todos los componentes que intervienen en su arquitectura o ambiente.

A través del siguiente gráfico se explicitará la estructura del Data Warehousing:

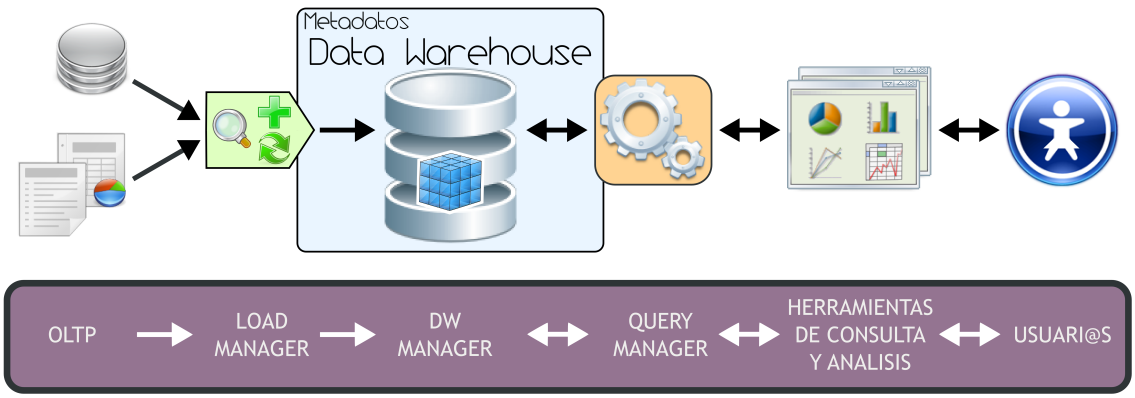


Figura 3.1: Data Warehousing, arquitectura.

Tal y como se puede apreciar, el ambiente esta formado por diversos elementos que interactúan entre sí y que cumplen una función específica dentro del sistema. Por ello es que al abordar la exposición de cada elemento se lo hará en forma ordenada y teniendo en cuenta su relación con las demás partes.

Básicamente, la forma de operar del esquema superior se resume de la siguiente manera:

- Los datos son extraídos desde aplicaciones, bases de datos, archivos, etc. Esta información generalmente reside en diferentes tipos de sistemas, orígenes y arqui-

tecturas y tienen formatos muy variados.

- Los datos son integrados, transformados y limpiados, para luego ser cargados en el DW.
- Principalmente, la información del DW se estructura en cubos multidimensionales, ya que estos preparan esta información para responder a consultas dinámicas con una buena performance. Pero también pueden utilizarse otros tipos de estructuras de datos para representar la información del DW, como por ejemplo Business Models.
- L@s usuari@s acceden a los cubos multidimensionales, Business Models (u otro tipo de estructura de datos) del DW utilizando diversas herramientas de consulta, exploración, análisis, reportes, etc.

A continuación se detallará cada uno de los componentes de la arquitectura del Data Warehousing, teniendo como referencia siempre el gráfico antes expuesto, pero resaltando el tema que se tratará.

3.2. OLTP

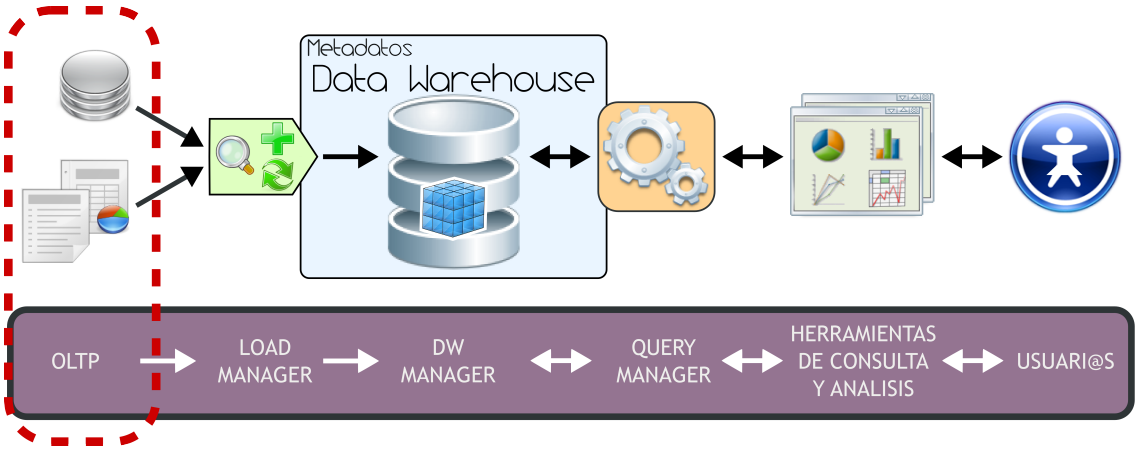


Figura 3.2: OLTP.

OLTP (On Line Transaction Processing), representa toda aquella información transaccional que genera la empresa en su accionar diario, además, de las fuentes externas con las que puede llegar a disponer.

Como ya se ha mencionado, estas fuentes de información, son de características muy disímiles entre sí, en formato, procedencia, función, etc.

Entre los OLTP más habituales que pueden existir en cualquier organización se encuentran:

- Archivos de textos.
- Hipertextos.

- Hojas de cálculos.
- Informes semanales, mensuales, anuales, etc.
- Bases de datos transaccionales.

3.3. Load Manager

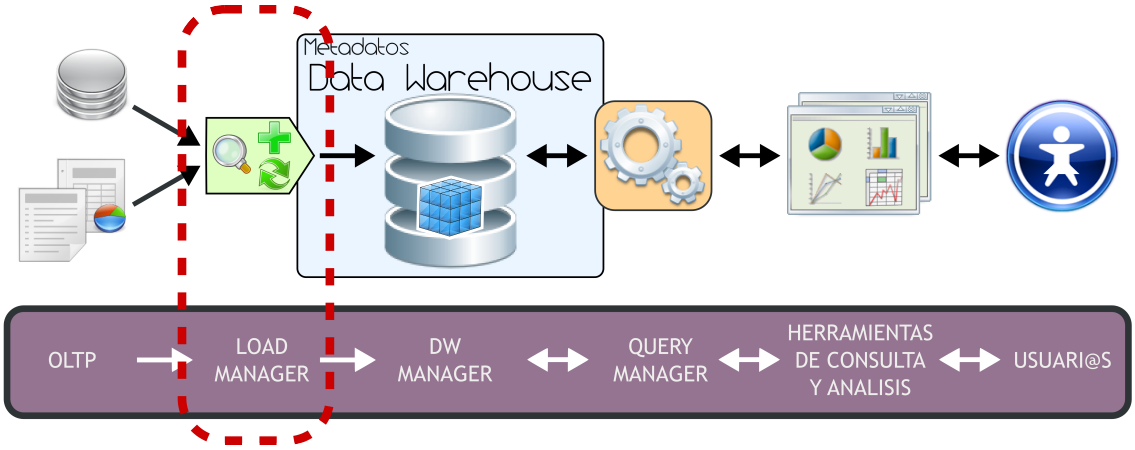


Figura 3.3: Load Manager.

Para poder extraer los datos desde los OLTP, para luego manipularlos, integrarlos y transformarlos, para posteriormente cargar los resultados obtenidos en el DW, es necesario contar con algún sistema que se encargue de ello. Precisamente, la Integración de Datos en quien cumplirá con tal fin.

La Integración de Datos agrupa una serie de técnicas y subprocesos que se encargan de llevar a cabo todas las tareas relacionadas con la extracción, manipulación, control, integración, depuración de datos, carga y actualización del DW, etc. Es decir, todas las tareas que se realizarán desde que se toman los datos de los diferentes OLTP hasta que se cargan en el DW.

Como se mencionó anteriormente cuando se trataron las características del DW[1], si bien los procesos ETL (Extracción, Transformación y Carga) son solo una de las muchas técnicas de la Integración de Datos, el resto de estas técnicas puede agruparse muy bien en sus diferentes etapas. Es decir, en el proceso de Extracción tendremos un grupo de técnicas enfocadas por ejemplo en tomar solo los datos indicados y mantenerlos en un almacenamiento intermedio; en el proceso de Transformación por ejemplo estarán aquellas técnicas que analizarán los datos para verificar que sean correctos y válidos; en el proceso de Carga de Datos se agruparán por ejemplo técnicas propias de la carga y actualización del DW.

A continuación, se detallará cada una de estas etapas, se expondrá cuál es el proceso que llevan a cabo los ETL y se enumerarán cuáles son sus principales tareas.

[1]Ver sección 2.3.2, en la página 11.

3.3.1. Extracción

Es aquí, en donde, basándose en las necesidades y requisitos de l@s usuari@s, se exploran las diversas fuentes OLTP que se tengan a disposición, y se extrae la información que se considere relevante al caso.

Si los datos operacionales residen en un SGBD Relacional, el proceso de extracción se puede reducir a, por ejemplo, consultas en SQL o rutinas programadas. En cambio, si se encuentran en un sistema no convencional o fuentes externas, ya sean textuales, hipertextuales, hojas de cálculos, etc, la obtención de los mismos puede ser un tanto más dificultoso, debido a que, por ejemplo, se tendrán que realizar cambios de formato y/o volcado de información a partir de alguna herramienta específica.

Una vez que los datos son seleccionados y extraídos, se guardan en un almacenamiento intermedio, lo cual permite, entre otras ventajas:

- Manipular los datos sin interrumpir ni paralizar los OLTP, ni tampoco el DW.
- No depender de la disponibilidad de los OLTP.
- Almacenar y gestionar los metadatos que se generarán en los procesos ETL.
- Facilitar la integración de las diversas fuentes, internas y externas.

> El almacenamiento intermedio constituye en la mayoría de los casos una base de datos en donde la información puede ser almacenada por ejemplo en tablas auxiliares, tablas temporales, etc. Los datos de estas tablas serán los que finalmente (luego de su correspondiente transformación) poblarán el DW.

3.3.2. Transformación

Esta función es la encargada de convertir aquellos datos inconsistentes en un conjunto de datos compatibles y congruentes, para que puedan ser cargados en el DW. Estas acciones se llevan a cabo, debido a que pueden existir diferentes fuentes de información, y es vital conciliar un formato y forma única, definiendo estándares, para que todos los datos que ingresarán al DW estén integrados.

Los casos más comunes en los que se deberá realizar integración, son los siguientes:

- Codificación.
- Medida de atributos.
- Convenciones de nombramiento.
- Fuentes múltiples.

Además de lo antes mencionado, esta función se encarga de realizar, entre otros, los procesos de Limpieza de Datos (Data Cleansing) y Calidad de Datos.

3.3.2.1. Codificación

Una inconsistencia muy típica que se encuentra al intentar integrar varias fuentes de datos, es la de contar con más de una forma de codificar un atributo en común. Por ejemplo, en el campo “estado”, algun@s diseñador@s completan su valor con “0” y “1”, otros con “Apagado” y “Encendido”, otros con “off” y “on”, etc. Lo que se debe realizar en estos casos, es seleccionar o recodificar estos atributos, para que cuando la información

llegue al DW, esté integrada de manera uniforme.

En la siguiente figura, se puede apreciar que de varias formas de codificar se escoge una, entonces cuando surge una codificación diferente a la seleccionada, se procede a su transformación.

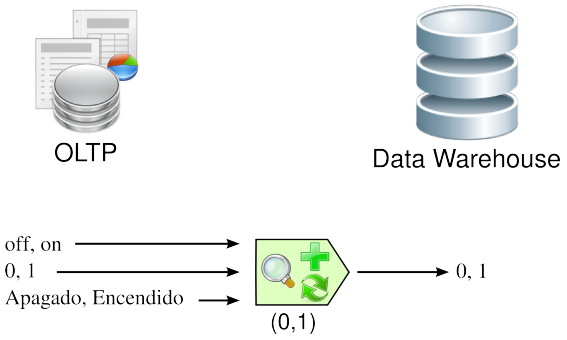


Figura 3.4: Transformación: codificación.

3.3.2.2. Medida de atributos

Los tipos de unidades de medidas utilizados para representar los atributos de una entidad, varían considerablemente entre sí, a través de los diferentes OLTP. Por ejemplo, al registrar la longitud de un producto determinado, de acuerdo a la aplicación que se emplee para tal fin, las unidades de medidas pueden ser explicitadas en centímetros, metros, pulgadas, etc.

En esta ocasión, se deberán estandarizar las unidades de medidas de los atributos, para que todas las fuentes de datos expresen sus valores de igual manera. Los algoritmos que resuelven estas inconsistencias son generalmente los más complejos.

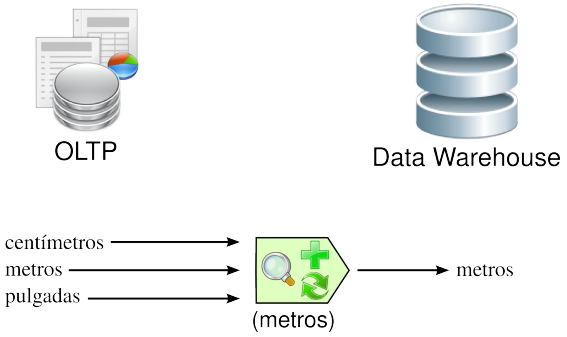


Figura 3.5: Transformación: medida de atributos.

3.3.2.3. Convenciones de nombramiento

Usualmente, un mismo atributo es nombrado de diversas maneras en los diferentes OLTP. Por ejemplo, al referirse al nombre del proveedor, puede hacerse como “nombre”, “razón_social”, “proveedor”, etc. Aquí, se debe utilizar la convención de nombramiento que para l@s usuari@s sea más comprensible.

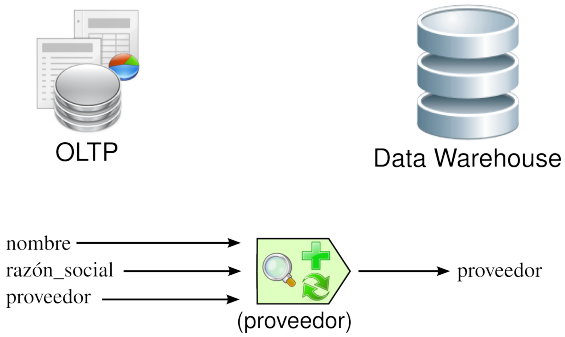


Figura 3.6: Transformación: convenciones de nombramiento.

3.3.2.4. Fuentes múltiples

Un mismo elemento puede derivarse desde varias fuentes. En este caso, se debe elegir aquella fuente que se considere más fiable y apropiada.

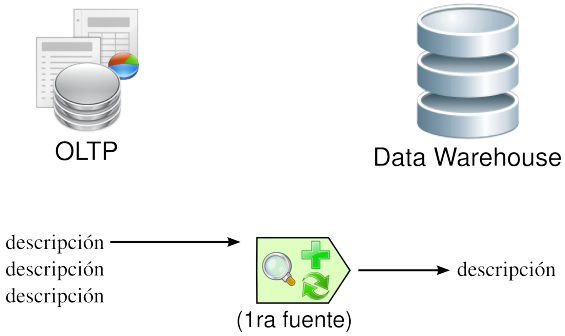


Figura 3.7: Transformación: fuentes múltiples.

3.3.2.5. Limpieza de datos

Su objetivo principal es el de realizar distintos tipos de acciones contra el mayor número de datos erróneos, inconsistentes e irrelevantes.

- Las acciones más típicas que se pueden llevar a cabo al encontrarse con Datos Anómalos (Outliers) son:
  - Ignorarlos.
  - Eliminar la columna.

- Filtrar la columna.
- Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
- Reemplazar el valor.
- Discretizar los valores de las columnas. Por ejemplo de 1 a 2, poner “bajo”; de 3 a 7, “óptimo”; de 8 a 10, “alto”. Para que los outliers caigan en “bajo” o en “alto” sin mayores problemas.

- Las acciones que suelen efectuarse contra Datos Faltantes (Missing Values) son:

  - Ignorarlos.
  - Eliminar la columna.
  - Filtrar la columna.
  - Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
  - Reemplazar el valor.
  - Esperar hasta que los datos faltantes estén disponibles.

Un punto muy importante que se debe tener en cuenta al elegir alguna acción, es el de identificar el por qué de la anomalía, para luego actuar en consecuencia, con el fin de evitar que se repitan, agregándole de esta manera más valor a los datos de la organización. Se puede dar que en algunos casos, los valores faltantes sean inexistentes, ya que por ejemplo, l@s nuev@ asociad@s o client@s, no poseerán consumo medio del último año.

3.3.3. Carga

1. Esta función se encarga, por un lado de realizar las tareas relacionadas con:

   - Carga Inicial (Initial Load).
   - Actualización o mantenimiento periódico (siempre teniendo en cuenta un intervalo de tiempo predefinido para tal operación).

   La carga inicial, se refiere precisamente a la primera carga de datos que se le realizará al DW. Por lo general, esta tarea consume un tiempo bastante considerable, ya que se deben insertar registros que han sido generados aproximadamente, y en casos ideales, durante más de cinco años.

   Los mantenimientos periódicos mueven pequeños volúmenes de datos, y su frecuencia está dada en función del gránulo del DW y los requerimientos de l@s usuari@s. El objetivo de esta tarea es añadir al depósito aquellos datos nuevos que se fueron generando desde el último refresco.

   Antes de realizar una nueva actualización, es necesario identificar si se han producido cambios en las fuentes originales de los datos recogidos, desde la fecha del último mantenimiento, a fin de no atentar contra la consistencia del DW. Para efectuar esta operación, se pueden realizar las siguientes acciones:

   - Cotejar las instancias de los OLTP involucrados.
   - Utilizar disparadores en los OLTP.
   - Recurrir a Marcas de Tiempo (Time Stamp), en los registros de los OLTP.
   - Comparar los datos existentes en los dos ambientes (OLTP y DW).

- Hacer uso de técnicas mixtas.

Si este control consume demasiado tiempo y esfuerzo, o simplemente no puede llevarse a cabo por algún motivo en particular, existe la posibilidad de cargar el DW desde cero, este proceso se denomina Carga Total (Full Load).

Ingresarán al DW, para su carga y/o actualización:

- Aquellos datos que han sido transformados y que residen en el almacenamiento intermedio.
- Aquellos datos de los OLTP que tienen correspondencia directa con el depósito de datos.

Se debe tener en cuenta, que los datos antes de moverse al almacén de datos, deben ser analizados con el propósito de asegurar su calidad, ya que este es un factor clave, que no debe dejarse de lado.

2. Por otra parte, el proceso de Carga tiene la tarea de mantener la estructura del DW, y trata temas relacionados con:
   - Relaciones muchos a muchos[2].
   - Claves Subrogadas[3].
   - Dimensiones Lentamente Cambiantes[4].
   - Dimensiones Degeneradas[5].

3.3.4. Proceso ETL

A continuación, se explicará en síntesis el accionar del proceso ETL, y cuál es la relación existente entre sus diversas funciones. En la siguiente figura se puede apreciar mejor lo antes descrito:

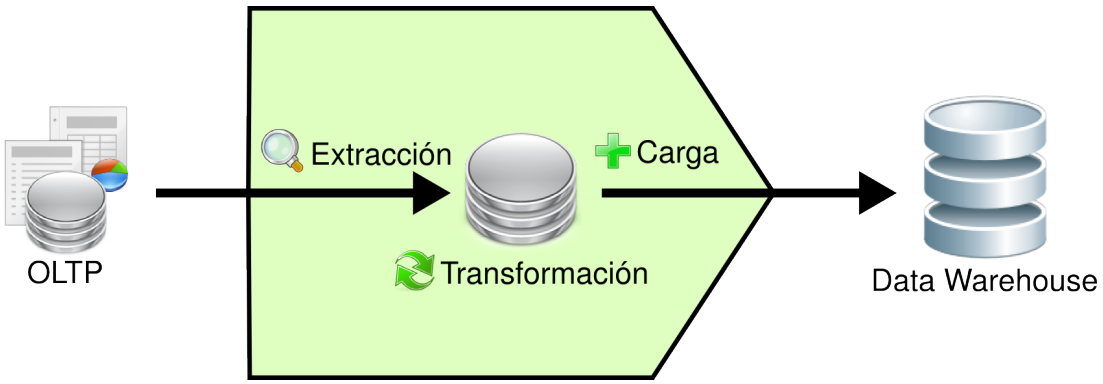


Figura 3.8: Proceso ETL.

Los pasos que se siguen son:

[2]Ver sección 6.12, en la página 123.
[3]Ver sección 6.13, en la página 124.
[4]Ver sección 6.14, en la página 125.
[5]Ver sección 6.15, en la página 129.

- Se extraen los datos relevantes desde los OLTP y se depositan en un almacenamiento intermedio.
- Se integran y transforman los datos, para evitar inconsistencias.
- Se cargan los datos desde el almacenamiento intermedio hasta el DW.

3.4. Data Warehouse Manager

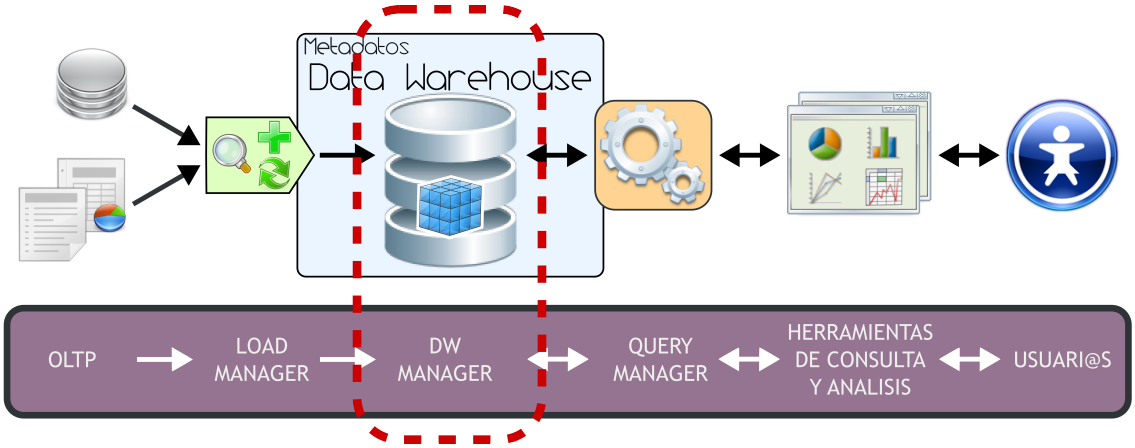


Figura 3.9: Data Warehouse Manager.

El DW Manager presenta las siguientes características y funciones principales:

- Se constituye típicamente al combinar un SGBD con software y aplicaciones dedicadas.
- Almacena los datos de forma multidimensional[6], es decir, a través de tablas de hechos[7] y tablas de dimensiones[8].
- Gestiona las diferentes estructuras de datos que se construyan o describan sobre el DW, como Cubos Multidimensionales[9], Business Models[10], etc.
- Gestiona y mantiene metadatos.

Además, el DW Manager se encarga de:

- Transformar e integrar los datos fuentes y del almacenamiento intermedio en un modelo adecuado para la toma de decisiones.
- Realizar todas las funciones de definición y manipulación del depósito de datos, para poder soportar todos los procesos de gestión del mismo.

[6] Ver sección 3.4.1, en la página 28.
[7] Ver sección 3.4.3, en la página 30.
[8] Ver sección 3.4.2, en la página 28.
[9] Ver sección 3.4.4, en la página 33.
[10] Ver sección 4.5, en la página 76.

- Ejecutar y definir las políticas de particionamiento[11]. El objetivo de realizar esto, es conseguir una mayor eficiencia y performance en las consultas al no tener que manejar todo el grueso de los datos. Esta política debe aplicarse sobre la tabla de hechos que, como se explicará más adelante, es en la que se almacena toda la información que será analizada.
- Realizar copias de resguardo incrementales o totales de los datos del DW.

3.4.1. Base de datos multidimensional

Una base de datos multidimensional es una base de datos en donde su información se almacena en forma multidimensional, es decir, a través de tablas de hechos y tablas de dimensiones.

Proveen una estructura que permite, a través de la creación y consulta a una estructura de datos determinada (cubo multidimensional[12], Business Model[13], etc), tener acceso flexible a los datos, para explorar y analizar sus relaciones, y consiguientes resultados.

Las bases de datos multidimensionales implican tres variantes posibles de modelamiento, que permiten realizar consultas de soporte de decisión:

- Esquema en estrella[14] (Star Scheme).
- Esquema copo de nieve[15] (Snowflake Scheme).
- Esquema constelación[16] o copo de estrellas (Starflake Scheme).

Los mencionados esquemas pueden ser implementados de diversas maneras, que, independientemente al tipo de arquitectura, requieren que toda la estructura de datos este desnormalizada o semi desnormalizada, para evitar desarrollar uniones (Join) complejas para acceder a la información, con el fin de agilizar la ejecución de consultas. Los diferentes tipos de implementación son los siguientes:

- Relacional – ROLAP[17].
- Multidimensional – MOLAP[18].
- Híbrido – HOLAP[19].

3.4.2. Tablas de Dimensiones

Las tablas de dimensiones definen como están los datos organizados lógicamente y proveen el medio para analizar el contexto del negocio. Contienen datos cualitativos.

Representan los aspectos de interés, mediante los cuales l@s usuari@s podrán filtrar y manipular la información almacenada en la tabla de hechos.

En la siguiente figura se pueden apreciar algunos ejemplos:

[11] Ver sección 4.4, en la página 76.
[12] Ver sección 3.4.4, en la página 33.
[13] Ver sección 4.5, en la página 76.
[14] Ver sección 3.4.5.1, en la página 37.
[15] Ver sección 3.4.5.2, en la página 39.
[16] Ver sección 3.4.5.3, en la página 40.
[17] Ver sección 3.4.7.1, en la página 42.
[18] Ver sección 3.4.7.2, en la página 43.
[19] Ver sección 3.4.7.3, en la página 44.

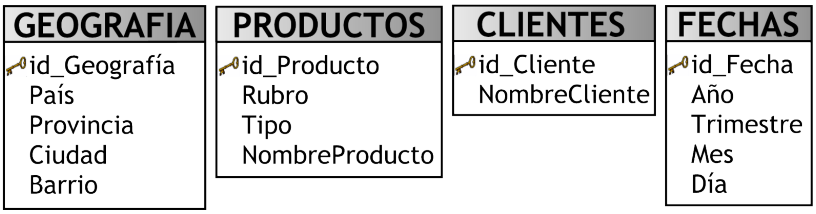


Figura 3.10: Tablas de Dimensiones.

Como se puede observar, cada tabla posee un identificador único y al menos un campo o dato de referencia que describe los criterios de análisis relevantes para la organización, estos son por lo general de tipo texto.

Los datos dentro de estas tablas, que proveen información del negocio o que describen alguna de sus características, son llamados datos de referencia.

Más detalladamente, cada tabla de dimensión podrá contener los siguientes campos:

- Clave principal o identificador único.
- Clave foráneas.
- Datos de referencia primarios: datos que identifican la dimensión. Por ejemplo: nombre del cliente.
- Datos de referencia secundarios: datos que complementan la descripción de la dimensión. Por ejemplo: e-mail del cliente, fax del cliente, etc.

Usualmente la cantidad de tablas de dimensiones, aplicadas a un tema de interés en particular, varían entre tres y quince.

Debe tenerse en cuenta, que no siempre la clave primaria del OLTP, se corresponde con la clave primaria de la tabla de dimensión relacionada. Es recomendable manejar un sistema de claves en el DW (Claves Subrogadas[20]) totalmente diferente al de los OLTP, ya que si estos últimos son recodificados, el almacén quedaría inconsistente y debería ser poblado nuevamente en su totalidad.

3.4.2.1. Tabla de Dimensión Tiempo

En un DW, la creación y el mantenimiento de una tabla de dimensión Tiempo es obligatoria, y la definición de granularidad y estructuración de la misma depende de la dinámica del negocio que se este analizando. Toda la información dentro del depósito, como ya se ha explicado, posee su propio sello de tiempo que determina la ocurrencia de un hecho específico, representando de esta manera diferentes versiones de una misma situación.

[20] Ver sección 6.13, en la página 124.

Es importante tener en cuenta que la dimensión tiempo no es sola una secuencia cronológica representada de forma numérica, sino que mantiene niveles jerárquicos especiales que inciden notablemente en las actividades de la organización. Esto se debe a que l@s usuari@s podrán por ejemplo analizar las ventas realizadas teniendo en cuenta el día de la semana en que se produjeron, quincena, mes, trimestre, semestre, año, estación, etc.

Existen muchas maneras de diseñar esta tabla, y en adición a ello no es una tarea sencilla de llevar a cabo. Por estas razones se considera una buena práctica evaluar con cuidado la temporalidad de los datos, la forma en que trabaja la organización, los resultados que se esperan obtener del almacén de datos relacionados con una unidad de tiempo y la flexibilidad que se desea obtener de dicha tabla.

Así mismo, si se requiere analizar los datos por Fecha (año, mes, día, etc) y por Hora (hora, minuto, segundo, etc), lo más recomendable es confeccionar dos tablas de dimensión Tiempo; una contendrá los datos referidos a la Fecha y la otra los referidos a la Hora.

> Si bien, el lenguaje SQL ofrece funciones del tipo DATE, en la tabla de dimensión Tiempo, se modelan y presentan datos temporales que no pueden calcularse a través de consultas SQL, lo cual le añade una ventaja más.
>
> Es conveniente mantener en la tabla de dimensión Tiempo un campo que se refiera al día Juliano. El día juliano se representa a través de un número secuencial e identifica unívocamente cada día. Mantener este campo permitirá la posibilidad de realizar consultas que involucren condiciones de filtrado de fechas desde-hasta, mayor que, menor que, etc, de manera sencilla e intuitiva; ya que si por ejemplo a partir de tal fecha se desea analizar los datos de los 81 días siguientes, el valor "desde" sería el día Juliano de la fecha en cuestión y el valor "hasta" sería igual a "desde" más 81.

3.4.3. Tablas de Hechos

Las tablas de hechos contienen, precisamente, los hechos que serán utilizados por l@s analistas de negocio para apoyar el proceso de toma de decisiones. Contienen datos cuantitativos.

Los hechos son datos instantáneos en el tiempo, que son filtrados, agrupados y explorados a través de condiciones definidas en las tablas de dimensiones.

Los datos presentes en las tablas de hechos constituyen el volumen de la bodega, y pueden estar compuestos por millones de registros dependiendo de su granularidad y antigüedad de la organización. Los más importantes son los de tipo numérico.

El registro del hecho posee una clave primaria que está compuesta por las claves primarias de las tablas de dimensiones relacionadas a este.

En la siguiente imagen se puede apreciar un ejemplo de lo antes mencionado:

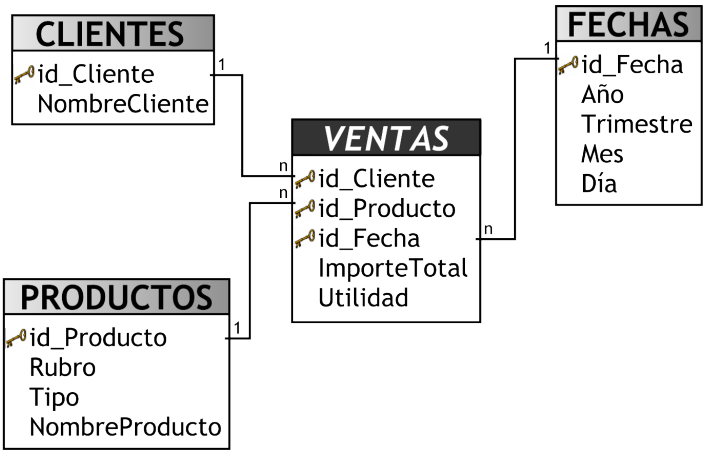


Figura 3.11: Tablas de Hechos.

Como se muestra en la figura anterior, la tabla de hechos “VENTAS” se ubica en el centro, e irradiando de ella se encuentran las tablas de dimensiones “CLIENTES”, “PRODUCTOS” y “FECHAS”, que están conectadas mediante sus claves primarias. Es por ello que la clave primaria de la tabla de hechos es la combinación de las claves primarias de sus dimensiones. Los hechos en este caso son “ImporteTotal” y “Utilidad”.

A continuación, se entrará más en detalle sobre la definición de un hecho, también llamado dato agregado:

- Los hechos son aquellos datos que residen en una tabla de hechos y que son utilizados para crear indicadores3.4.4.1, a través de sumarizaciones preestablecidas al momento de crear un cubo multidimensional, Business Model, etc. Debido a que una tabla de hechos se encuentra interrelacionada con sus respectivas tablas de dimensiones, permite que los hechos puedan ser accedidos, filtrados y explorados por los valores de los campos de estas tablas de dimensiones, obteniendo de este modo una gran capacidad analítica.

  Las sumarizaciones no están referidas solo a sumas, sino también a promedios, mínimos, máximos, totales por sector, porcentajes, fórmulas predefinidas, etc, dependiendo de los requerimientos de información del negocio.

Para ejemplificar este nuevo concepto de hechos, se enumerarán algunos que son muy típicos y fáciles de comprender:

- ImporteTotal = precioProducto * cantidadVendida
- Rentabilidad = utilidad / PN
- CantidadVentas = cantidad
- PromedioGeneral = AVG(notasFinales)

  A la izquierda de la igualdad se encuentran los hechos; a la derecha los campos de los OLTP que son utilizados para representarlos. En el último ejemplo se realiza un precálculo para establecer el hecho.

Existen dos tipos de hechos, los básicos y los derivados, a continuación se detallará cada uno de ellos, teniendo en cuenta para su ejemplificación la siguiente tabla de hechos:

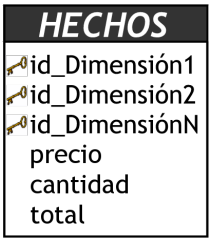


Figura 3.12: Hechos básicos y derivados.

- Hechos básicos: son los que se encuentran representados por un campo de una tabla de hechos. Los campos ”precio” y ”cantidad” de la tabla anterior son hechos básicos.

- Hechos derivados: son los que se forman al combinar uno o más hechos con alguna operación matemática o lógica y que también residen en una tabla de hechos. Estos poseen la ventaja de almacenarse previamente calculados, por lo cual pueden ser accedidos a través de consultas SQL sencillas y devolver resultados rápidamente, pero requieren más espacio físico en el DW, además de necesitar más tiempo de proceso en los ETL que los calculan. El campo ”total” de la tabla anterior en un hecho derivado, ya que se conforma de la siguiente manera:

  - total = precio * cantidad

    Los campos ”precio” y ”cantidad”, también pertenecen a la tabla ”HECHOS”. Cabe resaltar, que no es necesario que los hechos derivados se compongan únicamente con hechos pertenecientes a una misma tabla.

Los hechos son gestionados con el principal objetivo de que se construyan indicadores basados en ellos, a través de la creación de un cubo multidimensional, Business Model, u otra estructura de datos.

3.4.3.1. Tablas de hechos agregadas y preagregadas

Las tablas de hechos agregadas y preagregadas se utilizan para almacenar un resumen de los datos, es decir, se guardan los datos en niveles de granularidad superior a los que inicialmente fueron obtenidos y/o gestionados.

Para obtener tablas agregadas o preagregadas, es necesario establecer un criterio por el cual realizar el resumen. Por ejemplo, esto ocurre cuando se desea obtener información de ventas sumarizadas por mes.

Cada vez que se requiere que los datos en una consulta se presenten en un nivel de granularidad superior al que se encuentran alojados en el Data Warehouse, se debe llevar a cabo un proceso de agregación.

El objetivo general de las tablas de hechos agregadas y preagregadas es similar, pero cada una de ellas tiene una manera de operar diferente:

- Tablas de hechos agregadas: se generan luego de que se procesa la consulta correspondiente a la tabla de hechos que se resumirá. En general, la agregación se produce dinámicamente a través de una instrucción SQL que incluya sumarizaciones.
- Tablas de hechos preagregadas: se generan antes de que se procese la consulta correspondiente a la tabla de hechos que se resumirá. De esta manera, la consulta se realiza contra una tabla que ya fue previamente sumarizada. Habitualmente, estas sumarizaciones se calculan a través de procesos ETL.

Las tablas de hechos preagregadas cuentan con los siguientes beneficios:

- Reduce la utilización de recursos de hardware que normalmente son incurridos en el cálculo de las sumarizaciones.
- Reduce el número de registros que serán analizados por el usuario.
- Reduce el tiempo utilizado en la generación de consultas por parte del usuario.

Las tablas de hechos preagregadas son muy útiles en los siguientes casos generales:

- Cuando los datos a nivel detalle (menor nivel granular) son innecesarios y/o no son requeridos.
- Cuando una consulta sumarizada a determinado nivel de granularidad es solicitado con mucha frecuencia.
- Cuando los datos son muy abundantes, y las consultas demoran en ser procesadas demasiado tiempo.

Como contrapartida, las tablas de hechos preagregadas presentan una serie de desventajas:

- Requieren que se mantengan y gestionen nuevos procesos ETL.
- Demandan espacio de almacenamiento extra en el depósito de datos.

3.4.4. Cubo Multidimensional: introducción

Si bien existen diversas estructuras de datos, a través de las cuales se puede representar los datos del DW, solamente se entrará en detalle acerca de los cubos multidimensionales, por considerarse que esta estructura de datos es una de las más utilizadas y cuyo funcionamiento es el más complejo de entender.

Un cubo multidimensional o hipercubo, representa o convierte los datos planos que se encuentran en filas y columnas, en una matriz de N dimensiones.

Los objetos más importantes que se pueden incluir en un cubo multidimensional, son los siguientes:

- Indicadores[21]: sumarizaciones que se efectúan sobre algún hecho o expresiones basadas en sumarizaciones, pertenecientes a una tabla de hechos.
- Atributos[22]: campos o criterios de análisis, pertenecientes a tablas de dimensiones.
- Jerarquías[23]: representa una relación lógica entre dos o más atributos.

[21] Ver sección 3.4.4.1, en la página 34.
[22] Ver sección 3.4.2, en la página 28.
[23] Ver sección 3.4.4.3, en la página 35.

De esta manera en un cubo multidimensional, los atributos existen a lo largo de varios ejes o dimensiones, y la intersección de las mismas representa el valor que tomará el indicador que se está evaluando.

En la siguiente representación matricial se puede ver más claramente lo que se acaba de decir.

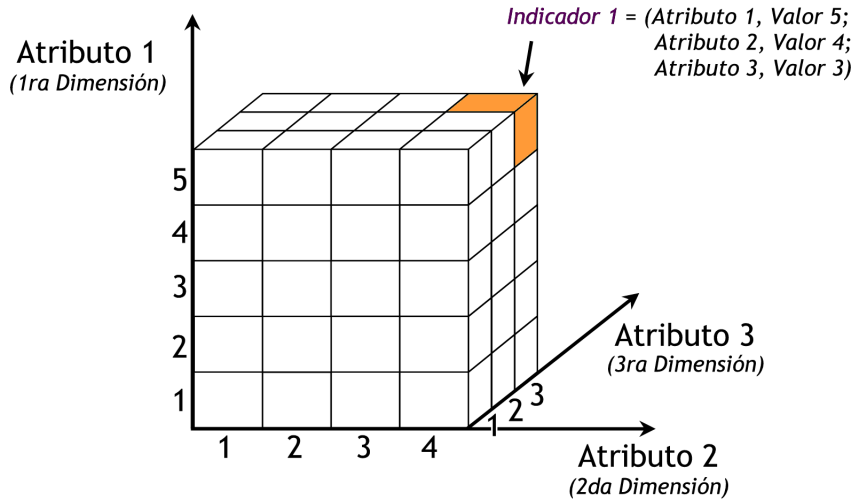


Figura 3.13: Cubo multidimensional.

Para la creación del cubo de la figura anterior, se definieron tres Atributos (“Atributo 1”, “Atributo 2” y “Atributo 3”) y se definió un Indicador (“Indicador 1”). Entonces el cubo quedo compuesto por 3 dimensiones o ejes (una por cada Atributo), cada una con sus respectivos valores asociados. También, se ha seleccionado una intersección al azar para demostrar la correspondencia con los valores de las Atributos. En este caso, el indicador “Indicador 1”, representa el cruce del Valor “5” de “Atributo 1”, con el Valor “4” de “Atributo 2” y con el Valor “3” de “Atributo 3”.

Se puede observar, que el resultado del análisis está dado por los cruces matriciales de acuerdo a los valores de las dimensiones seleccionadas.

Más específicamente, para acceder a los datos del DW, se pueden ejecutar consultas sobre algún cubo multidimensional previamente definido. Dicho cubo debe incluir entre otros objetos: indicadores, atributos, jerarquías, etc, basados en los campos de las tablas de dimensiones y de hechos, que se deseen analizar. De esta manera, las consultas son respondidas con gran performance, minimizando al máximo el tiempo que se hubiese incurrido en realizar dicha consulta sobre una base de datos transaccional.

3.4.4.1. Indicadores

Los indicadores son sumarizaciones efectuadas sobre algún hecho o expresiones basadas en sumarizaciones, que serán incluidos en algún cubo multidimensional, con el fin de analizar los datos almacenados en el DW. El valor que estos adopten estará condicionado por los atributos/jerarquías que se utilicen para analizarlos.

Los indicadores, además de hechos, pueden estar compuestos por otros indicadores, pero no ambos simultáneamente. Pueden utilizarse para su creación funciones de sumarización (suma, conteo, promedio, etc), funciones matemáticas, estadísticas, operadores matemáticos y lógicos.

3.4.4.2. Atributos

Los atributos constituyen los criterios de análisis que se utilizarán para analizar los indicadores dentro de un cubo multidimensional. Los mismos se basan, en su gran mayoría, en los campos de las tablas de dimensiones y/o expresiones.

Dentro de un cubo multidimensional, los atributos son los ejes del mismo.

3.4.4.3. Jerarquías

Una jerarquía representa una relación[24] lógica entre dos o más atributos pertenecientes a un cubo multidimensional; siempre y cuando posean su correspondiente relación "padre-hijo".

Las jerarquías poseen las siguientes características:

- Pueden existir varias en un mismo cubo.
- Están compuestas por dos o más niveles.
- Se tiene una relación "1-n" o "padre-hijo" entre atributos consecutivos de un nivel superior y uno inferior.

Por lo general, las jerarquías pueden identificarse fácilmente, debido a que existen relaciones "1-n" o "padre-hijo" entre los propios atributos de un cubo.

La principal ventaja de manejar jerarquías, reside en poder analizar los datos desde su nivel más general al más detallado y viceversa, al desplazarse por los diferentes niveles.

La siguiente figura muestra un pequeño ejemplo de lo recién expuesto:

[24] Ver sección 3.4.4.4, en la página 36.

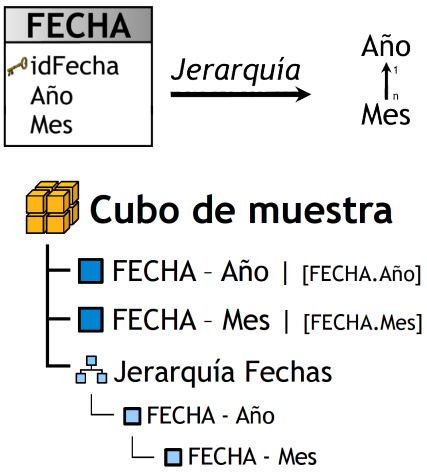


Figura 3.14: Jerarquía Fechas.

Aquí, se puede apreciar claramente como se construye una jerarquía:

1. Se crearon dos atributos, ”FECHA - Año” y ”FECHA - Mes”, los cuales están constituidos de la siguiente manera:

   - ”FECHA - Año” = FECHA.Año
   - ”FECHA - Mes” = FECHA.Mes

   A la izquierda de la igualdad se encuentra el nombre del atributo; a la derecha el nombre del campo de la tabla de dimensión “FECHA”.

2. Se creó la jerarquía llamada ”Jerarquía Fechas”, en la cual se colocó el atributo más general en la cabecera y se comenzó a disgregar los niveles hacia abajo. En este caso, la figura se explica como sigue:

   - Un mes del año pertenece solo a un año. Un año puede poseer uno o más meses del año.

3.4.4.4. a) Relación

Una relación representa la forma en que dos atributos interactúan dentro de una jerarquía. Existen básicamente dos tipos de relaciones:

- Explícitas: son las más comunes y se pueden modelar a partir de atributos directos y están en línea continua de una jerarquía, por ejemplo, un país posee una o más provincias y una provincia pertenece solo a un país.

- Implícitas: son las que ocurren en la vida real, pero su relación no es de vista directa, por ejemplo, una provincia tiene uno o más ríos, pero un río pertenece a una o más provincias. Como se puede observar, este caso se trata de una relación muchos a muchos[25].

[25] Ver sección 6.12, en la página 123.

3.4.4.5. b) Granularidad

La granularidad representa el nivel de detalle al que se desea almacenar la información sobre el negocio que se esté analizando. Por ejemplo, los datos referentes a ventas o compras realizadas por una empresa, pueden registrarse día a día, en cambio, los datos pertinentes a pagos de sueldos o cuotas de socios, podrán almacenarse a nivel de mes.

Mientras mayor sea el nivel de detalle de los datos, se tendrán mayores posibilidades analíticas, ya que los mismos podrán ser resumidos o sumarizados. Es decir, los datos que posean granularidad fina (nivel de detalle) podrán ser resumidos hasta obtener una granularidad media o gruesa. No sucede lo mismo en sentido contrario, ya que por ejemplo, los datos almacenados con granularidad media podrán resumirse, pero no tendrán la facultad de ser analizados a nivel de detalle. O sea, si la granularidad con que se guardan los registros es a nivel de día, estos datos podrán sumarizarse por semana, mes, semestre y año, en cambio, si estos registros se almacenan a nivel de mes, podrán sumarizarse por semestre y año, pero no lo podrán hacer por día y semana.

3.4.5. Tipos de modelamiento de un DW

3.4.5.1. Esquema en Estrella

El esquema en estrella, consta de una tabla de hechos central y de varias tablas de dimensiones relacionadas a esta, a través de sus respectivas claves. En la siguiente figura se puede apreciar un esquema en estrella estándar:

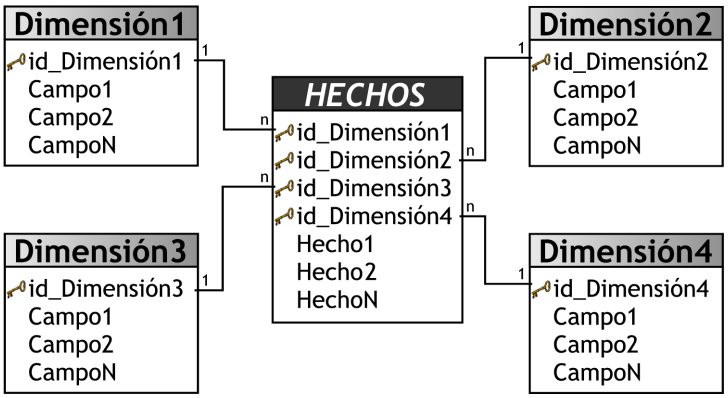


Figura 3.15: Esquema en Estrella.

El modelo ejemplificado cuando se abordo el tema de las tablas de hechos, es un esquema en estrella, por lo cual se lo volverá a mencionar para explicar sus cualidades.

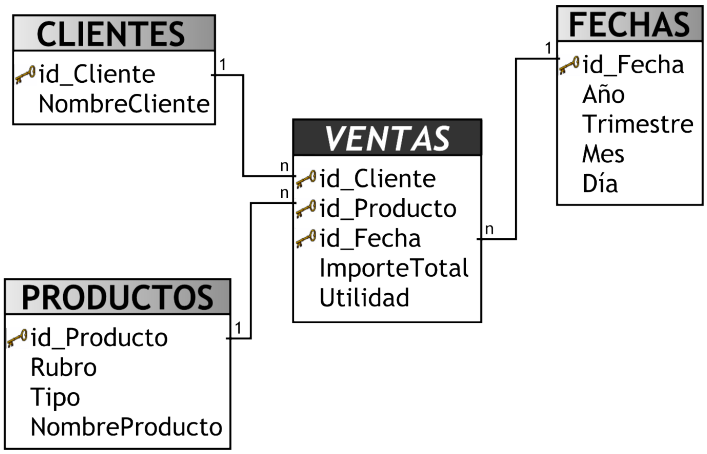


Figura 3.16: Esquema en Estrella, ejemplo.

Este modelo debe estar totalmente desnormalizado, es decir que no puede presentarse en tercera forma normal (3ra FN), es por ello que por ejemplo, la tabla de dimensión “PRODUCTOS” contiene los campos “Rubro”, “Tipo” y “NombreProducto”. Si se normaliza esta tabla, se obtendrá el siguiente resultado:

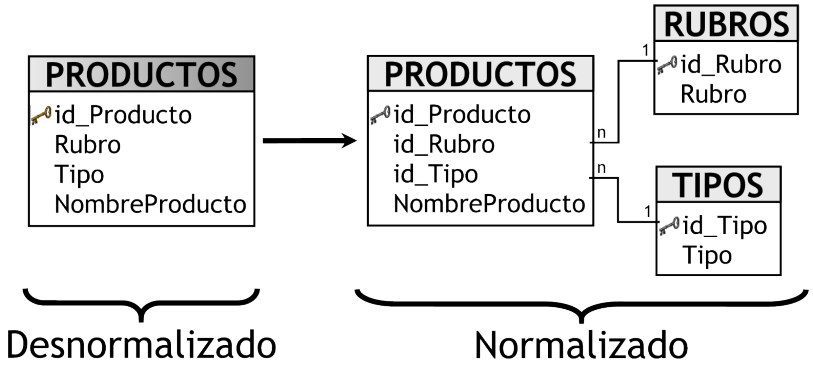


Figura 3.17: Desnormalización.

Cuando se normaliza, se pretende eliminar la redundancia, la repetición de datos y que las claves sean independientes de las columnas, pero en este tipo de modelos se requiere no evitar precisamente esto.

Las ventajas que trae aparejada la desnormalización, son las de obviar uniones (Join) entre las tablas cuando se realizan consultas, procurando así un mejor tiempo de respuesta y una mayor sencillez con respecto a su utilización. El punto en contra, es que se genera un cierto grado de redundancia, pero el ahorro de espacio no es significativo.

El esquema en estrella es el más simple de interpretar y optimiza los tiempos de respuesta ante las consultas de l@s usuari@s. Este modelo es soportado por casi todas las herramientas de consulta y análisis, y los metadatos son fáciles de documentar y man-

tener, sin embargo es el menos robusto para la carga y es el más lento de construir.

A continuación se destacarán algunas características de este modelo, que ayudarán a comprender mejor el por qué de sus ventajas:

- Posee los mejores tiempos de respuesta.
- Su diseño es fácilmente modificable.
- Existe paralelismo entre su diseño y la forma en que l@s usuari@s visualizan y manipulan los datos.
- Simplifica el análisis.
- Facilita la interacción con herramientas de consulta y análisis.

3.4.5.2. Esquema Copo de Nieve

Este esquema representa una extensión del modelo en estrella cuando las tablas de dimensiones se organizan en jerarquías de dimensiones.

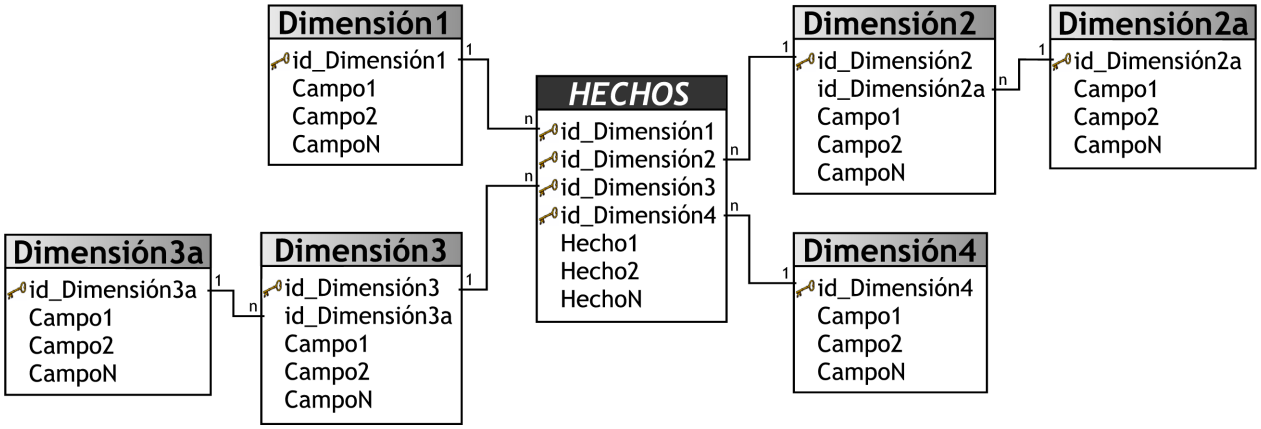


Figura 3.18: Esquema Copo de Nieve.

Como se puede apreciar en la figura anterior, existe una tabla de hechos central que está relacionada con una o más tablas de dimensiones, quienes a su vez pueden estar relacionadas o no con una o más tablas de dimensiones.

Este modelo es más cercano a un modelo de entidad relación, que al modelo en estrella, debido a que sus tablas de dimensiones están normalizadas.

Una de los motivos principales de utilizar este tipo de modelo, es la posibilidad de segregar los datos de las tablas de dimensiones y proveer un esquema que sustente los requerimientos de diseño. Otra razón es que es muy flexible y puede implementarse después de que se haya desarrollado un esquema en estrella.

Se pueden definir las siguientes características de este tipo de modelo:

- Posee mayor complejidad en su estructura.

- Hace una mejor utilización del espacio.

- Es muy útil en tablas de dimensiones de muchas tuplas.

- Las tablas de dimensiones están normalizadas, por lo que requiere menos esfuerzo de diseño.

- Puede desarrollar clases de jerarquías fuera de las tablas de dimensiones, que permiten realizar análisis de lo general a lo detallado y viceversa.

A pesar de todas las características y ventajas que trae aparejada la implementación del esquema copo de nieve, existen dos grandes inconvenientes de ello:

- Si se poseen múltiples tablas de dimensiones, cada una de ellas con varias jerarquías, se creará un número de tablas bastante considerable, que pueden llegar al punto de ser inmanejables.

- Al existir muchas uniones y relaciones entre tablas, el desempeño puede verse reducido.

  Las existencia de las diferentes jerarquías de dimensiones debe estar bien fundamentada, ya que de otro modo las consultas demorarán más tiempo en devolver los resultados, debido a que se deben realizar las uniones entre las tablas.

3.4.5.3. Esquema Constelación

Este modelo está compuesto por una serie de esquemas en estrella, y tal como se puede apreciar en la siguiente figura, está formado por una tabla de hechos principal (“HECHOS_A”) y por una o más tablas de hechos auxiliares (“HECHOS_B”), las cuales pueden ser sumarizaciones de la principal. Dichas tablas yacen en el centro del modelo y están relacionadas con sus respectivas tablas de dimensiones.

No es necesario que las diferentes tablas de hechos compartan las mismas tablas de dimensiones, ya que, las tablas de hechos auxiliares pueden vincularse con solo algunas de las tablas de dimensiones asignadas a la tabla de hechos principal, y también pueden hacerlo con nuevas tablas de dimensiones.

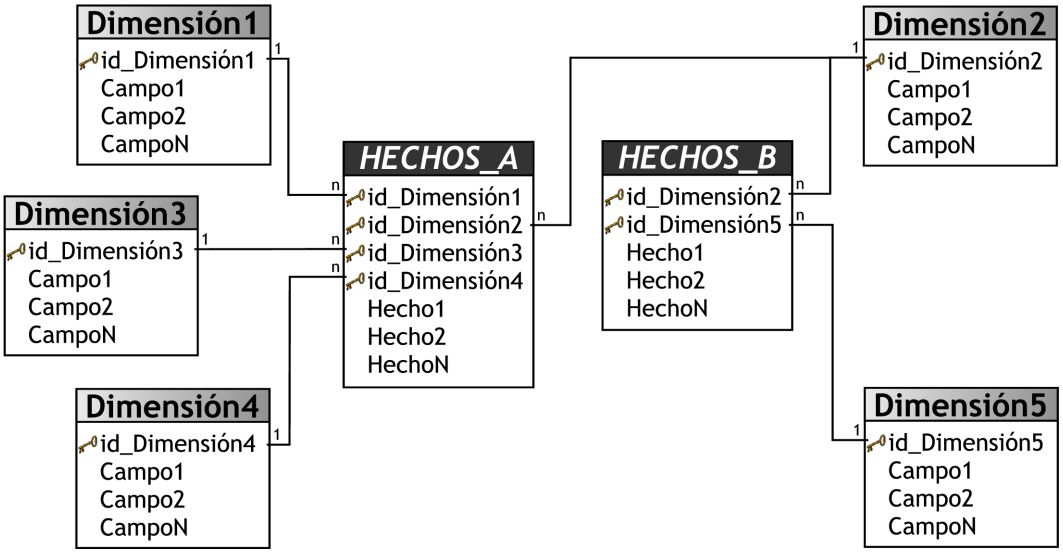


Figura 3.19: Esquema Constelación.

Su diseño y cualidades son muy similares a las del esquema en estrella, pero posee una serie de diferencias con el mismo, que son precisamente las que lo destacan y caracterizan. Entre ellas se pueden mencionar:

- Permite tener más de una tabla de hechos, por lo cual se podrán analizar más aspectos claves del negocio con un mínimo esfuerzo adicional de diseño.
- Contribuye a la reutilización de las tablas de dimensiones, ya que una misma tabla de dimensión puede utilizarse para varias tablas de hechos.
- No es soportado por todas las herramientas de consulta y análisis.

3.4.6. OLTP vs DW

Debido a que, ya se han explicado y caracterizado los distintos tipos de esquemas del DW, se procederá a exponer las razones de su utilización, como así también las causas de por qué no se emplean simplemente las estructuras de las bases de datos tradicionales:

- Los OLTP son diseñados para soportar el procesamiento de información diaria de las empresas, y el énfasis recae en maximizar la capacidad transaccional de sus datos. Su estructura es altamente normalizada, para brindar mayor eficiencia a sistemas con muchas transacciones que acceden a un pequeño número de registros y están fuertemente condicionadas por los procesos operacionales que deben soportar, para la óptima actualización de sus datos. Esta estructura es ideal para llevar a cabo el proceso transaccional diario, brindar consultas sobre los datos cargados y tomar decisiones diarias, en cambio los esquemas de DW están diseñados para poder llevar a cabo procesos de consulta y análisis para luego tomar decisiones estratégicas y tácticas de alto nivel.

A continuación se presentará una tabla comparativa entre los dos ambientes, que resume sus principales diferencias:

	OLTP	Data Warehouse
Objetivo	Soportar actividades transaccionales diarias.	Consultar y analizar información estratégica y táctica.
Tipo de datos	Operacionales.	Para la toma de decisiones.
Modelo de datos	Normalizado.	Desnormalizado.
Consulta	SQL.	SQL más extensiones.
Datos consultados	Actuales.	Actuales e históricos.
Horizonte de tiempo	60 - 90 días.	5 - 10 años.
Tipos de consultas	Repetitivas, predefinidas	No previsibles, dinámicas
Nivel de almacenamiento	Nivel de detalle.	Nivel de detalle y diferentes niveles de sumarización.
Acciones disponibles	Alta, baja, modificación y consulta.	Carga y consulta.
Número de transacciones	Elevado	Medio o bajo
Tamaño	Pequeño - Mediano.	Grande.
Tiempo de respuesta	Pequeño (segundos - minutos).	Variable (minutos - horas).
Orientación	Orientado a las aplicaciones.	Orientado al negocio.
Sello de tiempo	La clave puede o no tener un elemento de tiempo.	La clave tiene un elemento de tiempo.
Estructura	Generalmente estable.	Generalmente varía de acuerdo a su propia evolución y utilización.

Figura 3.20: OLTP vs Data Warehouse.

3.4.7. Tipos de implementación de un DW

3.4.7.1. ROLAP

Este tipo de organización física se implementa sobre tecnología relacional, pero disponen de algunas facilidades para mejorar el rendimiento.

Es decir, ROLAP (Relational On Line Analytic Processing) cuenta con todos los beneficios de una SGBD Relacional a los cuales se les provee extensiones y herramientas para poder utilizarlo como un Sistema Gestor de DW.

En los sistemas ROLAP, los cubos multidimensionales se generan dinámicamente al instante de realizar las diferentes consultas, haciendo de esta manera el manejo de cubos transparente l@s usuari@s. Este proceso se puede resumir a través de los siguientes pasos:

1. Se seleccionan los indicadores, atributos, jerarquías, etc, que compondrán el cubo multidimensional.
2. Se ejecutan las consultas sobre los atributos, indicadores, etc, seleccionados en el paso anterior. Entonces, de manera transparente a l@s usuari@s se crea y calcula dinámicamente el cubo correspondiente, el cual dará respuesta a las consultas que se ejecuten.

Al no tener que intervenir l@s usuari@s en la creación y el mantenimiento explícito de los cubos, ROLAP brinda mucha flexibilidad, ya que dichos cubos son generados dinámicamente al momento de ejecutar las consultas. Posibilitando de esta manera la obtención de consultas ad hoc.

La principal desventaja de los sistemas ROLAP, es que los datos de los cubos se deben calcular cada vez que se ejecuta una consulta sobre ellos. Esto provoca que ROLAP no

sea muy eficiente en cuanto a la rapidez de respuesta ante las consultas de l@s usuari@s.

> Para incrementar la velocidad de respuesta, en algunos casos se puede optar por almacenar los resultados obtenidos de ciertas consultas en la memoria caché (ya sea en el servidor o en una terminal), para que en un futuro, cuando se desee volver a ejecutar dicha consulta, los valores sean obtenidos más velozmente.

Cabe aclarar que si los datos del DW son almacenados y gestionados a través de un SGBD Relacional, no se requiere de otro software que administre y gestione los datos de manera Multidimensional.

Entre las características más importantes de ROLAP, se encuentran las siguientes:

- Almacena la información en una base de datos relacional.
- Utiliza índices de mapas de bits.
- Utiliza índices de Join.
- Posee optimizadores de consultas.
- Cuenta con extensiones de SQL (drill-up, drill-down, etc).

Como se aclaró anteriormente, el almacén de datos se organiza a través de una base de datos multidimensional, sin embargo, puede ser soportado por un SGBD Relacional. Para lograr esto se utilizan los diferentes esquemas, en estrella, copo de nieve y constelación, los cuales transformarán el modelo multidimensional y permitirán que pueda ser gestionado por un SGDB Relacional, ya que solo se almacenarán tablas.

3.4.7.2. MOLAP

El objetivo de los sistemas MOLAP (Multidimentional On Line Analytic Processing) es almacenar físicamente los datos en estructuras multidimensionales de manera que la representación externa y la interna coincidan.

Para ello, se dispone de estructuras de almacenamiento específicas (Arrays) y técnicas de compactación de datos que favorecen el rendimiento del DW.

MOLAP requiere que en una instancia previa se generen y calculen los cubos multidimensionales, para que luego puedan ser consultados. Este proceso se puede resumir a través de los siguientes pasos:

1. Se seleccionan los indicadores, atributos, jerarquías, etc., que compondrán el cubo multidimensional.
2. Se precalculan los datos del cubo.
3. Se ejecutan las consultas sobre los datos precalculados del cubo.

El principal motivo de precalcular los datos de los cubos, es que posibilita que las consultas sean respondidas con mucha rapidez, ya que los mismos no deben ser calculados en tiempo de ejecución, obteniendo de esta manera una muy buena performance.

Existen una serie de desventajas que están directamente relacionadas con la ventaja de precalcular los datos de los cubos multidimensionales, ellas son:

- Cada vez que se requiere o es necesario realizar cambios sobre algún cubo, se debe tener que recalcularlo totalmente, para que se reflejen las modificaciones llevadas a cabo. Provocando de esta manera una disminución importante en cuanto a flexibilidad.

- Se precisa más espacio físico para almacenar dichos datos (esta desventaja no es tan significativa).

Habitualmente, los datos del DW son almacenados y gestionados a través de SGBD Relacionales, ya que estos tienen la ventaja de poder realizar consultas directamente a través del lenguaje SQL. En estos casos, para la generación de los cubos multidimensionales se requiere de otro software que administre y gestione los datos de manera Multidimensional.

3.4.7.3. HOLAP

HOLAP (Hybrid On Line Analytic Processing) constituye un sistema híbrido entre MOLAP y ROLAP, que combina estas dos implementaciones para almacenar algunos datos en un motor relacional y otros en una base de datos multidimensional.

Los datos agregados y precalculados se almacenan en estructuras multidimensionales y los de menor nivel de detalle en estructuras relacionales. Es decir, se utilizará ROLAP para navegar y explorar los datos, y se empleará MOLAP para la realización de tableros.

Como contrapartida, hay que realizar un buen análisis para identificar los diferentes tipos de datos.

3.4.7.4. ROLAP vs MOLAP

En la siguiente tabla comparativa se pueden apreciar las principales diferencias entre estos dos tipos de implementación:

ROLAP	MOLAP
Brinda mucha flexibilidad, ya que los cubos son generados dinámicamente al momento de ejecutar las consultas.	Cada vez que se requiere o es necesario realizar cambios sobre algún cubo, se debe tener que recalcularlo totalmente, para que se reflejen las modificaciones llevadas a cabo. Provocando de esta manera una disminución importante en cuanto a flexibilidad.
Los datos de los cubos se deben calcular cada vez que se ejecuta una consulta sobre ellos. Esto provoca que ROLAP no sea muy eficiente en cuanto a la rapidez de respuesta ante las consultas de l@s usuari@s.	Las consultas son respondidas con mucha rapidez, ya que los mismos no deben ser calculados en tiempo de ejecución, obteniendo de esta manera una muy buena performance.

Figura 3.21: ROLAP vs MOLAP.

3.4.8. Cubo Multidimensional: profundización

Ahora que ya se tiene una visión general de los tipos de modelamiento e implementación de un DW, se volverá a abordar el tema de los cubos multidimensionales, pero esta vez se hará énfasis en su construcción y se ejemplificará cada paso, a fin de que se puedan visualizar mejor los resultados de cada acción.

La forma que se utilizará para graficar el cubo que se creará, será la siguiente:

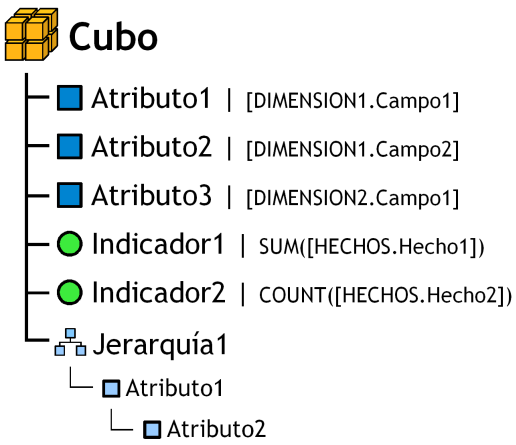


Figura 3.22: Cubo estándar.

Tal y como podemos observar, el gráfico toma una estructura de árbol, en la cuál en la raíz figura el cubo en cuestión y dependiendo de este sus diferentes objetos relacionados. En el caso de las jerarquías, los atributos que la componen, también deben

estructurarse en forma de árbol, teniendo en cuenta su respectiva relación padre-hijo.

Se tomará como base para la realización de los ejemplos, el siguiente esquema en estrella:

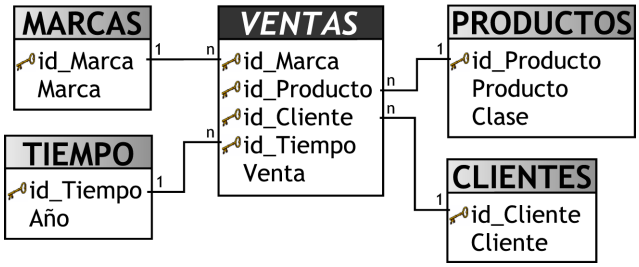


Figura 3.23: Esquema en Estrella.

Como primer paso se creará un cubo multidimensional llamado ”Cubo de Ventas”, gráficamente:

Figura 3.24: Cubo multidimensional, paso 1.

Luego se crearán dos atributos:

- De la tabla de dimensión “PRODUCTOS”, se tomará el campo “Producto” para la creación del atributo denominado:
  - “PRODUCTOS - Producto”.
- De la tabla dimensión “MARCAS”, se tomará el campo “Marca” para la creación del atributo denominado:
  - “MARCAS - Marca”.

Gráficamente:

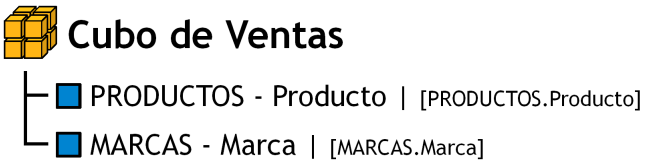


Figura 3.25: Cubo multidimensional, paso 2.

También se creará un indicador:

- De la tabla de hechos “VENTAS”, se sumarizará el hecho “Venta” para crear el indicador denominado:
  - “VENTAS - Venta”.

  La fórmula utilizada para crear este indicador es la siguiente:
  - “VENTAS - Venta” = SUM(VENTAS.Venta).

Gráficamente:

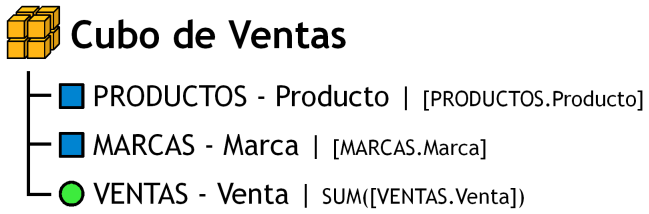


Figura 3.26: Cubo multidimensional, paso 3.

En este momento, tenemos un cubo multidimensional de dos dimensiones, cuya representación matricial sería la siguiente:

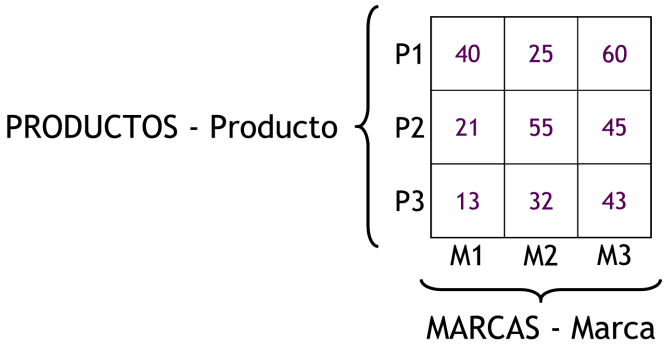


Figura 3.27: Cubo multidimensional de dos dimensiones.

Este cubo posee dos ejes o dimensiones, “PRODUCTOS - Producto” y “MARCAS - Marca”. La intersección de los ejes representa las ventas de cada producto con su respectiva marca (indicador “VENTAS - Venta”).

Por ejemplo:

- Las ventas asociadas al producto “P1” y a la marca “M1” son 40.
- Las ventas asociadas al producto “P1” y a la marca “M2” son 25.
- Las ventas asociadas al producto “P1” y a la marca “M3” son 60.

Ahora, al cubo planteado se le agregará un nuevo atributo:

- De la tabla de dimensión “CLIENTES”, se tomará el campo “Cliente” para la creación del atributo denominado:
    - “CLIENTES - Cliente”.

Gráficamente:

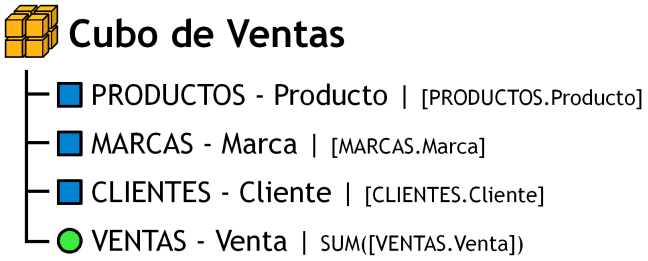


Figura 3.28: Cubo multidimensional, paso 4.

De esta manera, ahora tenemos un cubo multidimensional de tres dimensiones, cuya representación matricial sería la siguiente:

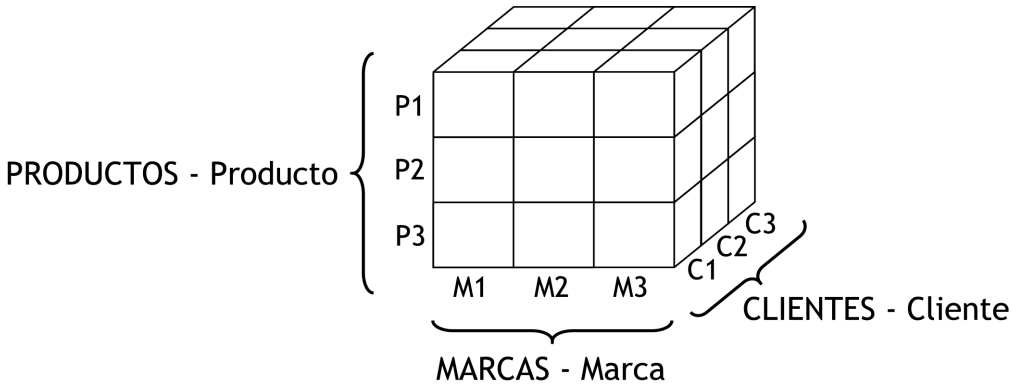


Figura 3.29: Cubo multidimensional de tres dimensiones.

En este caso los valores del indicador “VENTAS - Venta” están dados de acuerdo a las ventas de cada producto, de cada marca, a cada cliente.

Para finalizar, se añadirá un cuarto atributo al cubo:

- De la tabla de dimensión “TIEMPO”, se tomará el campo “Año” para la creación del atributo denominado:
    - “TIEMPO - Año”.

Gráficamente:

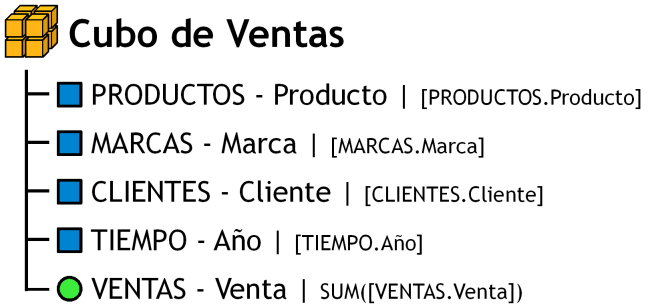


Figura 3.30: Cubo multidimensional, paso 5.

Entonces, la representación matricial del cubo multidimensional resultante sería la siguiente:

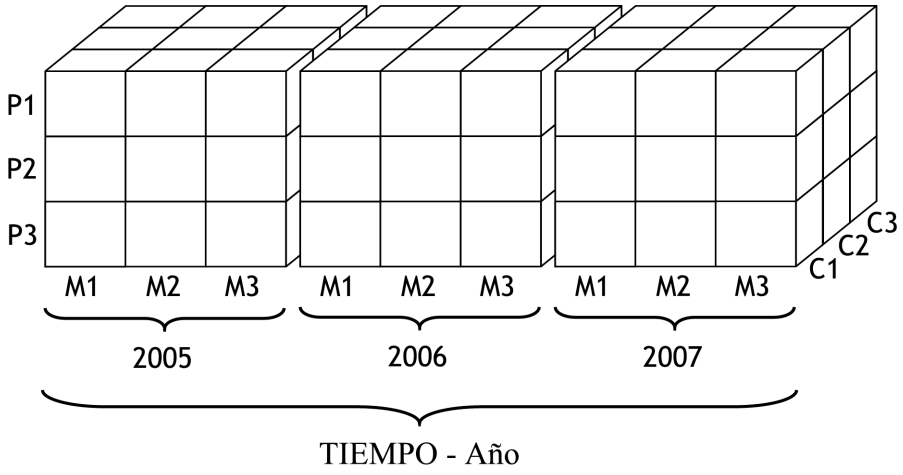


Figura 3.31: Cubo multidimensional de cuatro dimensiones.

Los valores del indicador “VENTAS - Venta” en este momento, estarán condicionados por las ventas de cada producto, de cada marca, a cada cliente, en cada año.

Esta última imagen, demuestra claramente los conceptos expuestos de la tabla de dimensión tiempo, donde se decía que pueden existir diferentes versiones de la situación del negocio.

Cabe aclarar que pueden crearse tantos cubos se deseen y que los mismos pueden coexistir sin ningún inconveniente.

3.4.9. Metadatos

Los metadatos son datos que describen o dan información de otros datos, que en este caso, existen en la arquitectura del Data Warehousing. Brindan información de localiza-

ción, estructura y significado de los datos, básicamente mapean los mismos.

El concepto de metadatos es análogo al uso de índices para localizar objetos en lugar de datos.

Es importante aclarar que existen metadatos también en las bases de datos transaccionales, pero los mismos son transparentes a l@s usuari@s. La gran ventaja que trae aparejada el Data Warehousing en relación con los metadatos es que l@s usuari@s pueden gestionarlos, exportarlos, importarlos, realizarles mantenimiento e interactuar con ellos, ya sea manual o automáticamente.

Las funciones que cumplen los metadatos en el ambiente Data Warehousing son muy importantes y significativas, algunas de ellas son:

- Facilitan el flujo de trabajo, convirtiendo datos automáticamente de un formato a otro.
- Contienen un directorio para facilitar la búsqueda y descripción de los contenidos del DW, tales como: bases de datos, tablas, nombres de atributos, sumarizaciones, acumulaciones, reglas de negocios, estructuras y modelos de datos, relaciones de integridad, jerarquías, etc.
- Poseen un guía para el mapping[26], de cómo se transforman e integran los datos de las fuentes operacionales y externos al ambiente del depósito de datos.
- Almacenan las referencias de los algoritmos utilizados para la esquematización entre el detalle de datos actuales, con los datos ligeramente resumidos y éstos con los datos altamente resumidos, etc.
- Contienen las definiciones del sistema de registro desde el cual se construye el DW.

Se pueden distinguir tres diferentes tipos de Metadatos:

- Los metadatos de los procesos ETL, referidos a las diversas fuentes utilizadas, reglas de extracción, transformación, limpieza, depuración y carga de los datos al depósito.
- Los metadatos operacionales, que son los que básicamente almacenan todos los contenidos del DW, para que este pueda desempeñar sus tareas.
- Los metadatos de consulta, que contienen las reglas para analizar y explotar la información del almacén, tales como drill-up y drill-down. Son estos metadatos los que las herramientas de análisis y consulta emplearán para realizar documentaciones y para navegar por los datos.

3.4.9.1. Mapping

El término mapping, se refiere a relacionar un conjunto de objetos, tal como actualmente están almacenados en memoria o en disco, con otros objetos. Por ejemplo: una estructura de base de datos lógica, se proyecta sobre la base de datos física.

[26] Ver sección 3.4.9.1, en la página 50.

3.5. Query Manager

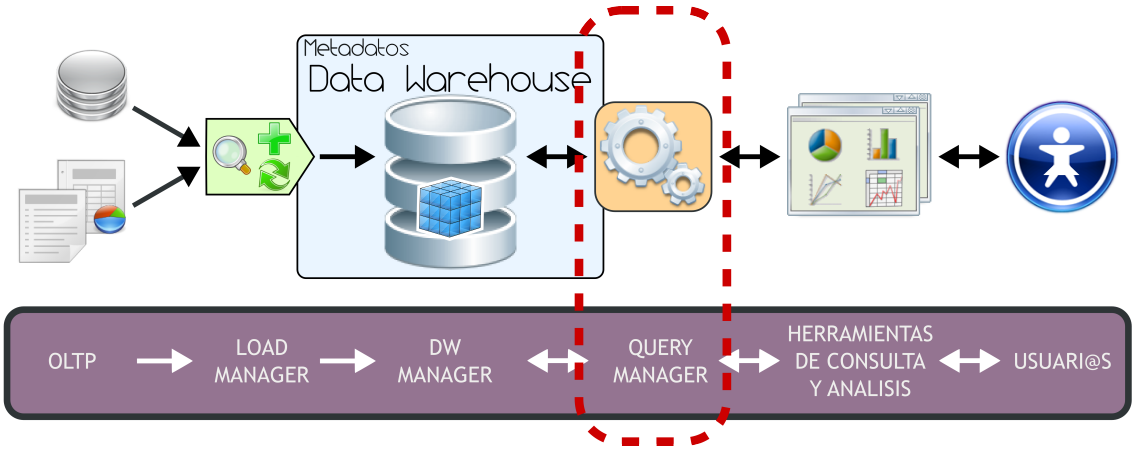


Figura 3.32: Query Manager.

Este componente realiza las operaciones necesarias para soportar los procesos de gestión y ejecución de consultas relacionales, tales como Join y agregaciones, y de consultas propias del análisis de datos, como drill-up y drill-down.

Query Manager recibe las consultas de l@s usuari@s, las aplica a la estructura de datos correspondiente (cubo multidimensional, Business Models, etc.) y devuelve los resultados obtenidos.

Cabe aclarar que una consulta a un DW, generalmente consiste en la obtención de indicadores a partir de los datos (hechos) de una tabla de hechos, restringidas por las propiedades o condiciones de los atributos que hayan sido creados.

Las operaciones que se pueden realizar sobre modelos multidimensionales y que son las que verdaderamente les permitirán a l@s usuari@s explorar e investigar los datos en busca de respuestas, son:

- Drill-down.
- Drill-up.
- Drill-across.
- Roll-across.
- Pivot.
- Page.
- Drill-through.

A continuación, se explicará cada una de ellas y se ejemplificará su utilización, para lo cual se utilizará como guía el siguiente esquema en estrella.

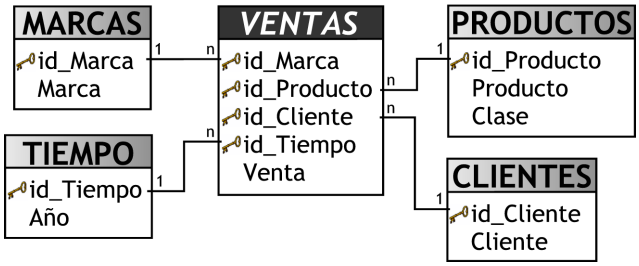


Figura 3.33: Esquema en Estrella.

El mismo posee cuatro tablas de dimensiones y una tabla de hechos central, en la cual el hecho “Venta” representa las ventas a un cliente, de un producto en particular, de una marca específica en un año dado.

Sobre este modelo, entonces, se creará un cubo llamado “Cubo - Query Manager” que será utilizado para explicar las operaciones del Query Manager. El mismo contiene los siguientes objetos:

- De la tabla de hechos “VENTAS”, se sumarizará el hecho “Venta” para crear el indicador denominado:
  - “VENTAS - Venta”.

  La fórmula utilizada para crear este indicador es la siguiente:
  - “VENTAS - Venta” = SUM(VENTAS.Venta).
- De la tabla de dimensión “MARCAS”, se tomará el campo “Marca” para la creación del atributo denominado:
  - “MARCAS - Marca”.
- De la tabla dimensión “TIEMPO”, se tomará el campo “Año” para la creación del atributo denominado:
  - “TIEMPO - Año”.
- De la tabla dimensión “PRODUCTOS”, se tomará el campo “Producto” para la creación del atributo denominado:
  - “PRODUCTOS - Producto”.
- De la tabla dimensión “PRODUCTOS”, se tomará el campo “Clase” para la creación del atributo denominado:
  - “PRODUCTOS - Clase”.
- Se definió la jerarquía “Jerarquía PRODUCTOS”, que se aplicará sobre los atributos recientemente creados, “PRODUCTOS - Producto” y “PRODUCTOS - Clase”, en donde:
  - Una clase de producto pertenece solo a un producto. Un producto puede ser de una o más clases.

Gráficamente:

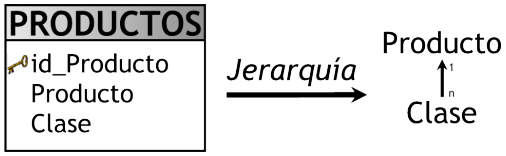


Figura 3.34: ”PRODUCTOS”, relación padre-hijo.

Entonces, el cubo quedará conformado de la siguiente manera:

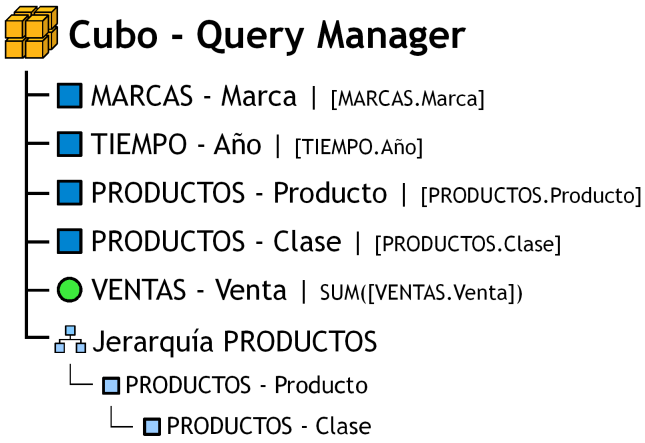


Figura 3.35: ”Cubo - Query Manager”.

Para simplificar los ejemplos que se presentarán, se utilizará solo una pequeña muestra de datos correspondientes al año 2007.

3.5.1. Drill-down

Permite apreciar los datos en un mayor detalle, bajando por una jerarquía definida en un cubo. Esto brinda la posibilidad de introducir un nuevo nivel o criterio de agregación en el análisis, disgregando los grupos actuales.

Drill-down es ir de lo general a lo específico. Gráficamente:

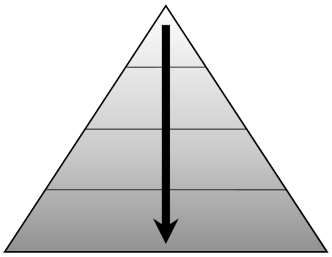


Figura 3.36: Drill-down.

Para explicar esta operación se utilizará la siguiente representación tabular:

TIEMPO - Año	PRODUCTOS - Producto	MARCAS - Marca	VENTAS - Venta
2007	Producto1	M1	40
2007	Producto1	M2	52
2007	Producto1	M3	25
2007	Producto2	M1	39
2007	Producto2	M2	65
2007	Producto2	M3	48

Figura 3.37: Resultados antes de aplicar Drill-down.

Como puede apreciarse, en la cabecera de la tabla se encuentran los atributos y el indicador (destacado con color de fondo diferente) definidos anteriormente en el cubo multidimensional; y en el cuerpo de la misma se encuentran los valores correspondientes. Se ha resaltado la primera fila, ya que es la que se analizará más en detalle.

En este caso, se realizará drill-down sobre la jerarquía “Jerarquía PRODUCTOS”, entonces:

TIEMPO - Año	PRODUCTOS - Producto	PRODUCTOS - Clase	MARCAS - Marca	VENTAS - Venta
2007	Producto1	A1	M1	22
2007	Producto1	B1	M1	18
2007	Producto1	A1	M2	33
2007	Producto1	B1	M2	19
2007	Producto1	A1	M3	15
2007	Producto1	B1	M3	10
2007	Producto2	A2	M1	21
2007	Producto2	B2	M1	18
2007	Producto2	A2	M2	30
2007	Producto2	B2	M2	35
2007	Producto2	A2	M3	26
2007	Producto2	B2	M3	22

Figura 3.38: Resultados después de aplicar Drill-down.

Tal y como puede apreciarse en los ítems resaltados de la tabla, se agregó un nuevo nivel de detalle (“PRODUCTOS – Clase”) a la lista inicial, y el valor “40” que pertenecía a las ventas del “Producto1”, de la marca “M1”, en el año “2007”, se dividió en dos filas. Esto se debe a que ahora se tendrá en cuenta el atributo “PRODUCTOS - Clase” para realizar las sumarizaciones del indicador “VENTAS - Venta”.

La siguiente imagen muestra este mismo proceso pero, representado matricialmente:

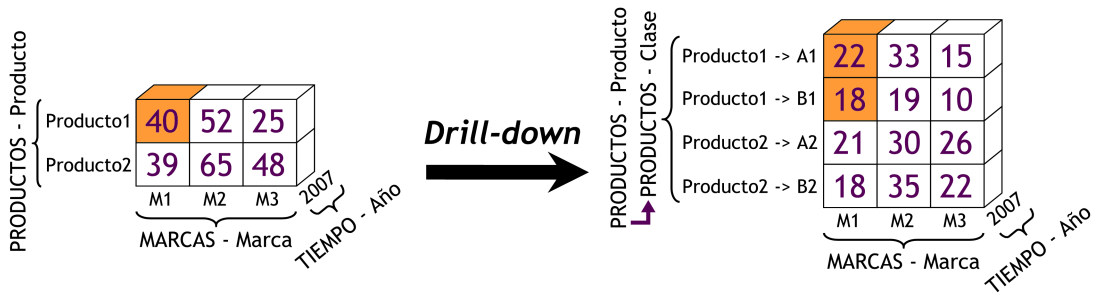


Figura 3.39: Drill-down, representación matricial.

De aquí en más se utilizará esta forma para explicar cada operación.

3.5.2. Drill-up

Permite apreciar los datos en menor nivel de detalle, subiendo por una jerarquía definida en un cubo. Esto brinda la posibilidad de quitar un nivel o criterio de agregación en el análisis, agregando los grupos actuales.

Drill-up es ir de lo específico a lo general. Gráficamente:

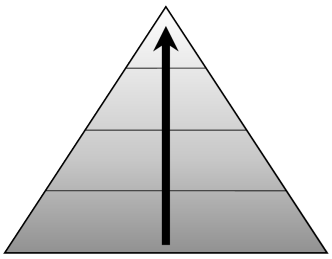


Figura 3.40: Drill-up.

Se tomará como referencia los resultados anteriores:

TIEMPO - Año	PRODUCTOS - Producto	PRODUCTOS - Clase	MARCAS - Marca	VENTAS - Venta
2007	Producto1	A1	M1	22
2007	Producto1	B1	M1	18
2007	Producto1	A1	M2	33
2007	Producto1	B1	M2	19
2007	Producto1	A1	M3	15
2007	Producto1	B1	M3	10
2007	Producto2	A2	M1	21
2007	Producto2	B2	M1	18
2007	Producto2	A2	M2	30
2007	Producto2	B2	M2	35
2007	Producto2	A2	M3	26
2007	Producto2	B2	M3	22

Figura 3.41: Resultados antes de aplicar Drill-up.

Se aplicará drill-up sobre la jerarquía “Jerarquía PRODUCTOS”, entonces:

TIEMPO - Año	PRODUCTOS - Producto	MARCAS - Marca	VENTAS - Venta
2007	Producto1	M1	40
2007	Producto1	M2	52
2007	Producto1	M3	25
2007	Producto2	M1	39
2007	Producto2	M2	65
2007	Producto2	M3	48

Figura 3.42: Resultados después de aplicar Drill-up.

Como se puede observar en la lista resultante, en la fila resaltada se sumarizaron los valores “22” y “18” de la tabla inicial, debido a que al eliminar el atributo “PRODUCTOS - Clase”, las ventas se agruparon o sumarizaron de acuerdo a “PRODUCTOS - Producto”, “MARCAS - Marca” y “TIEMPO - Año”.

La siguiente imagen muestra este mismo proceso pero, representado matricialmente:

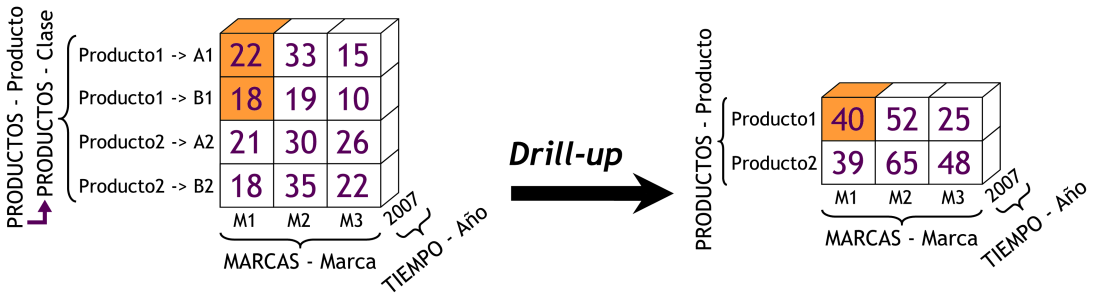


Figura 3.43: Drill-up, representación matricial.

3.5.3. Drill-across

Funciona de forma similar a drill-down, con la diferencia de que drill-across no se realiza sobre una jerarquía, sino que su forma de ir de lo general a lo específico es agregar un atributo a la consulta como nuevo criterio de análisis.

Se partirá de los siguientes resultados:

TIEMPO - Año	PRODUCTOS - Producto	VENTAS - Venta
2007	Producto1	117
2007	Producto2	152

Figura 3.44: Resultados antes de aplicar Drill-across.

Ahora, se aplicará drill-across, al agregar el atributo “MARCAS - Marca”, entonces:

TIEMPO - Año	PRODUCTOS - Producto	MARCAS - Marca	VENTAS - Venta
2007	Producto1	M1	40
2007	Producto1	M2	52
2007	Producto1	M3	25
2007	Producto2	M1	39
2007	Producto2	M2	65
2007	Producto2	M3	48

Figura 3.45: Resultados después de aplicar Drill-across.

La siguiente imagen muestra este mismo proceso pero, representado matricialmente:

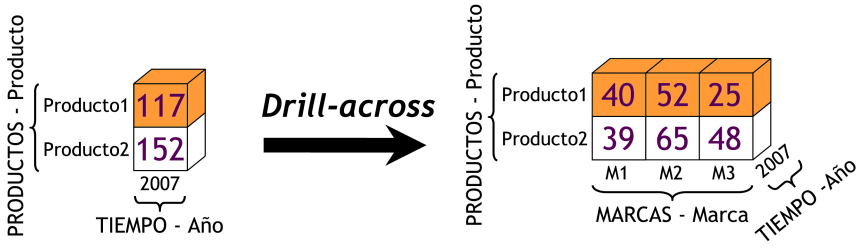


Figura 3.46: Drill-across, representación matricial.

3.5.4. Roll-across

Funciona de forma similar a drill-up, con la diferencia de que roll-across no se hace sobre una jerarquía, sino que su forma de ir de lo específico a lo general es quitar un atributo de la consulta, eliminando de esta manera un criterio de análisis.

Se tomará como base la representación tabular anterior:

TIEMPO - Año	PRODUCTOS - Producto	MARCAS - Marca	VENTAS - Venta
2007	Producto1	M1	40
2007	Producto1	M2	52
2007	Producto1	M3	25
2007	Producto2	M1	39
2007	Producto2	M2	65
2007	Producto2	M3	48

Figura 3.47: Resultados antes de aplicar Roll-across.

Se aplicará la operación roll-across, quitando de la consulta el atributo “MARCAS - Marca”, entonces:

TIEMPO - Año	PRODUCTOS - Producto	VENTAS - Venta
2007	Producto1	117
2007	Producto2	152

Figura 3.48: Resultados después de aplicar Roll-across.

La siguiente imagen muestra este mismo proceso pero, representado matricialmente:

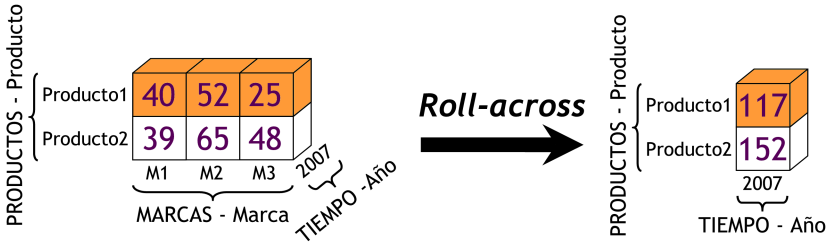


Figura 3.49: Roll-across, representación matricial.

3.5.5. Pivot

Permite seleccionar el orden de visualización de los atributos e indicadores, con el objetivo de analizar la información desde diferentes perspectivas.

Se tomará como referencia, para explicar esta operación, la siguiente tabla:

TIEMPO - Año	PRODUCTOS - Producto	MARCAS - Marca	VENTAS - Venta
2007	Producto1	M1	40
2007	Producto1	M2	52
2007	Producto1	M3	25
2007	Producto2	M1	39
2007	Producto2	M2	65
2007	Producto2	M3	48

Figura 3.50: Resultados antes de aplicar Pivot.

Como puede apreciarse, el orden de los atributos es: “TIEMPO - Año”, “PRODUCTOS - Producto” y “MARCAS - Marca”. Ahora, se hará pivot, reorientando la vista multidimensional:

MARCAS - Marca	TIEMPO - Año	PRODUCTOS - Producto	VENTAS - Venta
M1	2007	Producto1	40
M1	2007	Producto2	39
M2	2007	Producto1	52
M2	2007	Producto2	65
M3	2007	Producto1	25
M3	2007	Producto2	48

Figura 3.51: Resultados después de aplicar Pivot.

El nuevo orden de los atributos es: “MARCAS - Marca”, “TIEMPO - Año” y “PRODUCTOS - Producto”.

La siguiente imagen muestra este mismo proceso pero, representado matricialmente:

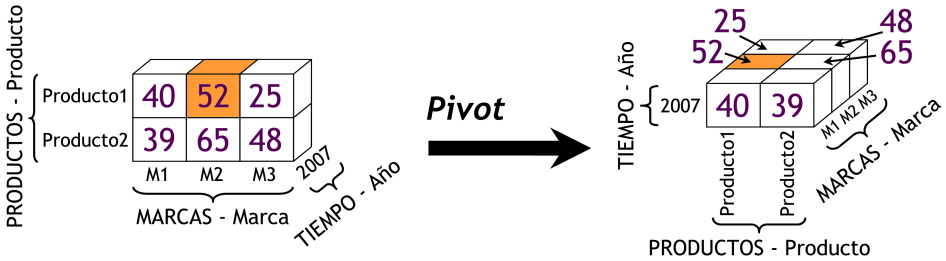


Figura 3.52: Pivot, representación matricial.

Pivot permite realizar las siguientes acciones:

- Mover un atributo o indicador desde el encabezado de fila al encabezado de columna.
- Mover un atributo o indicador desde el encabezado de columna al encabezado de fila.
- Cambiar el orden de los atributos o indicadores del encabezado de columna.
- Cambiar el orden de los atributos o indicadores del encabezado de fila.

3.5.6. Page

Presenta el cubo dividido en secciones, a través de los valores de un atributo, como si se tratase de páginas de un libro. Gráficamente:

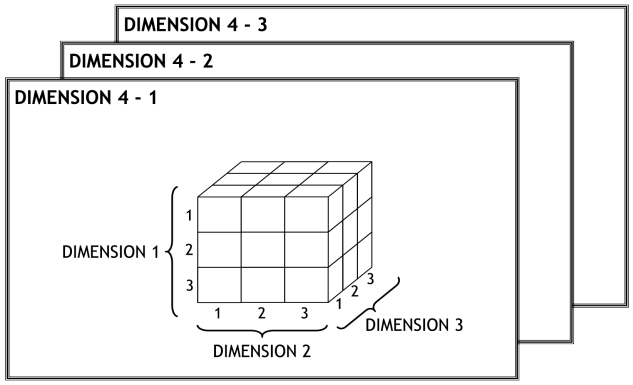


Figura 3.53: Page.

Page es muy útil cuando las consultas devuelven muchos registros y es necesario desplazarse por los datos para poder verlos en su totalidad.

Se tomará como referencia, para explicar esta operación, la siguiente tabla:

TIEMPO - Año	PRODUCTOS - Producto	MARCAS - Marca	VENTAS - Venta
2007	Producto1	M1	40
2007	Producto1	M2	52
2007	Producto1	M3	25
2007	Producto2	M1	39
2007	Producto2	M2	65
2007	Producto2	M3	48

Figura 3.54: Resultados antes de aplicar Page.

Se realizará Page sobre el atributo “PRODUCTOS - Producto”, entonces se obtendrán las siguientes páginas:

- Página Nro 1:

Producto1		
TIEMPO - Año	MARCAS - Marca	VENTAS - Venta
2007	M1	40
2007	M2	52
2007	M3	25

Figura 3.55: Página Nro 1, representación tabular.

Matricialmente:

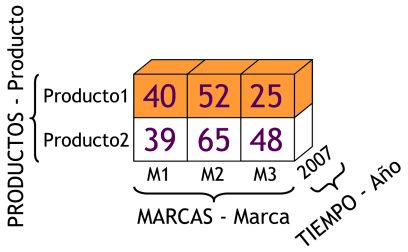


Figura 3.56: Página Nro 1, representación matricial.

- Página Nro 2:

Producto2		
TIEMPO - Año	MARCAS - Marca	VENTAS - Venta
2007	M1	39
2007	M2	65
2007	M3	48

Figura 3.57: Página Nro 2, representación tabular.

Matricialmente:

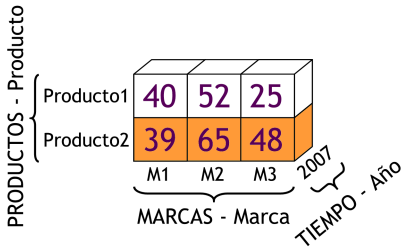


Figura 3.58: Página Nro 2, representación matricial.

Cuando existe más de un criterio por el cual realizar Page, debe tenerse en cuenta el orden en que estos serán procesados, ya que dependiendo de esto, de podrán obtener diferentes resultados sobre una misma consulta. Para ejemplificar este concepto se utilizará como base la tabla expuesta al inicio.

Entonces, si se desea realizar Page por “PRODUCTOS - Producto” y “MARCAS - Marca”, se dispondrá de dos opciones de ordenación:

1. Primero por “PRODUCTOS - Producto” y luego por “MARCAS - Marca”: en este caso, si se selecciona la página correspondiente al producto “Producto1”, se obtendrán las siguientes sub-páginas, que se corresponden con los valores de las marcas: “M1”, “M2” y “M3”. Expresado en esquema de árbol jerárquico, quedaría como sigue:

   - Página Nro 1: “Producto1”
     - Sub-página Nro 1.1: “M1”.
     - Sub-página Nro 1.2: “M2”.
     - Sub-página Nro 1.3: “M3”.
   - Página Nro 2: “Producto2”
     - Sub-página Nro 2.1: “M1”.
     - Sub-página Nro 2.2: “M2”.
     - Sub-página Nro 2.3: “M3”.

   Como puede observarse, se obtendrán dos páginas, con tres sub-páginas cada una de ellas.

2. Primero por “MARCAS - Marca” y luego por “PRODUCTOS - Producto”: en este caso, si se selecciona la página correspondiente a la marca “M1”, se obtendrán las siguientes sub-páginas, que se corresponden con los valores de los productos: “Producto1” y “Producto2”. Expresado en esquema de árbol jerárquico, quedaría como sigue:

   - Página Nro 1: “M1”
     - Sub-página Nro 1.1: “Producto1”.
     - Sub-página Nro 1.2: “Producto2”.
   - Página Nro 2: “M2”
     - Sub-página Nro 2.1: “Producto1”.
     - Sub-página Nro 2.2: “Producto2”.
   - Página Nro 3: “M3”
     - Sub-página Nro 3.1: “Producto1”.
     - Sub-página Nro 3.2: “Producto2”.

   Como puede observarse, se obtendrán tres páginas, con dos sub-páginas cada una de ellas.

Es decir, el primer criterio utilizado para realizar Page condiciona los valores disponibles en el segundo, y así sucesivamente.

3.5.7. Drill-through

Permite apreciar los datos en su máximo nivel de detalle. Esto brinda la posibilidad de analizar cuáles son los datos relacionados al valor de un Indicador, que se ha sumarizado dentro del cubo multidimensional.

Se tomará como referencia, para explicar esta operación, la siguiente tabla:

TIEMPO - Año	PRODUCTOS - Producto	VENTAS - Venta
2007	Producto1	117
2007	Producto2	152

Figura 3.59: Resultados antes de aplicar Drill-through.

Se aplicará la operación drill-through sobre el Indicador de la fila seleccionada, para obtener el detalle de este valor:

TIEMPO - Año	PRODUCTOS - Producto	PRODUCTOS - Clase	MARCAS - Marca	VENTAS - Venta
2007	Producto1	A1	M1	22
2007	Producto1	B1	M1	18
2007	Producto1	A1	M2	33
2007	Producto1	B1	M2	19
2007	Producto1	A1	M3	15
2007	Producto1	B1	M3	10

Figura 3.60: Resultados después de aplicar Drill-through.

La siguiente imagen muestra este mismo proceso pero, representado matricialmente:

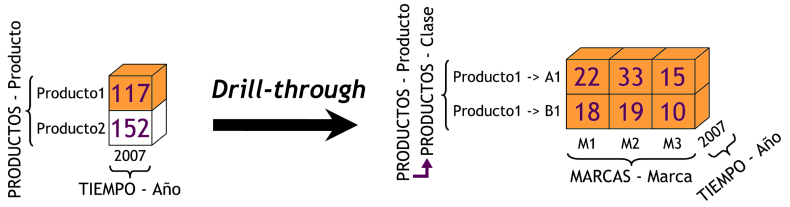


Figura 3.61: Drill-through, representación matricial.

3.6. Herramientas de Consulta y Análisis

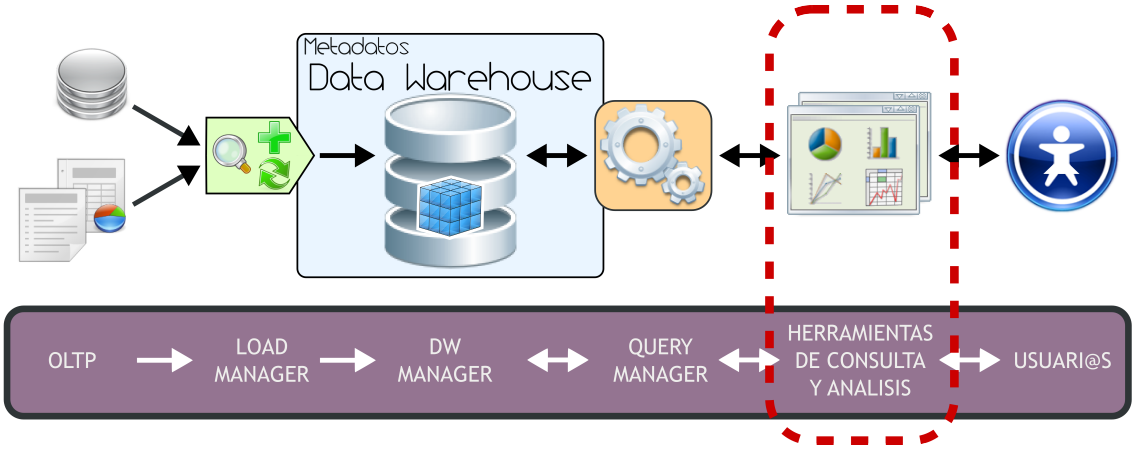


Figura 3.62: Herramientas de Consulta y Análisis.

Las herramientas de consulta y análisis son sistemas que permiten a l@s usuari@s realizar la exploración de datos del DW. Básicamente constituyen el nexo entre el depósito de datos y l@s usuari@s.

Utilizan la metadata de las estructuras de datos que han sido creadas previamente (cubos multidimensionales, Business Models, etc.) para trasladar a través de consultas SQL los requerimientos de l@s usuari@s, para luego, devolver el resultado obtenido.

Estas herramientas también pueden emplear simples conexiones a bases de datos (JNDI, JDBC, ODBC), para obtener la información deseada.

A través de una interfaz gráfica y una serie de pasos, l@s usuari@s generan consultas que son enviadas desde la herramienta de consulta y análisis al Query Manager, este a su vez realiza la extracción de información al DW Manager y devuelve los resultados obtenidos a la herramienta que se los solicitó. Luego, estos resultados son expuestos ante l@s usuari@s en formatos que le son familiares.

Este proceso se puede comprender mejor al observar la siguiente figura:

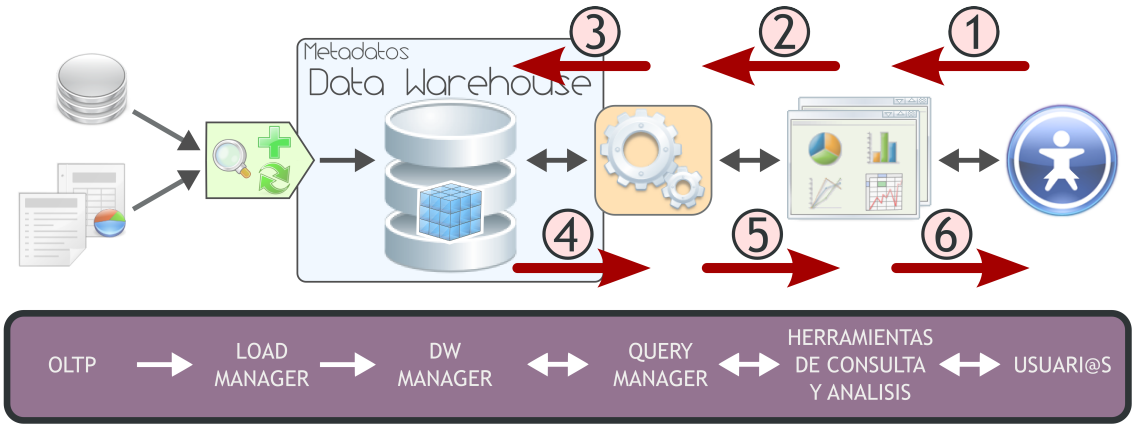


Figura 3.63: Proceso de Consulta y Análisis.

El mismo, se lleva a cabo a través de seis pasos sucesivos:

1. L@s usuari@s seleccionan o establecen que datos desean obtener del DW, mediante las interfaces de la herramienta que utilice.
2. La herramienta recibe el pedido de l@s usuari@s, construye la consulta (utilizando la metadata) y la envía al Query Manager.
3. El Query Manager ejecuta la consulta sobre la estructura de datos con la que se esté trabajando (cubo multidimensional, Business Model, etc.).
4. El Query Manager obtiene los resultados de la consulta.
5. El Query Manager envía los datos a la herramienta de consulta y análisis.
6. La herramienta presentan a l@s usuari@s la información requerida.

Una de las principales ventajas de utilizar estas herramientas, es que l@s usuari@s no se tienen que preocupar por conocer cuáles son las características y funcionalidades de las estructuras de datos utilizadas, ni por saber emplear el lenguaje SQL, solo se deben enfocar en el análisis.

Las herramientas de consulta y análisis, en general, comparten las siguientes características:

- Accesibilidad a la información: permiten el acceso a la información a través de las diferentes estructuras de datos de forma transparente a l@s usuari@s finales, para que est@s solo se enfoquen en el análisis y no en el origen y procedencia de los datos.
- Apoyo en la toma de decisiones: permiten la exploración de los datos, a fin de seleccionar, filtrar y personalizar los mismos, para la obtención de información oportuna, relevante y útil, para apoyar el proceso de toma de decisiones.
- Orientación l@s usuari@s finales: permiten a través de entornos amigables e intuitivos, que l@s usuari@s puedan realizar análisis y consultas, sin poseer conocimientos técnicos. Si bien lo realmente importante son los datos mismos, que estos puedan ser interpretados y analizados por l@s usuari@s dependerá en gran medida de cómo se presenten y dispongan.

Existen diferentes tipos de herramientas de consulta y análisis, y de acuerdo a la necesidad, tipos de usuari@s y requerimientos de información, se deberán seleccionar las más propicias al caso. Entre ellas se destacan las siguientes:

- Reportes y Consultas.
- OLAP.
- Dashboards.
- Data Mining.
- EIS.

3.6.1. Reportes y Consultas

Se han desarrollado muchas herramientas para la producción de consultas y reportes, que ofrecen a l@s usuari@s, a través de pantallas gráficas intuitivas, la posibilidad de generar informes avanzados y detallados del tema de interés de interés que se este analizando. L@s usuari@s solo deben seguir una serie de simples pasos, como por ejemplo seleccionar opciones de un menú, presionar tal o cual botón para especificar los elementos de datos, sus condiciones, criterios de agrupación y demás atributos que se consideren significativos.

Actualmente las herramientas de generación de reportes y consultas cuentan con muchas prestaciones, las cuales permiten dar variadas formas y formatos a la presentación de la información. Entre las opciones más comunes se encuentran las siguientes:

- Parametrización de los datos devueltos.
- Selección de formatos de salida (planilla de cálculo, HTML, PDF, etc.).
- Inclusión de gráficos de tortas, barras, etc.
- Utilización de plantillas de formatos de fondos.
- Inclusión de imágenes.
- Formatos tipográficos.
- Links a otros reportes.

3.6.2. OLAP

El procesamiento analítico en línea OLAP (On Line Analytic Processing), es la componente más poderosa del Data Warehousing, ya que es el motor de consultas especializado del depósito de datos.

Las herramientas OLAP, son una tecnología de software para análisis en línea, administración y ejecución de consultas, que permiten inferir información del comportamiento del negocio.

Su principal objetivo es el de brindar rápidas respuestas a complejas preguntas, para interpretar la situación del negocio y tomar decisiones. Cabe destacar que lo que es realmente interesante en OLAP, no es la ejecución de simples consultas tradicionales, sino la posibilidad de utilizar operadores tales como drill-up, drill-down, etc, para explotar profundamente la información.

Además, a través de este tipo de herramientas, se puede analizar el negocio desde diferentes escenarios históricos, y proyectar como se ha venido comportando y evolucionando en un ambiente multidimensional, o sea, mediante la combinación de diferentes perspectivas, temas de interés o dimensiones. Esto permite deducir tendencias, por medio del descubrimiento de relaciones entre las perspectivas que a simple vista no se podrían encontrar sencillamente.

Las herramientas OLAP requieren que los datos estén organizados dentro del depósito en forma multidimensional, por lo cual se utilizan cubos multidimensionales.

Además de las características ya descritas, se pueden enumerar las siguientes:

- Permite recolectar y organizar la información analítica necesaria para l@s usuari@s y disponer de ella en diversos formatos, tales como tablas, gráficos, reportes, tableros de control, etc.
- Soporta análisis complejos de grandes volúmenes de datos.
- Complementa las actividades de otras herramientas que requieran procesamiento analítico en línea.
- Presenta a l@s usuari@s una visión multidimensional de los datos (matricial) para cada tema de interés del negocio.
- Es transparente al tipo de tecnología que soporta el DW, ya sea ROLAP, MOLAP u HOLAP.
- No tiene limitaciones con respecto al número máximo de dimensiones permitidas.
- Permite a l@s usuari@s, analizar la información basándose en más criterios que un análisis de forma tradicional.
- Al contar con muestras grandes, se pueden explorar mejor los datos en busca de respuestas.
- Permiten realizar agregaciones y combinaciones de los datos de maneras complejas y específicas, con el fin de realizar análisis más estratégicos.

3.6.3. Dashboards

Los Dashboards se pueden entender como una colección de reportes, consultas y análisis interactivos que hacen referencia a un tema en particular y que están relacionados entre sí.

Existen diversas maneras de diseñar un Dashboard, cada una de las cuales tiene sus objetivos particulares, pero a modo de síntesis se expondrán algunas características generales que suelen poseer:

- Presentan la información altamente resumida.
- Se componen de consultas, reportes, análisis interactivos, gráficos (de torta, barras, etc), semáforos, indicadores causa-efecto, etc.
- Permiten evaluar la situación de la empresa con un solo golpe de vista.
- Poseen un formato de diseño visual muy llamativo.

3.6.4. Data Mining

Esta herramienta constituye una poderosa tecnología con un gran potencial que ayuda y brinda soporte a l@s usuari@s, con el fin de permitirles analizar y extraer conocimientos ocultos y predecibles a partir de los datos almacenados en un DW o en un OLTP. Claro que es deseable que la fuente de información sea un DW, por todas las ventajas que aporta.

La integración con el depósito de datos facilita que las decisiones operacionales sean implementadas directamente y monitorizadas.

Implementar Data Mining permitirá analizar factores de influencia en determinados procesos, predecir o estimar variables o comportamientos futuros, segmentar o agrupar ítems similares, además de obtener secuencias de eventos que provocan comportamientos específicos.

Una de las principales ventajas del Data Mining es que, como recién se ha hecho mención, permite inferir comportamientos, modelos, relaciones y estimaciones de los datos, para poder desarrollar predicciones sobre los mismos, sin la necesidad de contar con patrones o reglas preestablecidas, permitiendo tomar decisiones proactivas y basadas en un conocimiento acabado de la información.

Además brinda la posibilidad de dar respuesta a preguntas complicadas sobre los temas de interés, como por ejemplo ¿Qué está pasando?, ¿Por qué? y ¿Qué pasaría sí?, estos cuestionamientos aplicados a una empresa podrían ser: ¿Cuál de los productos de tal marca y clase serán más vendidos en la zona norte en el próximo semestre? y ¿por qué? Además se podrán ver los resultados en forma de reportes tabulares, matriciales, gráficos, tableros, etc.

Entonces, se puede definir Data Mining como una técnica para descubrir patrones y relaciones entre abundantes cantidades de datos, que a simple vista o que mediante otros tipos de análisis no se pueden deducir, ya que tradicionalmente consumiría demasiado tiempo o estaría fuera de las expectativas.

Los sistemas Data Mining se desarrollan bajo lenguajes de última generación basados en Inteligencia Artificial y utilizan métodos matemáticos tales como:

- Redes Neuronales.
- Sistemas Expertos.
- Programación Genética.
- Árboles de Decisión.

Soporta además, sofisticadas operaciones de análisis como los sistemas Scoring, aplicaciones de Detección de Desviación y Detección de Fraude.

Es muy importante tener en cuenta que en las herramientas OLAP y en los reportes y consultas, el análisis parte de una pregunta o hipótesis generada por l@s usuari@s, en cambio Data Mining permite generar estas hipótesis.

Generalmente las herramientas de Data Mining se integran con plataformas de hardware y software existentes (como DW) para incrementar el valor de las fuentes de datos establecidas y para que puedan ser integradas con nuevos productos y sistemas en línea

(como OLAP). En adición a esto, hacer minería de datos sobre un depósito de datos permite entre otras ventajas contar con los beneficios de los procesos ETL y de las técnicas de limpieza de datos, tan necesarios en este tipo de análisis.

3.6.4.1. Redes Neuronales

Se utilizan para construir modelos predictivos no lineales que aprenden a través de entrenamiento y que semejan la estructura de una red neuronal biológica.

Una red neuronal es un modelo computacional con un conjunto de propiedades específicas, como la habilidad de adaptarse o aprender, generalizar u organizar la información, todo ello basado en un procesamiento eminentemente paralelo.

Por ejemplo, las redes neuronales pueden emplearse para:

- Resolver problemas en dominios complejos con variables continuas y categóricas.
- Modelizar relaciones no lineales.
- Clasificar y predecir resultados.

3.6.4.2. Sistemas Expertos

Un sistema experto, puede definirse como un sistema informático (hardware y software) que simula a l@s expert@s human@s en un área de especialización dada.

La principal ventaja de estos sistemas es que l@s usuari@s con poca experiencia pueden resolver problemas que requieren el conocimiento de una persona experta en el tema.

Por ejemplo, los sistemas expertos pueden utilizarse para:

- Realizar transacciones bancarias a través de cajeros automáticos.
- Controlar y regular el flujo de tráfico en las calles y en los ferrocarriles, mediante la operación automática de semáforos.
- Resolver complicados problemas de planificación en los cuales intervienen muchas variables.
- Descubrir relaciones entre diversos conjuntos de variables.

3.6.4.3. Programación Genética

El principal objetivo de la programación genética es lograr que las computadoras aprendan a resolver problemas sin ser explícitamente programadas para solucionarlos, generando de esta manera soluciones a partir de la inducción de los programas. El verdadero valor de esta inducción está fundamentado en que todos los problemas se pueden expresar como un programa de computadora.

Por ejemplo, la programación genética se utiliza para:

- Resolver problemas, para los cuales es difícil y no natural tratar de especificar o restringir con anticipación el tamaño y forma de una solución eventual.
- Analizar sistemas que actúan sobre condiciones inestables en ambientes cambiantes.
- Generar de manera automática programas que solucionen problemas planteados.

3.6.4.4. Árboles de Decisión

Son estructuras de forma de árbol que representan conjuntos de decisiones. Estas decisiones generan reglas para la clasificación de un conjunto de datos, las cuales explican el comportamiento de una variable con relación a otras, y pueden traducirse fácilmente en reglas de negocio.

Son utilizados con finalidad predictiva y de clasificación.

Por ejemplo, los árboles de decisión pueden emplearse para:

- Optimizar respuestas de campañas.
- Identificar clientes potenciales.
- Realizar evaluación de riesgos.

3.6.4.5. Detección de Desviación

Analiza una serie de datos similares, y cuando encuentra un elemento que no coincide con el resto lo considera una desviación.

Usualmente para la detección de la desviación en base de datos grandes se utiliza la información explícita externa a los datos, así como las limitaciones de integridad o modelos predefinidos. En un método lineal, al contrario, se enfoca el problema desde el interior de los datos, empleando la redundancia implícita de los mismos.

Por ejemplo, la detección de desviación puede utilizarse para:

- Descubrir excepciones a modelos establecidos.
- Delimitar grupos que cumplan con condiciones preestablecidas.

3.6.5. EIS

EIS (Executive Information System) proporciona medios sencillos para consultar, analizar y acceder a la información de estado del negocio. Además, pone a disposición facilidades para que l@s usuari@s puedan conseguir los datos buscados rápidamente, empleando el menor tiempo posible para comprender el uso de la herramienta.

Usualmente, EIS se utiliza para analizar los indicadores de performance y desempeño del negocio o área de interés, a través de la presentación de vistas con datos simplificados, altamente consolidados, mayormente estáticos y preferentemente gráficos.

El concepto principal de esta herramienta, se basa en el simple hecho de que l@s ejecutiv@s no poseen tiempo, ni las habilidades necesarias para analizar grandes cantidades de datos.

Al igual que OLAP y Data Mining, los EIS, se pueden aplicar independientemente de la plataforma DW. Pero tener como base un depósito de datos para implementar esta herramienta, conlleva todas las ventajas implícitas del mismo.

3.7. Usuari@s

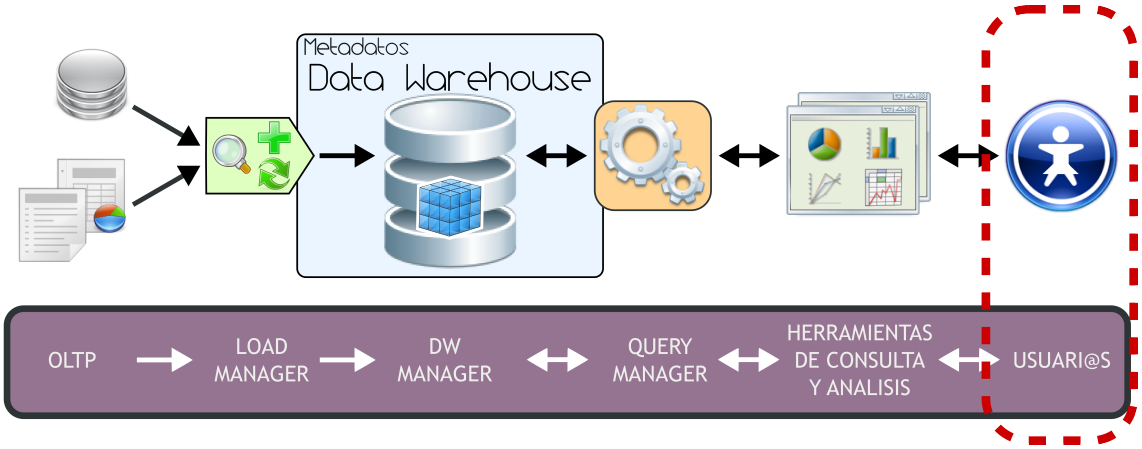


Figura 3.64: Usuari@s.

L@s usuari@s que posee el DW son aquell@s que se encargan de tomar decisiones y de planificar las actividades del negocio, es por ello que se hace tanto énfasis en la integración, limpieza de datos, etc, para poder conseguir que la información posea toda la calidad posible.

Es a través de las herramientas de consulta y análisis, que l@s usuari@s exploran los datos en busca de respuestas para poder tomar decisiones proactivas.

Para comprender mejor a l@s usuari@s del almacén de datos, se hará referencia a las diferencias que estos tienen con respecto a los del OLTP:

- L@s usuari@s que acceden al DW concurrentemente son poc@s, en cambio los que acceden a los OLTP en un tiempo determinado son much@s más, pueden ser cientos o incluso miles. Esto se debe principalmente al tipo de información que contiene cada fuente.
- L@s usuari@s del DW generan por lo general consultas complejas, no predecibles y no anticipadas. Usualmente, cuando se encuentra una respuesta a una consulta se formulan nuevas preguntas más detalladas y así sucesivamente. Es decir, primero se analiza la información a nivel de datos actual para averiguar el “qué”, luego, para obtener mayor detalle y examinar el “cómo”, se trabajan con los datos ligeramente resumidos, derivados de la consulta anterior, y desde allí se puede explorar los datos altamente resumidos. Teniendo en cuenta siempre la posibilidad de utilizar el detalle de datos histórico. Al contrario, l@s usuari@s de los OLTP solo manejan consultas predefinidas.
- L@s usuari@s del DW, generan consultas sobre una gran cantidad de registros, en cambio los del OLTP lo hacen sobre un pequeño grupo. Esto se debe a que como ya se ha mencionado, el depósito contiene información histórica e integra varias fuentes de datos.
- Las consultas de l@s usuari@s del DW no tienen tiempos de respuesta críticos, aunque sí se espera que se produzcan en el mismo día en que fueron realizadas.

Mientras mayor sea el tamaño del depósito y mientras más compleja sea la consulta, mayores serán los tiempos de respuestas. En cambio, las respuestas de las consultas en un OLTP son y deben ser inmediatas.

- En los OLTP, l@s usuari@s típicamente realizan actualizaciones, tales como agregar, modificar, eliminar y consultar algún registro. En cambio en un DW, la única operación que pueden realizar es la de consulta.

Las mencionadas diferencias entre estos dos tipos de usuari@s se pueden apreciar mejor en la siguiente tabla comparativa:

	Usuari@s de OLTP	**Usuari@s de Data Warehouse**
Acceso concurrente	Much@s.	Poc@s.
Tipo de consultas	Predefinidas.	Complejas, no predecibles y no anticipadas.
Registros consultados	Pocos.	Muchos
Tiempo de respuesta	Crítico.	No crítico.
Acciones permitidas	Agregar, modificar, eliminar y consultar.	Consultar.

Figura 3.65: Usuari@s de OLTP vs Usuari@s de DW.

Capítulo 4

CONCEPTOS COMPLEMENTARIOS

4.1. Sistema de Misión Crítica

L@s usuari@s siempre poseen una cierta resistencia al cambio cada vez que se les presenta una nueva herramienta o software, es por ello que al principio no tod@s confiarán en el DW, y por ende no lo utilizarán. Pero a medida que pasa el tiempo y l@s usuari@s pueden comprobar por sí mism@s su buen funcionamiento, se adapten, aprendan a usarlo y disuelvan sus dudas e incertidumbres, tanto el número de usuari@s como su utilización se incrementará de manera significativa.

Además, a medida que las empresas confían y emplean más el almacén de datos, y están más pendientes de la disponibilidad de información que él contiene, como así también en su acceso, este se torna fundamental para la misión del negocio o área que apoya, convirtiéndose paulatinamente en un Sistema de Misión Crítica. Llegando al punto en que, un error en el mismo puede provocar una falla en las actividades del negocio.

En resumen, conforme la empresa comienza a utilizar cada vez más los datos del DW, y desde luego se fían de su buen funcionamiento y desempeño para producir de forma sencilla, rápidas consultas, l@s usuari@s comenzarán a dejar para último momento la generación de la información necesaria. Por este motivo, es de suma importancia que el DW posea una buena performance, seguridad y consistencia, y que todas las aplicaciones o herramientas que lo manipulen estén a disposición en todo momento.

Teniendo toda esta información presente, se puede afirmar que es prácticamente imposible construir un DW perfecto en una primera instancia, es más, tratar de alcanzar este objetivo terminaría por ralentizar los procesos sin conseguir tal fin. De este modo, la maduración del DW se conseguirá paulatinamente con cada nueva iteración o requerimiento.

4.2. Data Mart

Un Data Mart (DM) es la implementación de un DW con alcance restringido a un área funcional, problema en particular, departamento, tema o grupo de necesidades.

Muchos depósitos de datos comienzan siendo Data Mart, para, entre otros motivos, minimizar riesgos y producir una primera entrega en tiempos razonables. Pero, una vez

que estos se han implementado exitosamente, su alcance se irá ampliando paulatinamente.

De acuerdo a las operaciones que se deseen o requieran desarrollar, los DM pueden adoptar las siguientes arquitecturas:

- Top-Down: primero se define el DW y luego se desarrollan, construyen y cargan los DM a partir del mismo. En la siguiente figura se encuentra detallada esta arquitectura:

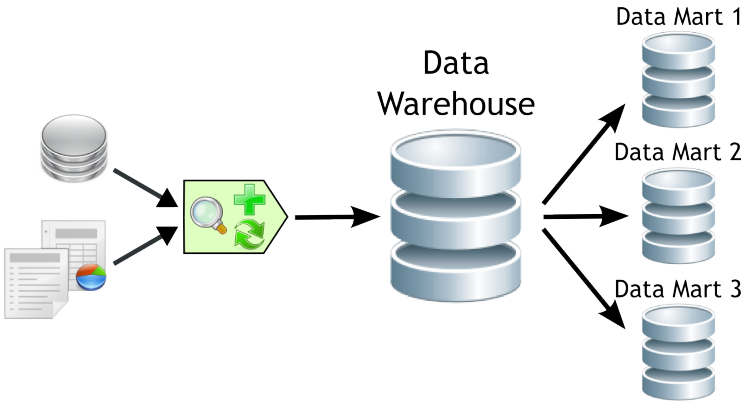


Figura 4.1: Top-Down.

  Como se puede apreciar, el DW es cargado a través de procesos ETL y luego este alimenta a los diferentes DM, cada uno de los cuales recibirá los datos que correspondan al tema o departamento que traten.

  Esta forma de implementación cuenta con la ventaja de no tener que incurrir en complicadas sincronizaciones de hechos, pero requiere una gran inversión y una gran cantidad de tiempo de construcción.

- Bottom-Up: en esta arquitectura, se definen previamente los DM y luego se integran en un DW centralizado. La siguiente figura presenta esta implementación:

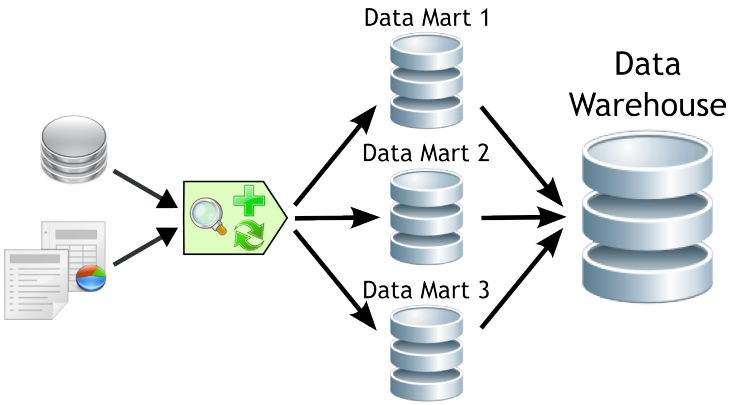


Figura 4.2: Bottom-Up

Los DM se cargan a través de procesos ETL, los cuales suministrarán la información adecuada a cada uno de ellos. En muchas ocasiones, los DM son implementados sin que exista el DW, ya que tienen sus mismas características pero con la particularidad de que están enfocados en un tema específico. Luego de que hayan sido creados y cargados todos los DM, se procederá a su integración con el depósito.

La ventaja que trae aparejada este modelo es que cada DM se crea y pone en funcionamiento en un corto lapso de tiempo y se puede tener una pequeña solución a un costo no tan elevado. Luego que todos los DM estén puestos en marcha, se puede decidir si construir el DW o no. El mayor inconveniente está dado en tener que sincronizar los hechos al momento de la consolidación en el depósito.

Dentro de las ventajas de aplicar un Data Mart a un negocio, se han seleccionado las siguientes:

- Son simples de implementar.
- Conllevan poco tiempo de construcción y puesta en marcha.
- Permiten manejar información confidencial.
- Reflejan rápidamente sus beneficios y cualidades.
- Reducen la demanda del depósito de datos.

4.3. SGBD

Los SGBD (Sistema de Gestión de Base de Datos) son un tipo de software muy específico, dedicados a servir de interfaz entre la base de datos, l@s usuari@s y las aplicaciones que lo utilizan. Se compone de lenguajes de definición, manipulación, consulta y seguridad de datos.

El propósito general de los SGBD es el de manejar de manera clara, sencilla y ordenada un conjunto de datos.

Existen diferentes objetivos que deben cumplir los SGBD, de los cuales se han enumerado los siguientes:

- Hacer transparente a l@s usuari@s los detalles del almacenamiento físico de los datos, mediante varios niveles de abstracción de la información.
- Permitir la realización de cambios a la estructura de la base de datos, sin tener que modificar la aplicación que la emplea.
- Proveer a l@s usuari@s la seguridad de que sus datos no podrán ser accedidos, ni manipulados por quien no tenga permiso para ello. Debido a esto, debe poseer un complejo sistema que maneje grupos, usuari@s y permisos para las diferentes actividades que se pueden realizar dentro del mismo.
- Mantener la integridad de los datos.
- Proporcionar una manera eficiente de realizar copias de seguridad de la información almacenada en ellos, y permitir a partir de estas copias restaurar los datos.
- Controlar el acceso concurrente de l@s usuari@s.
- Facilitar el manejo de grandes volúmenes de información.

Existen dos tipos de SGBD:

1. SGBD Multidimensionales: estos aportan mucha performance al DW en cuanto a la velocidad de respuesta, ya que los datos son almacenados en forma multidimensional, sin embargo son difíciles de gestionar y de mantener.
2. SGBD Relacionales: estos son cada vez más potentes y poseen una interfaz gráfica más avanzada.

4.4. Particionamiento

En un DW, el particionamiento se utiliza mayormente para dividir una tabla de hechos, en varias tablas más pequeñas, a través de un criterio preestablecido. Usualmente, existen dos razones principales, por las cuales se emplea esta práctica:

- Posibilitar un fácil y optimizado mantenimiento del DW y de sus correspondientes ETL.
- Aumentar la performance de las consultas.

Las particiones mejoran los resultados de las consultas, ya que reducen al mínimo el número de registros de una tabla que deben leerse para satisfacer las consultas. Mediante la distribución de los datos en varias tablas, las particiones mejoran la velocidad y la eficacia de las consultas al almacén.

El tiempo es el criterio más comúnmente utilizado para realizar particiones, ya que de esta manera se limita el crecimiento de las tablas y se aumenta la estabilidad.

Las particiones pueden ser lógicas, físicas, horizontales o verticales.

4.5. Business Models

Un Business Model es un representación de los datos desde una perspectiva empresarial, que permite que se pueda visualizar la información del negocio y su respectiva interrelación.

Se compone de entidades, atributos y relaciones, que están enfocados en dar respuesta a las preguntas de la información que se desea conocer.

El Business Model permite definir en comportamiento que tendrá cada miembro dentro de este, como por ejemplo indicar cuáles campos serán utilizados para realizar sumarizaciones y cuál será el criterio empleado a tal fin y cuáles serán los campos que se utilizarán para analizar la información.

Pero lo más importante de este tipo de estructura de datos, es que el mismo se define a través de reglas de negocio y teniendo en cuenta las áreas temáticas que son de interés en la empresa.

A continuación se listarán algunas de sus características más sobresalientes:

- Es completamente independiente de las estructuras organizacionales.
- Plantea la información de la empresa como si fuesen piezas que encajan entre sí.

4.6. Áreas de Datos

Dentro del diseño de la arquitectura de un sistema de Data Warehouse es conveniente tener en consideración los diferentes entornos por los que han de pasar los datos en su camino hacia el DW o hacia los Data Marts de destino. Dada la cantidad de transformaciones que se han de realizar, y que normalmente el DW, además de cumplir su función de soporte a los requerimientos analíticos, realiza una función de integración de datos que van a conformar el Almacén Corporativo[1] y que van a tener que ser consultados también de la manera tradicional por los sistemas operacionales, es muy recomendable crear diferentes áreas de datos en el camino entre los sistemas origen y las herramientas OLAP.

Cada una de estas áreas se distingue por las funciones que realiza, de qué manera se organizan los datos en la misma, y a qué tipo de necesidad pueden dar servicio. El área que se encuentra 'al final del camino' es importante, pero no va a ser la única que almacene los datos que van a explotar las herramientas de reporting.

Tampoco hay una convención estándar sobre lo que abarca exactamente cada área, y la obligatoriedad de utilizar cada una de ellas. Cada proyecto es diferente, e influyen muchos factores como la complejidad, el volumen de información del mismo, si realmente se quiere utilizar el Data Warehouse como almacén corporativo o Sistema Maestro de Datos, o si existen necesidades reales de soporte al reporting operacional.

En los siguientes puntos se explican las áreas de datos que suelen utilizarse, y se perfila una propuesta de arquitectura que hay que adaptar a las necesidades de cada proyecto, y teniendo en cuenta que la utilización de cada área de datos ha de estar justificada. No siempre todas son necesarias.

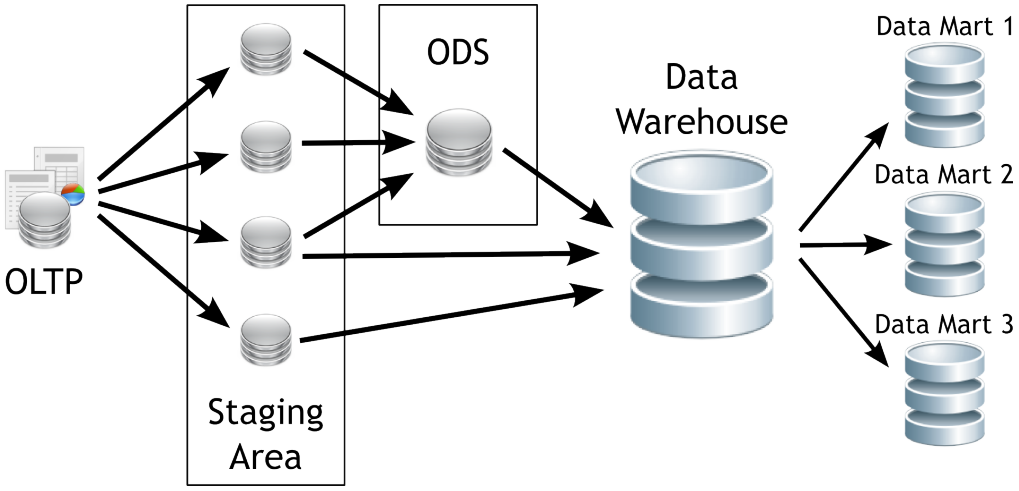


Figura 4.3: Areas de Datos.

4.6.1. Staging Area

Es un área temporal donde se recogen los datos que se necesitan de los sistemas origen.

[1] Ver sección 4.6.2, en la página 78.

Se recogen los datos estrictamente necesarios para las cargas, y se aplica el mínimo de transformaciones a los mismos. No se aplican restricciones de integridad ni se utilizan claves, los datos se tratan como si las tablas fueran ficheros planos. De esta manera se minimiza la afectación a los sistemas origen, la carga es lo más rápida posible para acotar la ventana horaria necesaria, y se reduce también al mínimo la posibilidad de error. Una vez que los datos han sido traspasados, el DW se independiza de los sistemas origen hasta la siguiente carga. Lo único que se suele añadir es algún campo que almacene la fecha de la carga.

Obviamente estos datos no van a dar servicio a ninguna aplicación de reporting, son datos temporales que una vez hayan cumplido su función son eliminados, de hecho en el esquema lógico de la arquitectura muchas veces no aparece, ya que su función es meramente operativa.

Algun@s autor@s consideran que la Staging Area abarca más de lo comentado, o incluso que engloba todo el entorno donde se realizan los procesos de ETL, en este documento se considera sólo como área temporal.

4.6.2. Operational Data Store

Como su nombre indica, este área es la que da soporte a los sistemas operacionales.

El modelo de datos del Almacén de Datos Operacional (ODS) sigue una estructura relacional y normalizada, para que cualquier herramienta de reporting o sistema operacional pueda consultar sus datos. Está dentro del Data Warehouse porque se aprovecha el esfuerzo de integración que supone la creación del Almacén de Datos Corporativo para poder atender también a necesidades operacionales, pero no es obligatorio. Ni siquiera es algo específico del BI, los ODS ya existían antes de que surgieran los conceptos de Data Warehousing y Business Intelligence.

No almacena datos históricos, muestra la imagen del momento actual, aunque eso no significa que no se puedan registrar los cambios.

Los datos del ODS se recogen de la Staging Area, y en este proceso sí que se realizan transformaciones, limpieza de datos y controles de integridad referencial para que los datos estén perfectamente integrados en el modelo relacional normalizado.

Se debe tener en cuenta que la actualización de los datos del ODS no es instantánea, los cambios en los datos de los sistemas origen no se ven reflejados hasta que finaliza la carga correspondiente. Es decir, que los datos se refrescan cada cierto tiempo, cosa que hay que explicar a l@s usuari@s finales, porque los informes que se lancen contra el ODS siempre devolverán información a fecha de la última carga.

Por esta razón es recomendable definir una mayor frecuencia de carga para el ODS que para el Almacén Corporativo. Se puede refrescar el ODS cada 15 minutos, y el resto cada día, por ejemplo.

4.6.3. Almacén de Datos Corporativo

El Almacén de Datos Corporativo (DW) sí que contiene datos históricos, y está orientado a la explotación analítica de la información que recoge.

Las herramientas DSS o de reporting analítico consultan tanto los Data Marts como el Almacén de Datos Corporativo. El DW puede servir consultas en las que se precisa mostrar a la vez información que se encuentre en diferentes Data Marts.

En él se almacenan datos que pueden provenir tanto de la Staging Area como del ODS. Si ya se realizan procesos de transformación e integración en el ODS no se repiten para pasar los mismos datos al Almacén Corporativo. Lo que no se pueda recoger desde el ODS sí que hay que ir a buscarlo a la Staging Area.

El esquema se parece al de un modelo relacional normalizado, pero en él ya se aplican técnicas de desnormalización. No debería contener un número excesivo de tablas ni de relaciones ya que, por ejemplo, muchas relaciones jerárquicas que en un modelo normalizado se implementarían con tablas separadas aquí ya deberían crearse en una misma tabla, que después representará una dimensión.

Otra particularidad es que la mayoría de las tablas han de incorporar campos de fecha para controlar la fecha de carga, la fecha en que se produce un hecho, o el periodo de validez del registro.

Si el Data Warehouse no es demasiado grande, o el nivel de exigencia no es muy elevado en cuanto a los requerimientos 'operacionales', para simplificar la estructura se puede optar por prescindir del ODS, y si es necesario adecuar el Almacén de Datos Corporativo para servir tanto al reporting operacional como al analítico. En este caso, el área resultante sería el DW Corporativo, pero en ocasiones también se denomina como ODS.

4.6.4. Data Mart

Otro área de datos es el lugar donde se crean los Data Marts.

Éstos acostumbran a obtenerse a partir de la información recopilada en el área del Almacén Corporativo, aunque también puede ser a la inversa. Cada Data Mart es como un subconjunto de este almacén, pero orientado a un tema de análisis, normalmente asociado a un departamento de la empresa.

El Data Mart se diseña con estructura multidimensional, cada objeto de análisis es una tabla de hechos enlazada con diversas tablas de dimensiones. Si se diseña siguiendo el Modelo en Estrella habrá prácticamente una tabla para cada dimensión, es la versión más desnormalizada. Si se sigue un modelo de Copo de Nieve las tablas de dimensiones estarán menos desnormalizadas y para cada dimensión se podrán utilizar varias tablas enlazadas jerárquicamente.

Este área puede residir en la misma base de datos que las demás si la herramienta de explotación es de tipo ROLAP, o también puede crearse ya fuera de la BD, en la estructura de datos propia que generan las aplicaciones de tipo MOLAP, más conocida como los cubos multidimensionales.

Si se sigue una aproximación Top-down para la creación de los Data Mart, el paso del área de DW a esta ha de ser bastante simple, cosa que además proporciona una cierta independencia sobre el software que se utiliza para el reporting analítico. Si por cualquier razón es necesario cambiar la herramienta de OLAP hay que hacer poco más que redefinir los metadatos y regenerar los cubos, y si el cambio es entre dos de tipo ROLAP ni siquiera esto último sería necesario. En cualquier caso, las áreas anteriores no

tienen porqué ser modificadas.

Parte II

HEFESTO: Metodología para la Construcción de un Data Warehouse

RESUMEN

En esta segunda parte de la publicación, se presentará una metodología propia para la construcción de un Data Warehouse, que partirá de la recolección de requerimientos y necesidades de información de l@s usuari@s, y concluirá en la confección de un esquema lógico y sus respectivos procesos de extracción, transformación y carga de datos. Además, se ejemplificará cada etapa de la metodología a través de su aplicación a una empresa real, que servirá de guía para que se puedan visualizar los resultados que se esperan de cada paso y para clarificar los conceptos enunciados.

Primero, se describirán los aspectos más sobresalientes de la metodología y luego se explicará cada paso con su respectiva aplicación. Finalmente, se expondrán algunas consideraciones que deben tenerse en cuenta al momento de construir e implementar un Data Warehouse.

El principal objetivo es facilitar el arduo trabajo que significa construir un Data Warehouse desde cero, aportando información que permitirá aumentar la performance del mismo. En adición a ello, esta metodología estará orientada a evitar el tedio que provoca el tener que seguir pasos sin terminar de comprender el por qué de los mismos.

Adicional a todo esto, se ejemplificará la creación de cubos multidimensionales basados en el DW resultante del caso práctico.

Capítulo 5

METODOLOGÍA HEFESTO

5.1. Introducción

En esta sección se presentará la metodología HEFESTO, que permitirá la construcción de Data Warehouse de forma sencilla, ordenada e intuitiva. Su nombre fue inspirado en el dios griego de la construcción y el fuego, y su logotipo es el siguiente:

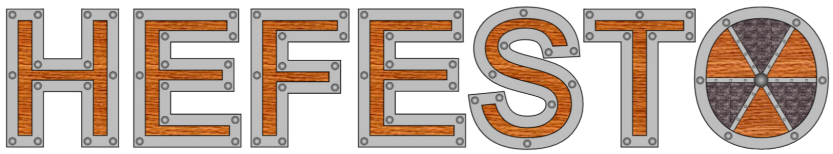


Figura 5.1: Metodología HEFESTO, logotipo.

Figura 5.2: Metodología HEFESTO, logotipo versión 2.0.

HEFESTO es una metodología propia, cuya propuesta está fundamentada en una muy amplia investigación, comparación de metodologías existentes, experiencias propias en procesos de confección de almacenes de datos. Cabe destacar que HEFESTO está en continua evolución, y se han tenido en cuenta, como gran valor agregado, todos los feedbacks que han aportado quienes han utilizado esta metodología en diversos países y con diversos fines.

La idea principal, es comprender cada paso que se realizará, para no caer en el tedio de tener que seguir un método al pie de la letra sin saber exactamente qué se está haciendo, ni por qué.

La construcción e implementación de un DW puede adaptarse muy bien a cualquier ciclo de vida de desarrollo de software, con la salvedad de que para algunas fases en particular, las acciones que se han de realizar serán muy diferentes. Lo que se debe tener muy en cuenta, es no entrar en la utilización de metodologías que requieran fases extensas de reunión de requerimientos y análisis, fases de desarrollo monolítico que conlleve demasiado tiempo y fases de despliegue muy largas. Lo que se busca, es entregar una primera implementación que satisfaga una parte de las necesidades, para demostrar las ventajas del DW y motivar a l@s usuari@s.

La metodología HEFESTO, puede ser embebida en cualquier ciclo de vida que cumpla con la condición antes declarada.

Con el fin de que se llegue a una total comprensión de cada paso o etapa, se acompañará con la implementación en una empresa real, para demostrar los resultados que se deben obtener y ejemplificar cada concepto.

5.2. Descripción

La metodología HEFESTO puede resumirse a través del siguiente gráfico:

Figura 5.3: Metodología HEFESTO, pasos.

Como se puede apreciar, se comienza recolectando las necesidades de información de l@s usuari@s y se obtienen las preguntas claves del negocio. Luego, se deben identificar los indicadores resultantes de los interrogativos y sus respectivas perspectivas de análisis, mediante las cuales se construirá el modelo conceptual de datos del DW.

Después, se analizarán los OLTP para determinar cómo se construirán los indicadores, señalar las correspondencias con los datos fuentes y para seleccionar los campos de estudio de cada perspectiva.

Una vez hecho esto, se pasará a la construcción del modelo lógico del depósito, en

donde se definirá cuál será el tipo de esquema que se implementará. Seguidamente, se confeccionarán las tablas de dimensiones y las tablas de hechos, para luego efectuar sus respectivas uniones.

Por último, utilizando técnicas de limpieza y calidad de datos, procesos ETL, etc, se definirán políticas y estrategias para la Carga Inicial del DW y su respectiva actualización.

5.3. Características

Esta metodología cuenta con las siguientes características:

- Los objetivos y resultados esperados en cada fase se distinguen fácilmente y son sencillos de comprender.
- Se basa en los requerimientos de l@s usuari@s, por lo cual su estructura es capaz de adaptarse con facilidad y rapidez ante los cambios en el negocio.
- Reduce la resistencia al cambio, ya que involucra a l@s usuari@s finales en cada etapa para que tome decisiones respecto al comportamiento y funciones del DW.
- Utiliza modelos conceptuales y lógicos, los cuales son sencillos de interpretar y analizar.
- Es independiente del tipo de ciclo de vida que se emplee para contener la metodología.
- Es independiente de las herramientas que se utilicen para su implementación.
- Es independiente de las estructuras físicas que contengan el DW y de su respectiva distribución.
- Cuando se culmina con una fase, los resultados obtenidos se convierten en el punto de partida para llevar a cabo el paso siguiente.
- Se aplica tanto para Data Warehouse como para Data Mart.

5.4. Empresa analizada

Antes de comenzar con el primer paso, es menester describir las características principales de la empresa a la cual se le aplicará la metodología HEFESTO, así se podrá tener como base un ámbito predefinido y se comprenderá mejor cada decisión que se tome con respecto a la implementación y diseño del DW.

Además, este análisis ayudará a conocer el funcionamiento y accionar de la empresa, lo que permitirá examinar e interpretar de forma óptima las necesidades de información de la misma, como así también apoyará a una mejor construcción y adaptación del depósito de datos.

La descripción de la empresa se encuentra en el Apéndice A (página 133).

5.5. Pasos y aplicación metodológica

5.5.1. PASO 1) ANÁLISIS DE REQUERIMIENTOS

Lo primero que se hará será identificar los requerimientos de l@s usuari@s a través de preguntas que expliciten los objetivos de su organización. Luego, se analizarán estas preguntas a fin de identificar cuáles serán los indicadores y perspectivas que serán tomadas en cuenta para la construcción del DW. Finalmente se confeccionará un modelo conceptual en donde se podrá visualizar el resultado obtenido en este primer paso.

Es muy importante tener en cuenta que HEFESTO se puede utilizar para construir un Data Warehouse o un Data Mart a la vez, es decir, si se requiere construir por ejemplo dos Data Marts, se deberá aplicar la metodología dos veces, una por cada Data Mart. Del mismo modo, si se analizan dos áreas de interés de negocio, como el área de ”Ventas” y ”Compras”, se deberá aplicar la metodología dos veces.

5.5.1.1. a) Identificar preguntas

El primer paso comienza con el acopio de las necesidades de información, el cual puede llevarse a cabo a través de muy variadas y diferentes técnicas, cada una de las cuales poseen características inherentes y específicas, como por ejemplo entrevistas, cuestionarios, observaciones, etc.

El análisis de los requerimientos de l@s diferentes usuari@s, es el punto de partida de esta metodología, ya que ell@s son l@s que deben, en cierto modo, guiar la investigación hacia un desarrollo que refleje claramente lo que se espera del depósito de datos, en relación a sus funciones y cualidades.

El objetivo principal de esta fase, es la de obtener e identificar las necesidades de información clave de alto nivel, que es esencial para llevar a cabo las metas y estrategias de la empresa, y que facilitará una eficaz y eficiente toma de decisiones.

Debe tenerse en cuenta que dicha información, es la que proveerá el soporte para desarrollar los pasos sucesivos, por lo cual, es muy importante que se preste especial atención al relevar los datos.

Una forma de asegurarse de que se ha realizado un buen análisis, es corroborar que el resultado del mismo haga explícitos los objetivos estratégicos planteados por la empresa que se está estudiando.

Otra forma de encaminar el relevamiento, es enfocar las necesidades de información en los procesos principales que desarrolle la empresa en cuestión.

La idea central es, que se formulen preguntas complejas sobre el negocio, que incluyan variables de análisis que se consideren relevantes, ya que son estas las que permitirán estudiar la información desde diferentes perspectivas.

Un punto importante que debe tenerse muy en cuenta, es que la información debe estar soportada de alguna manera por algún OLTP, ya que de otra forma, no se podrá elaborar el DW.

Caso práctico:
Se indagó a l@s usuari@s en busca de sus necesidades de información, pero las mismas abarcaban casi todas las actividades de la empresa, por lo cual se les pidió que escogieran el proceso que considerasen más importante en las actividades diarias de la misma y que estuviese soportado de alguna manera por algún OLTP. El proceso elegido fue el de Ventas.

A continuación, se procedió a identificar qué era lo que les interesaba conocer acerca de este proceso y cuáles eran las variables o perspectivas que debían tenerse en cuenta para poder tomar decisiones basadas en ello.

Se les preguntó cuáles eran según ell@s, los indicadores que representan de mejor modo el proceso de Ventas y qué sería exactamente lo que se desea analizar del mismo. La respuesta obtenida, fue que se deben tener en cuenta y consultar datos sobre la cantidad de unidades vendidas y el monto total de ventas.

Luego se les preguntó cuáles serían las variables o perspectivas desde las cuales se consultarán dichos indicadores. Para simplificar esta tarea se les presentó una serie de ejemplos concretos de otros casos similares.

Las preguntas de negocio obtenidas fueron las siguientes:

- Se desea conocer cuántas unidades de cada producto fueron vendidas a sus clientes en un periodo determinado. O en otras palabras: "Unidades vendidas de cada producto a cada cliente en un tiempo determinado".
- Se desea conocer cuál fue el monto total de ventas de productos a cada cliente en un periodo determinado. O en otras palabras: "Monto total de ventas de cada producto a cada cliente en un tiempo determinado".

Debido a que la dimensión Tiempo es un elemento fundamental en el DW, se hizo hincapié en él. Además, se puso mucho énfasis en dejar en claro a l@s usuari@s, a través de ejemplos prácticos, que es este componente el que permitirá tener varias versiones de los datos a fin de realizar un correcto análisis posterior.

Como se puede apreciar, las necesidades de información expuestas están acorde a los objetivos y estrategias de la empresa, ya que es precisamente esta información requerida la que proveerá un ámbito para la toma de decisiones, que en este caso permitirá analizar el comportamiento de l@s client@s a l@s que se pretende satisfacer ampliamente, para así lograr obtener una ventaja competitiva y maximizar las ganancias.

5.5.1.2. b) Identificar indicadores y perspectivas

Una vez que se han establecido las preguntas de negocio, se debe proceder a su descomposición para descubrir los indicadores que se utilizarán y las perspectivas de análisis que intervendrán.

Para ello, se debe tener en cuenta que los indicadores, para que sean realmente efectivos son, en general, valores numéricos y representan lo que se desea analizar concretamente, por ejemplo: saldos, promedios, cantidades, sumatorias, fórmulas, etc.

En cambio, las perspectivas se refieren a los objetos mediante los cuales se quiere examinar los indicadores, con el fin de responder a las preguntas planteadas, por ejemplo: clientes, proveedores, sucursales, países, productos, rubros, etc. Cabe destacar, que el Tiempo es muy comúnmente una perspectiva.

Caso práctico:

A continuación, se analizarán las preguntas obtenidas en el paso anterior y se detallarán cuáles son sus respectivos indicadores y perspectivas.

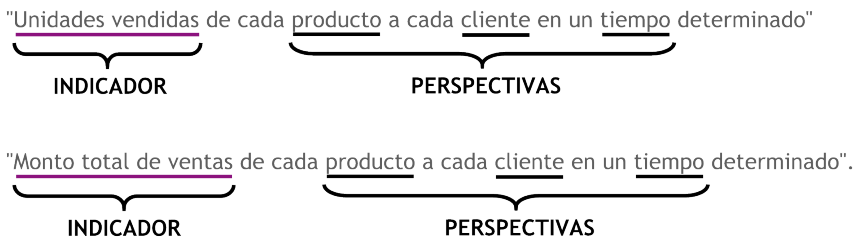


Figura 5.4: Caso práctico, indicadores y perspectivas.

En síntesis, los indicadores son:

- Unidades vendidas.
- Monto total de ventas.

Y las perspectivas de análisis son:

- Clientes.
- Productos.
- Tiempo.

5.5.1.3. c) Modelo Conceptual

En esta etapa, se construirá un modelo conceptual[1] a partir de los indicadores y perspectivas obtenidas en el paso anterior.

A través de este modelo, se podrá observar con claridad cuáles son los alcances del proyecto, para luego poder trabajar sobre ellos, además al poseer un alto nivel de definición de los datos, permite que pueda ser presentado ante l@s usuari@s y explicado con facilidad.

La representación gráfica del modelo conceptual es la siguiente:

[1]Modelo Conceptual: descripción de alto nivel de la estructura de la base de datos, en la cual la información es representada a través de objetos, relaciones y atributos.

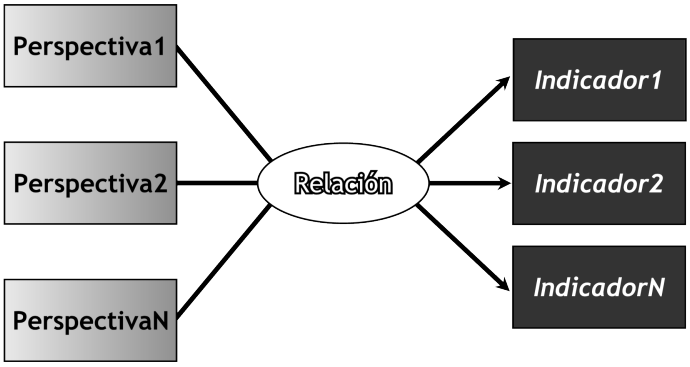


Figura 5.5: Modelo Conceptual.

A la izquierda se colocan las perspectivas seleccionadas, que serán unidas a un óvalo central que representa y lleva el nombre de la relación que existe entre ellas. La relación, constituye el proceso o área de estudio elegida. De dicha relación y entrelazadas con flechas, se desprenden los indicadores, estos se ubican a la derecha del esquema.

Como puede apreciarse en la figura anterior, el modelo conceptual permite de un solo vistazo y sin poseer demasiados conocimientos previos, comprender cuáles serán los resultados que se obtendrán, cuáles serán las variables que se utilizarán para analizarlos y cuál es la relación que existe entre ellos.

Caso práctico:
El modelo conceptual resultante de los datos que se han recolectado, es el siguiente:

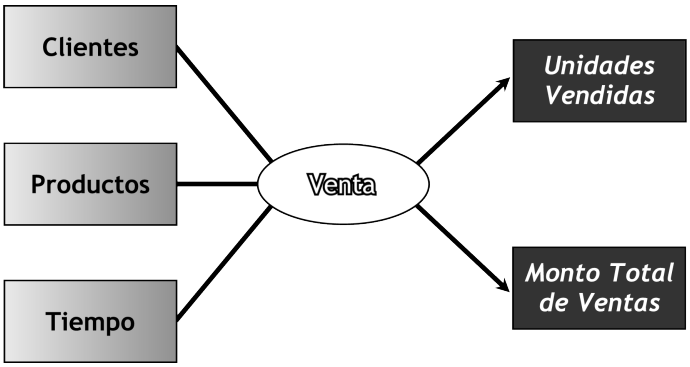


Figura 5.6: Caso práctico, Modelo Conceptual.

Como puede observarse, la relación mediante la cuál se unen las diferentes perspectivas, para obtener como resultado los indicadores requeridos por l@s usuari@s, es precisamente ”Venta”.

5.5.2. PASO 2) ANÁLISIS DE LOS OLTP

Seguidamente, se analizarán las fuentes OLTP para determinar cómo serán calculados los indicadores y para establecer las respectivas correspondencias entre el modelo conceptual creado en el paso anterior y las fuentes de datos. Luego, se definirán qué campos se incluirán en cada perspectiva. Finalmente, se ampliará el modelo conceptual con la información obtenida en este paso.

5.5.2.1. a) Conformar indicadores

En este paso se deberán explicitar cómo se calcularán los indicadores, definiendo los siguientes conceptos para cada uno de ellos:

- Hecho/s que lo componen, con su respectiva fórmula de cálculo. Por ejemplo: Hecho1 + Hecho2.
- Función de sumarización que se utilizará para su agregación. Por ejemplo: SUM, AVG, COUNT, etc.

Caso práctico:
Los indicadores se calcularán de la siguiente manera:

- "Unidades Vendidas":
  - Hechos: Unidades Vendidas.
  - Función de sumarización: SUM.

    Aclaración: el indicador "Unidades Vendidas" representa la sumatoria de las unidades que se han vendido de un producto en particular.

- "Monto Total de Ventas":
  - Hechos: (Unidades Vendidas) * (Precio de Venta).
  - Función de sumarización: SUM.

    Aclaración: el indicador "Monto Total de Ventas" representa la sumatoria del monto total que se ha vendido de cada producto, y se obtiene al multiplicar las unidades vendidas, por su respectivo precio.

5.5.2.2. b) Establecer correspondencias

El objetivo de este paso, es el de examinar los OLTP disponibles que contengan la información requerida, como así también sus características, para poder identificar las correspondencias entre el modelo conceptual y las fuentes de datos.

La idea es, que todos los elementos del modelo conceptual estén correspondidos en los OLTP.

Caso práctico:
En el OLTP de la empresa analizada, el proceso de venta está representado por el diagrama de entidad relación[2] de la siguiente figura.

[2]Diagrama de Entidad Relación: representa la información a través de entidades, relaciones, cardinalidades, claves, atributos y jerarquías de generalización.

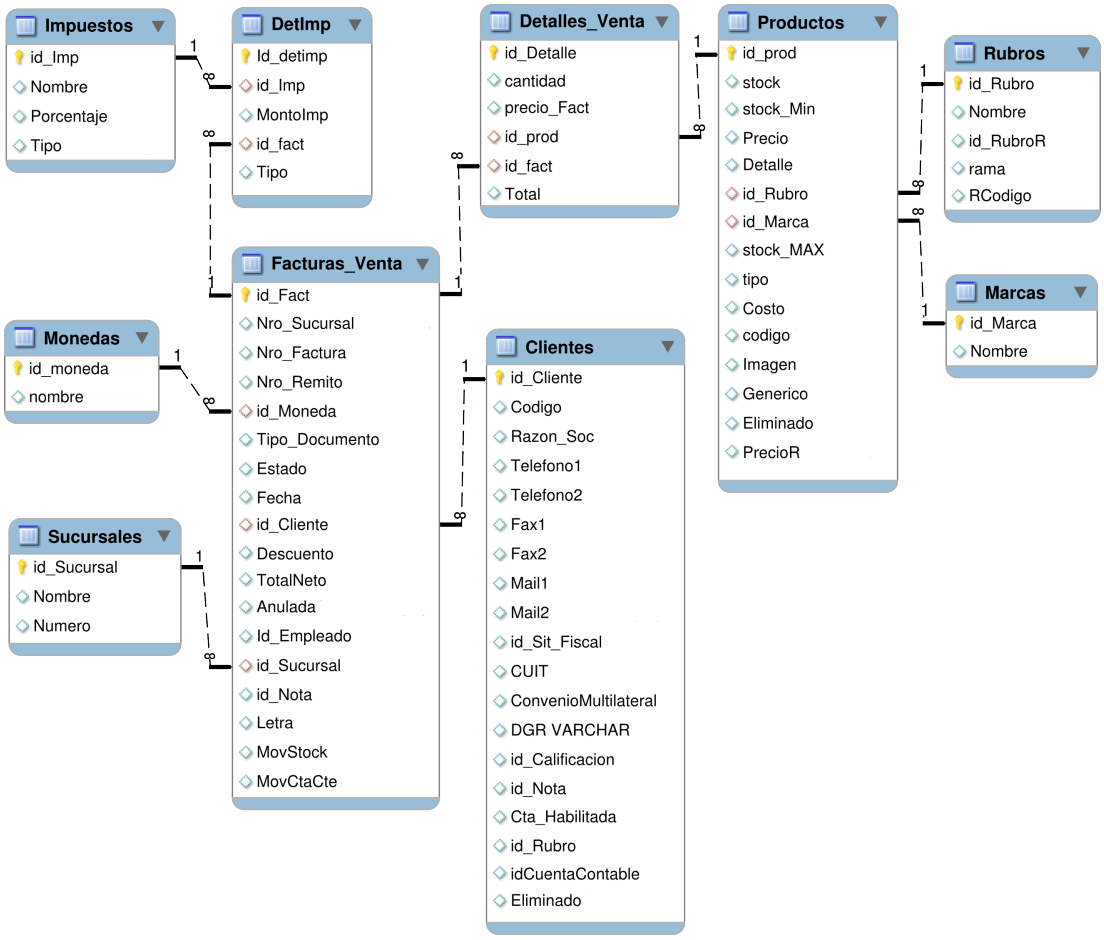


Figura 5.7: Caso práctico, Diagrama de Entidad Relación.

A continuación, se expondrá la correspondencia entre los dos modelos:

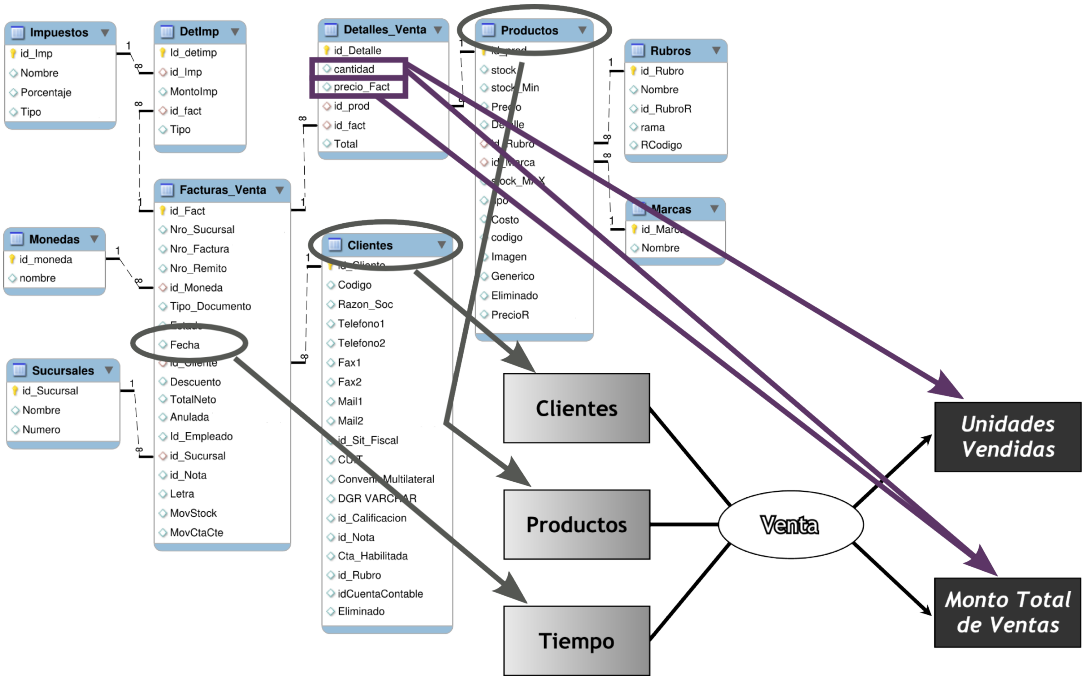


Figura 5.8: Caso práctico, correspondencia.

Las relaciones identificadas fueron las siguientes:

- La tabla ”Productos” se relaciona con la perspectiva ”Productos”.
- La tabla ”Clientes” con la perspectiva ”Clientes”.
- El campo ”fecha” de la tabla ”Facturas_Venta” con la perspectiva ”Tiempo” (debido a que es la fecha principal en el proceso de venta).
- El campo ”cantidad” de la tabla ”Detalles_Venta” con el indicador ”Unidades Vendidas”.
- El campo ”cantidad” de la tabla ”Detalles_Venta” multiplicado por el campo ”precio_Fact” de la misma tabla, con el indicador ”Monto Total de Ventas”.

5.5.2.3. c) Nivel de granularidad

Una vez que se han establecido las relaciones con los OLTP, se deben seleccionar los campos que contendrá cada perspectiva, ya que será a través de estos por los que se examinarán y filtrarán los indicadores.

Para ello, basándose en las correspondencias establecidas en el paso anterior, se debe presentar a l@s usuari@s los datos de análisis disponibles para cada perspectiva. Es muy importante conocer en detalle que significa cada campo y/o valor de los datos encontrados en los OLTP, por lo cual, es conveniente investigar su sentido, ya sea a través de diccionarios de datos, reuniones con l@s encargad@s del sistema, análisis de los datos propiamente dichos, etc.

Luego de exponer frente a l@s usuari@s los datos existentes, explicando su significado, valores posibles y características, est@s deben decidir cuales son los que consideran relevantes para consultar los indicadores y cuales no.

Con respecto a la perspectiva “Tiempo”, es muy importante definir el ámbito mediante el cual se agruparán o sumarizarán los datos. Sus campos posibles pueden ser: día de la semana, quincena, mes, trimestres, semestre, año, etc.

Al momento de seleccionar los campos que integrarán cada perspectiva, debe prestarse mucha atención, ya que esta acción determinará la granularidad de la información encontrada en el DW.

Caso práctico:

De acuerdo a las correspondencias establecidas, se analizaron los campos residentes en cada tabla a la que se hacia referencia, a través de dos métodos diferentes. Primero se examinó la base de datos para intuir los significados de cada campo, y luego se consultó con el encargado del sistema sobre algunos aspectos de los cuales no se comprendía su sentido.

De todas formas, y como puede apreciarse en el diagrama de entidad relación antes expuesto, los nombres de los campos son bastante explícitos y se deducen con facilidad, pero aún así fue necesario investigarlos para evitar cualquier tipo de inconvenientes.

- Con respecto a la perspectiva ”Clientes”, los datos disponibles son los siguientes:
  - id_Cliente: es la clave primaria de la tabla ”Clientes”, y representa unívocamente a un cliente en particular.
  - Codigo: representa el código del cliente, este campo es calculado de acuerdo a una combinación de las iniciales del nombre del cliente, el grupo al que pertenece y un número incremental.
  - Razon_Soc: nombre o razón social del cliente.
  - Telefono1: número de teléfono del cliente.
  - Telefono2: segundo número telefónico del cliente.
  - Fax1: número de fax del cliente.
  - Fax2: segundo número de fax del cliente.
  - Mail1: dirección de correo electrónico del cliente.
  - Mail2: segunda dirección de correo del cliente.
  - id_Sit_Fiscal: representa a través de una clave foránea el tipo de situación fiscal que posee el cliente. Por ejemplo: Consumidor Final, Exento, Responsable No Inscripto, Responsable Inscripto.
  - CUIT: número de C.U.I.T. del cliente.
  - ConvenioMultilateral: indica si el cliente posee o no convenio multilateral.
  - DGR: número de D.G.R. del cliente.
  - id_Clasificación: representa a través de una clave foránea la clasificación del cliente. Por ejemplo: Muy Bueno, Bueno, Regular, Malo, Muy Malo.
  - id_Nota: representa a través de una clave foránea una observación realizada acerca del cliente.
  - Cta_Habilitada: indica si el cliente posee su cuenta habilitada.
  - id_Rubro: representa a través de una clave foránea el grupo al que pertenece el cliente. Por ejemplo: Bancos, Construcción, Educación Privada, Educación Pública, Particulares.
  - idCuentaContable: representa la cuenta contable asociada al cliente, la cual se utilizará para imputar los movimientos contables que este genere.

- Eliminado: indica si el cliente fue eliminado o no. Si fue eliminado, no figura en las listas de clientes actuales.

- En la perspectiva ”Productos”, los datos que se pueden utilizar son los siguientes:
  - id_prod: es la clave primaria de la tabla ”Productos”, y representa unívocamente a un producto en particular.
  - stock: stock actual del producto.
  - stock_min: stock mínimo del producto, se utiliza para dar alerta si el stock actual está cerca del mismo, al ras o si ya lo superó.
  - Precio: precio de venta del producto.
  - Detalle: nombre o descripción del producto.
  - id_Rubro: representa a través de una clave foránea el rubro al que pertenece el producto.
  - id_Marca: representa a través de una clave foránea la marca a la que pertenece el producto.
  - stock_MAX: stock máximo del producto. Al igual que ”stock_min”, se utiliza para dar alertas del nivel de stock actual.
  - tipo: clasificación del producto. Por ejemplo: Producto, Servicio, Compuesto.
  - Costo: precio de costo del producto.
  - codigo: representa el código del producto, este campo es calculado de acuerdo a una combinación de las iniciales del nombre del producto, el rubro al que pertenece y un número incremental.
  - Imagen: ruta de acceso a una imagen o dibujo mediante la cual se quiera representar al producto. Este campo no es utilizado actualmente.
  - Generico: indica si el producto es genérico o no.
  - Eliminado: indica si el producto fue eliminado o no. Si fue eliminado, no figura en las listas de productos actuales.
  - PrecioR: precio de lista del producto.

- Con respecto a la perspectiva ”Tiempo”, que es la que determinará la granularidad del depósito de datos, los datos más típicos que pueden emplearse son los siguientes:
  - Año.
  - Semestre.
  - Cuatrimestre.
  - Trimestre.
  - Número de mes.
  - Nombre del mes.
  - Quincena.
  - Semana.
  - Número de día.
  - Nombre del día.

Una vez que se recolectó toda la información pertinente y se consultó con l@s usuari@s cuales eran los datos que consideraban de interés para analizar los indicadores ya expuestos, los resultados obtenidos fueron los siguientes:

- Perspectiva ”Clientes”:
  - ”Razon_Soc” de la tabla ”Clientes”. Ya que este hace referencia al nombre del cliente.
- Perspectiva ”Productos”:
  - ”detalle” de la tabla ”Productos”. Ya que este hace referencia al nombre del producto.
  - ”Nombre” de la tabla ”Marcas”. Ya que esta hace referencia a la marca a la que pertenece el producto. Este campo es obtenido a través de la unión con la tabla ”Productos”

- Perspectiva "Tiempo":
  - "Mes". Referido al nombre del mes.
  - "Trimestre".
  - "Año".

5.5.2.4. d) Modelo Conceptual ampliado

En este paso, y con el fin de graficar los resultados obtenidos en los pasos anteriores, se ampliará el modelo conceptual, colocando bajo cada perspectiva los campos seleccionados y bajo cada indicador su respectiva fórmula de cálculo. Gráficamente:

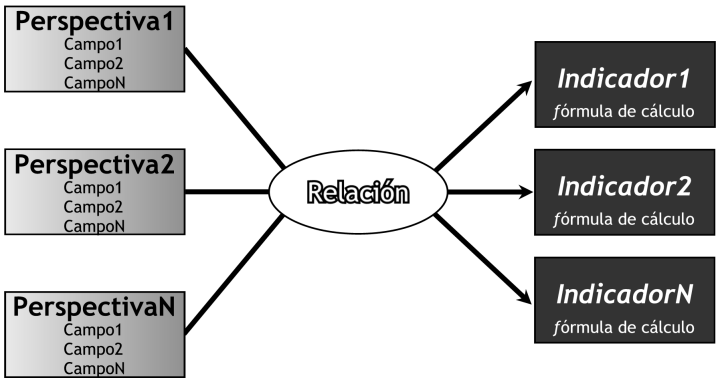


Figura 5.9: Modelo Conceptual ampliado.

Caso práctico:
Teniendo esto en cuenta, se completará el diseño del diagrama conceptual:

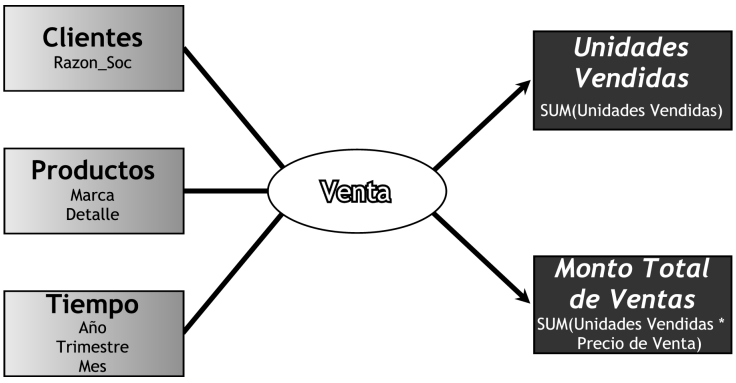


Figura 5.10: Caso práctico, Modelo Conceptual ampliado.

5.5.3. PASO 3) MODELO LÓGICO DEL DW

A continuación, se confeccionará el modelo lógico[3] de la estructura del DW, teniendo como base el modelo conceptual que ya ha sido creado. Para ello, primero se definirá el tipo de modelo que se utilizará y luego se llevarán a cabo las acciones propias al caso, para diseñar las tablas de dimensiones y de hechos. Finalmente, se realizarán las uniones pertinentes entre estas tablas.

5.5.3.1. a) Tipo de Modelo Lógico del DW

Se debe seleccionar cuál será el tipo de esquema que se utilizará para contener la estructura del depósito de datos, que se adapte mejor a los requerimientos y necesidades de l@s usuari@s. Es muy importante definir objetivamente si se empleará un esquema en estrella, constelación o copo de nieve, ya que esta decisión afectará considerablemente la elaboración del modelo lógico.

Caso práctico:
El esquema que se utilizará será en estrella, debido a sus características, ventajas y diferencias con los otros esquemas.

5.5.3.2. b) Tablas de dimensiones

En este paso se deben diseñar las tablas de dimensiones que formaran parte del DW.

Para los tres tipos de esquemas, cada perspectiva definida en en modelo conceptual constituirá una tabla de dimensión. Para ello deberá tomarse cada perspectiva con sus campos relacionados y realizarse el siguiente proceso:

- Se elegirá un nombre que identifique la tabla de dimensión.
- Se añadirá un campo que represente su clave principal.
- Se redefinirán los nombres de los campos si es que no son lo suficientemente intuitivos.

Gráficamente:

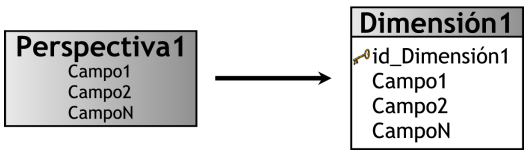


Figura 5.11: Diseño de tablas de dimensiones.

Para los esquemas copo de nieve, cuando existan jerarquías dentro de una tabla de dimensión, esta tabla deberá ser normalizada. Por ejemplo, se tomará como referencia la siguiente tabla de dimensión y su respectivas relaciones padre-hijo entre sus campos:

[3] Modelo Lógico: representación de una estructura de datos, que puede procesarse y almacenarse en algún SGBD.

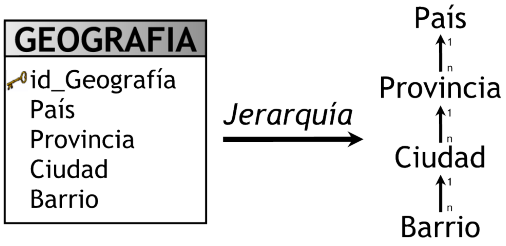


Figura 5.12: Jerarquía de ”GEOGRAFIA”.

Entonces, al normalizar esta tabla se obtendrá:

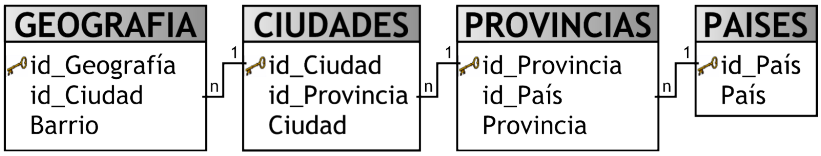


Figura 5.13: Normalización de ”GEOGRAFIA”.

Caso práctico:
A continuación, se diseñaran las tablas de dimensiones.

- Perspectiva “Clientes”:
  - La nueva tabla de dimensión tendrá el nombre “CLIENTE”.
  - Se le agregará una clave principal con el nombre “idCliente”.
  - Se modificará el nombre del campo “Razon_Soc” por “Cliente”.

  Se puede apreciar el resultado de estas operaciones en la siguiente gráfica:

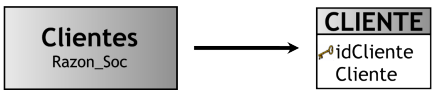


Figura 5.14: Caso práctico, tabla de dimensión ”CLIENTE”.

- Perspectiva “Productos”:
  - La nueva tabla de dimensión tendrá el nombre “PRODUCTO”.
  - Se le agregará una clave principal con el nombre “idProducto”.
  - El nombre del campo “Marca” no será cambiado.
  - Se modificará el nombre del campo “Detalle” por “Producto”.

  Se puede apreciar el resultado de estas operaciones en la siguiente gráfica:

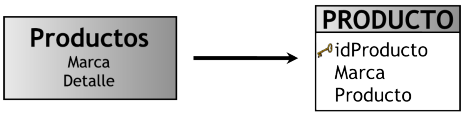


Figura 5.15: Caso práctico, tabla de dimensión ”PRODUCTO”.

- Perspectiva “Tiempo”:
  - La nueva tabla de dimensión tendrá el nombre “FECHA”.
  - Se le agregará una clave principal con el nombre “idFecha”.
  - El nombre los campos no serán modificados.

  Se puede apreciar el resultado de estas operaciones en la siguiente gráfica:

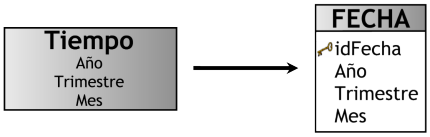


Figura 5.16: Caso práctico, tabla de dimensión ”FECHA”.

5.5.3.3. c) Tablas de hechos

En este paso, se definirán las tablas de hechos, que son las que contendrán los hechos a través de los cuales se construirán los indicadores de estudio.

- Para los esquemas en estrella y copo de nieve, se realizará lo siguiente:
  - Se le deberá asignar un nombre a la tabla de hechos que represente la información analizada, área de investigación, negocio enfocado, etc.
  - Se definirá su clave primaria, que se compone de la combinación de las claves primarias de cada tabla de dimensión relacionada.
  - Se crearán tantos campos de hechos como indicadores se hayan definido en el modelo conceptual y se les asignará los mismos nombres que estos. En caso que se prefiera, podrán ser nombrados de cualquier otro modo.

    Gráficamente:

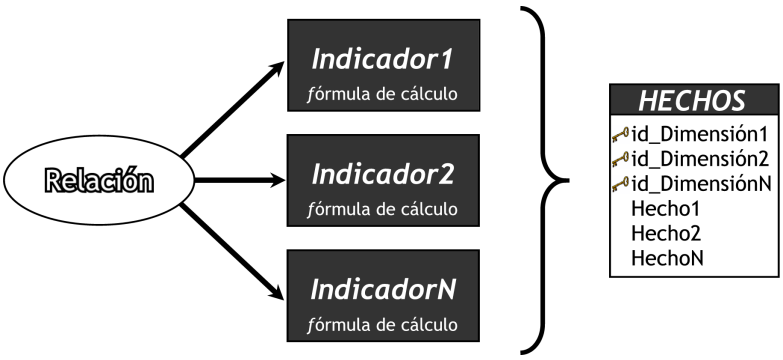


Figura 5.17: Tabla de hechos.

- Para los esquemas constelación se realizará lo siguiente:
  - Las tablas de hechos se deben confeccionar teniendo en cuenta el análisis de las preguntas realizadas por l@s usuari@s en pasos anteriores y sus respectivos indicadores y perspectivas.
  - Cada tabla de hechos debe poseer un nombre que la identifique, contener sus hechos correspondientes y su clave debe estar formada por la combinación de las claves de las tablas de dimensiones relacionadas.
  - Al diseñar las tablas de hechos, se deberá tener en cuenta:
    - Caso 1: Si en dos o más preguntas de negocio figuran los mismos indicadores pero con diferentes perspectivas de análisis, existirán tantas tablas de hechos como preguntas cumplan esta condición. Por ejemplo:

"Analizar el Indicador1 por Perspectiva1 y por Perspectiva2".
"Analizar el Indicador1 por Perspectiva2 y por Perspectiva3".

Figura 5.18: Caso 1, preguntas.

Entonces se obtendrá:

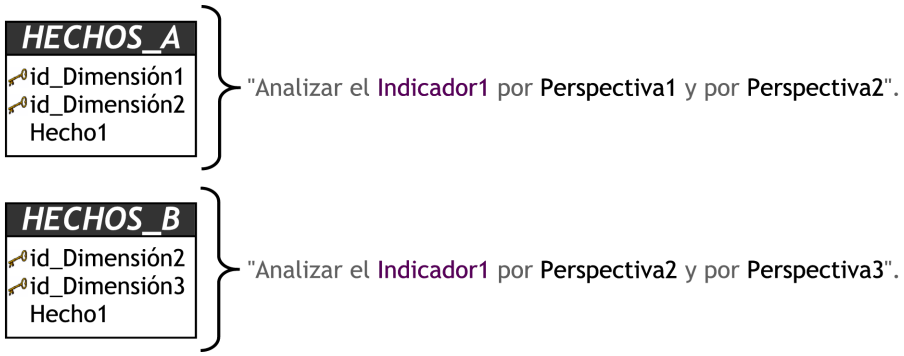


Figura 5.19: Caso 1, diseño de tablas de hechos.

- Caso 2: Si en dos o más preguntas de negocio figuran diferentes indicadores con diferentes perspectivas de análisis, existirán tantas tablas de hechos como preguntas cumplan esta condición. Por ejemplo:

"Analizar el Indicador1 por Perspectiva1 y por Perspectiva2".
"Analizar el Indicador2 por Perspectiva2 y por Perspectiva3".

Figura 5.20: Caso 2, preguntas.

Entonces se obtendrá:

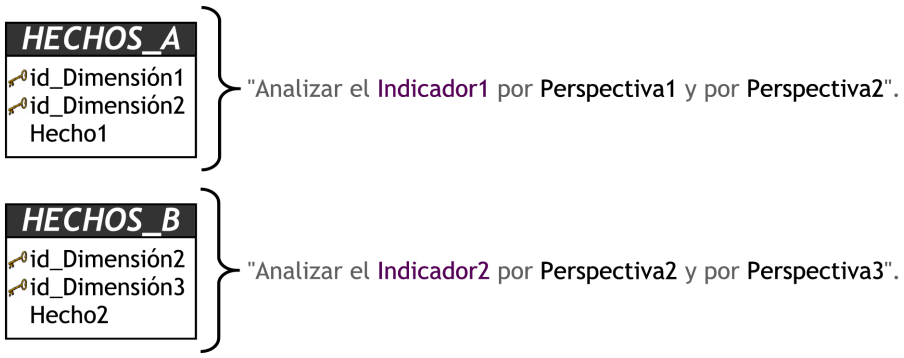


Figura 5.21: Caso 2, diseño de tablas de hechos.

- Caso 3: Si el conjunto de preguntas de negocio cumplen con las condiciones de los dos puntos anteriores se deberán unificar aquellos interrogantes que posean diferentes indicadores pero iguales perspectivas de análisis, para luego reanudar el estudio de las preguntas. Por ejemplo:

"Analizar el Indicador1 por Perspectiva1 y por Perspectiva2".
"Analizar el Indicador2 por Perspectiva1 y por Perspectiva2".

Figura 5.22: Caso 3, preguntas.

Se unificarán en:

"Analizar el Indicador1 y el Indicador2 por Perspectiva1 y por Perspectiva2".

Figura 5.23: Caso 3, unificación.

Caso práctico:
A continuación, se confeccionará la tabla de hechos:

- La tabla de hechos tendrá el nombre “VENTAS”.
- Su clave principal será la combinación de las claves principales de las tablas de dimensiones antes definidas: “idCliente”, “idProducto” e “idFecha”.

- Se crearán dos hechos, que se corresponden con los dos indicadores y serán renombrados, “Unidades Vendidas” por “Cantidad” y “Monto Total de Ventas” por “MontoTotal”.

En el gráfico siguiente se puede apreciar mejor este paso:

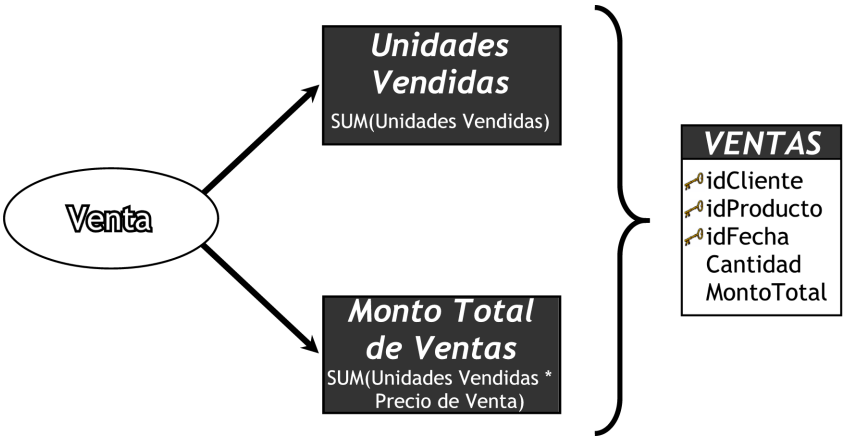


Figura 5.24: Caso práctico, diseño de la tabla de hechos.

5.5.3.4. d) Uniones

Para los tres tipos de esquemas, se realizarán las uniones correspondientes entre sus tablas de dimensiones y sus tablas de hechos.

Caso práctico:
Se realizarán las uniones pertinentes, de acuerdo corresponda:

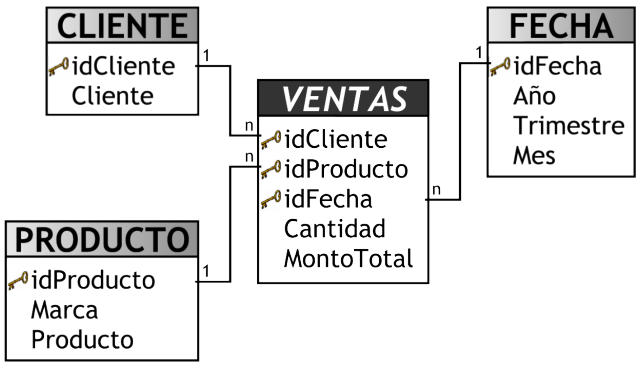


Figura 5.25: Caso práctico, uniones.

5.5.4. PASO 4) INTEGRACIÓN DE DATOS

Una vez construido el modelo lógico, se deberá proceder a poblarlo con datos, utilizando técnicas de limpieza y calidad de datos, procesos ETL, etc.; luego se definirán las reglas y políticas para su respectiva actualización, así como también los procesos que la llevarán a cabo.

5.5.4.1. a) Carga Inicial

Debemos en este paso realizar la Carga Inicial[4] al DW, poblando el modelo de datos que hemos construido anteriormente. Para lo cual debemos llevar adelante una serie de tareas básicas, tales como limpieza de datos, calidad de datos, procesos ETL, etc.

La realización de estas tareas pueden contener una lógica realmente compleja en algunos casos. Afortunadamente, en la actualidad existen muchos softwares que se pueden emplear a tal fin, y que nos facilitarán el trabajo.

Se debe evitar que el DW sea cargado con valores faltantes o anómalos, así como también se deben establecer condiciones y restricciones para asegurar que solo se utilicen los datos de interés.

Cuando se trabaja con un esquema constelación, hay que tener presente que varias tablas de dimensiones serán compartidas con diferentes tablas de hechos, ya que puede darse el caso de que algunas restricciones aplicadas sobre una tabla de dimensión en particular para analizar una tabla de hechos, se puedan contraponer con otras restricciones o condiciones de análisis de otras tablas de hechos.

Primero se cargarán los datos de las dimensiones y luego los de las tablas de hechos, teniendo en cuenta siempre, la correcta correspondencia entre cada elemento. En el caso en que se esté utilizando un esquema copo de nieve, cada vez que existan jerarquías de dimensiones, se comenzarán cargando las tablas de dimensiones del nivel más general al más detallado.

Concretamente, en este paso se deberá registrar en detalle las acciones llevadas a cabo con los diferentes softwares. Por ejemplo, es muy común que sistemas ETL trabajen con "pasos" y "relaciones", en donde cada "paso" realiza una tarea en particular del proceso ETL y cada "relación" indica hacia donde debe dirigirse el flujo de datos. En este caso lo que se debe hacer es explicar que hace el proceso en general y luego que hace cada "paso" y/o "relación". Es decir, se partirá de lo más general y se irá a lo más específico, para obtener de esta manera una visión general y detallada de todo el proceso.

Es importante tener presente, que al cargar los datos en las tablas de hechos pueden utilizarse preagregaciones[5], ya sea al nivel de granularidad de la misma o a otros niveles diferentes.

Caso práctico:

Para simplificar la aplicación del ejemplo, el caso práctico solo se centrará en los aspectos más importantes del proceso ETL, obviando entrar en detalle de cómo se realizan

[4] Ver sección 3.3.3, en la página 25.
[5] Ver sección 3.4.3.1, en la página 32.

algunas funciones y/o pasos.

El proceso ETL planteado para la Carga Inicial es el siguiente:

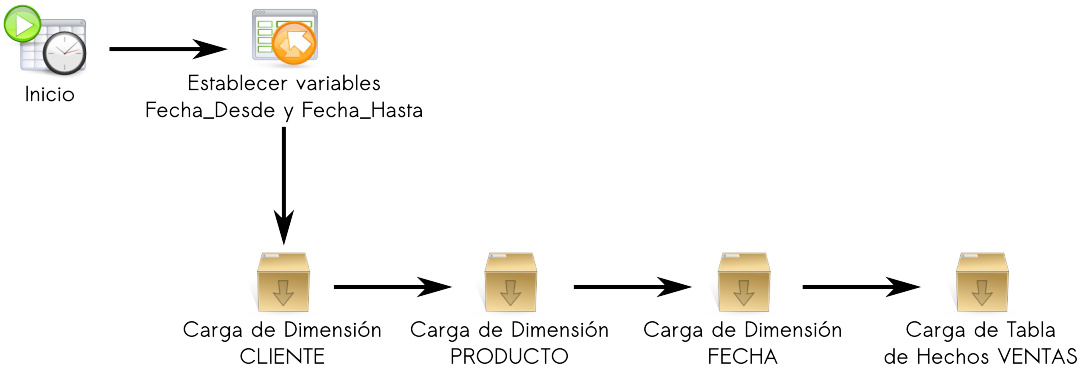


Figura 5.26: Caso práctico, Carga Inicial.

Las tareas que lleva a cabo este proceso son:

- Inicio: inicia la ejecución de los pasos en el momento en que se le indique.
- Establecer variables Fecha_Desde y Fecha_Hasta: establece dos variables globales que serán utilizadas posteriormente por algunos pasos.
  - Para la variable "Fecha_Desde" se obtiene el valor de la fecha en que se realizó la primera venta.
  - Para la variable "Fecha_Hasta" se obtiene el valor de la fecha actual.
- Carga de Dimensión CLIENTE: ejecuta el contenedor de pasos que cargará la dimensión CLIENTE, más adelante se detallará el mismo.
- Carga de Dimensión PRODUCTO: ejecuta el contenedor de pasos que cargará la dimensión PRODUCTO, más adelante se detallará el mismo.
- Carga de Dimensión FECHA: ejecuta el contenedor de pasos que cargará la dimensión FECHA, más adelante se detallará el mismo.
- Carga de Tabla de Hechos VENTAS: ejecuta el contenedor de pasos que cargará la tabla de hechos VENTAS, más adelante se detallará el mismo.

A continuación, se especificarán las tareas llevadas a cabo por "Carga de Dimensión CLIENTE". Este paso es un contenedor de pasos, así que incluye las siguientes tareas:

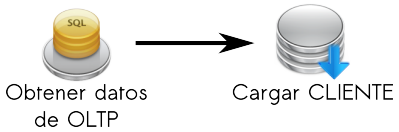


Figura 5.27: Caso práctico, Carga de Dimensión CLIENTE.

- Obtener datos de OLTP: obtiene a través de una consulta SQL los datos del OLTP necesarios para cargar la dimensión CLIENTE.

Se tomará como fuente de entrada la tabla “Clientes” del OLTP mencionado anteriormente.

Se consultó con l@s usuari@s y se averiguó que deseaban tener en cuenta solo aquellos clientes que no estén eliminados y que tengan su cuenta habilitada.

Es importante destacar que aunque existían numerosos movimientos de clientes que en la actualidad no poseen su cuenta habilitada o que figuran como eliminados, se decidió no incluirlos debido a que el énfasis está puesto en analizar los datos a través de aquellos clientes que no cuentan con estas condiciones.

Los clientes eliminados son referenciados mediante el campo “Eliminado”, en el cual un valor “1” indica que este fue eliminado, y un valor “0” que aún permanece vigente. Cuando se examinaron los registros de la tabla, para muchos clientes no había ningún valor asignado para este campo, lo cual, según comunicó el encargado del sistema, se debía a que este se agregó poco después de haberse creado la base de datos inicial, razón por la cual existían valores faltantes. Además, comentó que en el sistema, si un cliente posee en el campo “Eliminado” un valor “0” o un valor faltante, es considerado como vigente.

Con respecto a la cuenta habilitada, el campo del OLTP que le hace mención es “Cta_Habilitada”, y un valor “0” indica que no está habilitada y un valor “1” que sí.

Seguidamente, se expondrá la sentencia SQL que contiene este paso:

```
SELECT
        Clientes.id_Cliente AS idCliente,
        Clientes.Razon_Soc AS Cliente
FROM
        Clientes
WHERE
        (Clientes.Eliminado <> 1)
AND     (Clientes.Cta_Habilitada <> 0)
ORDER BY
        Clientes.id_Cliente,
        Clientes.Razon_Soc
```

Figura 5.28: Caso práctico, CLIENTE - Obtener datos de OLTP.

- Cargar CLIENTE: almacena en la tabla de dimensión CLIENTE los datos obtenidos en el paso anterior.

A continuación, se especificará las tareas llevadas a cabo por "Carga de Dimensión PRODUCTO". Este paso es un contenedor de pasos, así que incluye las siguientes tareas:

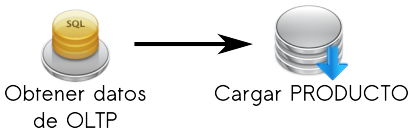


Figura 5.29: Caso práctico, Carga de Dimensión PRODUCTO.

- Obtener datos de OLTP: obtiene a través de una consulta SQL los datos del OLTP necesarios para cargar la dimensión PRODUCTO.

  Las fuentes que se utilizarán, son las tablas “Productos” y “Marcas”.

  En este caso, aunque existían productos eliminados, l@s usuari@s decidieron que esta condición no fuese tomada en cuenta, ya que habían movimientos que hacían referencia a productos con este estado.

  Es necesario realizar una unión entre la tabla “Productos” y “Marcas”, por lo cual se debió asegurar que ningún producto hiciera mención a alguna marca que no existiese, y se tomaron medidas contra su futura aparición.

  El SQL que contiene este paso es el siguiente:

```
SELECT
        Productos.id_prod AS idProducto,
        Marcas.Nombre AS Marca,
        Productos.Detalle AS Producto
FROM
        Productos LEFT OUTER JOIN
        Marcas ON Productos.id_Marca = Marcas.id_Marca
ORDER BY
        Productos.id_prod,
        Marcas.Nombre,
        Productos.Detalle
```

Figura 5.30: Caso práctico, PRODUCTO - Obtener datos de OLTP.

- Cargar PRODUCTO: almacena en la tabla de dimensión PRODUCTO los datos obtenidos en el paso anterior.

A continuación, se especificarán las tareas llevadas a cabo por "Carga de Dimensión FECHA". Este paso es un contenedor de pasos, así que incluye las siguientes tareas:

Figura 5.31: Caso práctico, Carga de Dimensión FECHA.

Para generar esta tabla de dimensión, infaltable en todo DW, existen varias herramientas y utilidades de software que proporcionan diversas opciones para su confección. Pero, si no se cuenta con ninguna, se puede realizar manualmente o mediante algún programa, llenando los datos en un archivo, tabla, hoja de cálculo, etc, y luego exportándolos a donde se requiera.

Lo que se hizo, fue realizar un procedimiento que hace lo siguiente:

- Recibe como parámetros los valores de "Fecha_Desde" y "Fecha_Hasta".
- Recorre una a una las fechas que se encuentran dentro de este intervalo.
- Analiza cada fecha y realiza una serie de operaciones para crear los valores de los campos de la tabla de la dimensión FECHA:

```
"idFecha";"Año";"Trimestre";"Mes"
20000101;2000;"1er Tri";"Enero"
20000102;2000;"1er Tri";"Enero"
20000103;2000;"1er Tri";"Enero"
20000104;2000;"1er Tri";"Enero"
20000105;2000;"1er Tri";"Enero"
20000106;2000;"1er Tri";"Enero"
20000107;2000;"1er Tri";"Enero"
20000108;2000;"1er Tri";"Enero"
20000109;2000;"1er Tri";"Enero"
20000110;2000;"1er Tri";"Enero"
```

Figura 5.32: Caso práctico, datos de FECHA.

  - idFecha = YEAR(fecha)*10000 + MONTH(fecha)*100 + DAY(fecha).
  - Año = YEAR(fecha).
  - Trimestre = CASE WHEN QUARTER(fecha) = 1 then '1er Tri' ... END.
  - Mes = CASE WHEN MONTH(fecha) = 1 then 'Enero' ... END.
- Inserta los valores obtenidos en la tabla de dimensión FECHA.

Como puede observarse, la clave principal "idFecha" es un campo numérico representado por el formato "yyyymmdd".

A continuación, se especificará las tareas llevadas a cabo por "Carga de Tabla de Hechos VENTAS". Este paso es un contenedor de pasos, así que incluye las siguientes tareas:

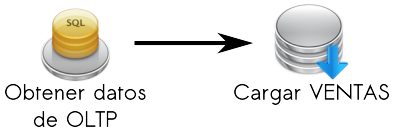


Figura 5.33: Caso práctico, Carga de Tabla de Hechos VENTAS.

- Obtener datos de OLTP: obtiene a través de una consulta SQL los datos del OLTP necesarios para cargar la tabla de hechos VENTAS.

Para la confección de la tabla de hechos, se tomaron como fuente las tablas "Facturas_Ventas" y "Detalles_Venta". Al igual que en las tablas de dimensiones, se recolectaron las condiciones que deben cumplir los datos para considerarse de interés, y en este caso, se trabajará solamente con aquellas facturas que no hayan sido anuladas.

Se investigó al respecto, y se llegó a la conclusión de que el campo que da dicha información en "Anulada" de la tabla "Facturas_Ventas" y si el mismo posee el valor "1" significa que efectivamente fue anulada.

Otro punto importante a tener en cuenta es que la fecha se debe convertir al formato numérico "yyyymmdd".

Se decidió aplicar una preagregación a los hechos que formarán parte de la tabla de hechos, es por esta razón que se utilizará la cláusula GROUP BY para agrupar todos los registros a través de las claves primarias de esta tabla.

La sentencia SQL que contiene este paso fue la siguiente:

```
SELECT
        Facturas_Venta.id_Cliente AS idCliente,
        Detalles_Venta.id_prod AS idProducto,
        ((YEAR(Facturas_Venta.Fecha) * 10000) + (MONTH(Facturas_Venta.Fecha) * 100) +
                (DAY(Facturas_Venta.Fecha))) AS idFecha,
        SUM(Detalles_Venta.cantidad) AS Cantidad,
        SUM(Detalles_Venta.cantidad * Detalles_Venta.precio_Fact) AS MontoTotal
FROM
        Facturas_Venta INNER JOIN
        Detalles_Venta ON Facturas_Venta.id_Fact = Detalles_Venta.id_fact
WHERE
        (Facturas_Venta.Anulada <> 1)
GROUP BY
        Facturas_Venta.id_Cliente,
        Detalles_Venta.id_prod,
        Facturas_Venta.Fecha
ORDER BY
        Facturas_Venta.id_Cliente,
        Detalles_Venta.id_prod,
        idFecha,
        Cantidad,
        MontoTotal
```

Figura 5.34: Caso práctico, VENTAS - Obtener datos de OLTP

- Cargar VENTAS: almacena en la tabla de hechos VENTAS los datos obtenidos en el paso anterior.

5.5.4.2. b) Actualización

Cuando se haya cargado en su totalidad el DW, se deben establecer sus políticas y estrategias de actualización o refresco de datos.

Una vez realizado esto, se tendrán que llevar a cabo las siguientes acciones:

- Especificar las tareas de limpieza de datos, calidad de datos, procesos ETL, etc., que deberán realizarse para actualizar los datos del DW.
- Especificar de forma general y detallada las acciones que deberá realizar cada software.

Caso práctico:

Las políticas de Actualización que se han convenido con l@s usuari@s son las siguientes:

- La información se refrescará todos los días a las doce de la noche.
- Los datos de las tablas de dimensiones "PRODUCTO" y "CLIENTE" serán cargados totalmente cada vez.
- Los datos de la tabla de dimensión "FECHA" se cargarán de manera incremental teniendo en cuenta la fecha de la última actualización.
- Los datos de la tabla de hechos que corresponden al último mes (30 días) a partir de la fecha actual, serán reemplazados cada vez.
- Estas acciones se realizarán durante un periodo de prueba, para analizar cuál es la manera más eficiente de generar las actualizaciones, basadas en el estudio de los cambios que se producen en los OLTP y que afectan al contenido del DW.

Para evitar que se extienda demasiado la aplicación del ejemplo, el caso práctico solo incluirá lo que debería realizar el proceso ETL para actualizar el DW.

El proceso ETL para la actualización del DW es muy similar al de Carga Inicial, pero cuenta con las siguientes diferencias:

- Inicio: iniciará la ejecución de los pasos todos los días a las doce de la noche.
- Establecer variables Fecha_Desde y Fecha_Hasta:
  - La variable "Fecha_Desde" obtendrá el valor resultante de restarle a la fecha actual treinta días.
  - La variable "Fecha_Hasta" obtendrá el valor de la fecha actual.
- Carga de Dimensión CLIENTE: a la serie de tareas que realiza este paso, se le antecederá un nuevo paso que borrará los datos de la dimensión CLIENTE.
- Carga de Dimensión PRODUCTO: a la serie de tareas que realiza este paso, se le antecederá un nuevo paso que borrará los datos de la dimensión PRODUCTO.
- Carga de Dimensión FECHA: en este paso, en vez de recibir el valor de la variable "Fecha_Desde", se tomará la fecha del último registro cargado en la dimensión FECHA.
- Carga de Tabla de Hechos VENTAS:
  - a la serie de tareas que realiza este paso, se le antecederá un nuevo paso que borrará los datos de la tabla de HECHOS correspondientes al intervalo entre "Fecha_Desde" y "Fecha_Hasta".
  - en el paso "Obtener datos de OLTP" se le agregará a la sentencia SQL la siguiente condición:
    - WHERE Facturas_Venta.Fecha >= {Fecha_Desde} AND Facturas_Venta.Fecha <= {Fecha_Hasta}

5.6. Creación de Cubos Multidimensionales

A continuación se creará un cubo multidimensional de ejemplo, que será llamado ”Cubo de Ventas” y que estará basado en el modelo lógico diseñado en el caso práctico de la metodología Hefesto:

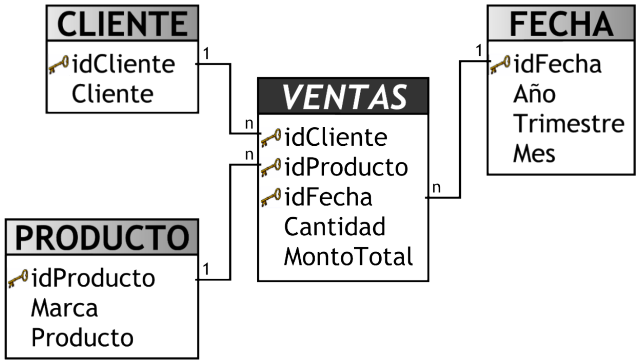


Figura 5.35: Caso práctico, modelo lógico.

La creación de este cubo tiene las siguientes finalidades:

- Ejemplificar la creación de cubos multidimensionales.
- Propiciar la correcta distinción entre hechos de una tabla de hechos e indicadores de un cubo.
- Propiciar la correcta distinción entre campos de una tabla de dimensión y atributos de un cubo.

5.6.1. Creación de Indicadores

En este momento se crearán dos indicadores que serán incluidos en el cubo ”Cubo de Ventas”:

- De la tabla de hechos “VENTAS”, se sumarizará el hecho “Cantidad” para crear el indicador denominado:
  - “Unidades Vendidas”.

  La fórmula utilizada para crear este indicador es la siguiente:
  - “Unidades Vendidas” = SUM(VENTAS.Cantidad).
- De la tabla de hechos “VENTAS”, se sumarizará el hecho “MontoTotal” para crear el indicador denominado:
  - “Monto Total de Ventas”.

  La fórmula utilizada para crear este indicador es la siguiente:
  - “Monto Total de Ventas” = SUM(VENTAS.MontoTotal).

Entonces, el cubo quedaría conformado de la siguiente manera:

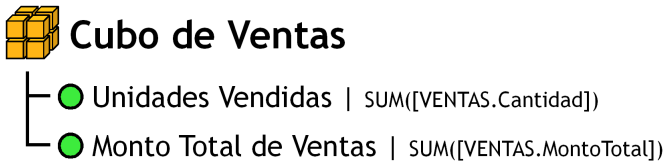


Figura 5.36: Cubo ejemplo, paso 1.

5.6.2. Creación de Atributos

Ahora se crearán y agregarán al cubo seis atributos:

- De la tabla de dimensión “CLIENTE”, se tomará el campo “Cliente” para la creación del atributo denominado:
  - “Clientes”.
- De la tabla de dimensión “PRODUCTO”, se tomará el campo “Marca” para la creación del atributo denominado:
  - “Marcas”.
- De la tabla de dimensión “PRODUCTO”, se tomará el campo “Producto” para la creación del atributo denominado:
  - “Productos”.
- De la tabla de dimensión “FECHA”, se tomará el campo “Año” para la creación del atributo denominado:
  - “Años”.
- De la tabla de dimensión “FECHA”, se tomará el campo “Trimestre” para la creación del atributo denominado:
  - “Trimestres”.
- De la tabla de dimensión “FECHA”, se tomará el campo “Mes” para la creación del atributo denominado:
  - “Meses”.

Entonces, el cubo quedaría conformado de la siguiente manera:

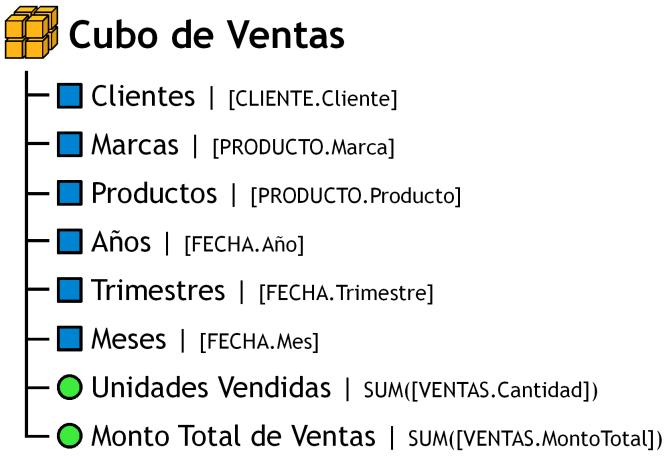


Figura 5.37: Cubo ejemplo, paso 2.

5.6.3. Creación de Jerarquías

Finalmente se crearán y agregarán al cubo dos jerarquías:

- Se definió la jerarquía “Jerarquía Productos”, que se aplicará sobre los atributos recientemente creados, “Marcas” y “Productos”, en donde:
  - Un producto en especial pertenece solo a una marca. Una marca puede tener uno o más productos.

  Gráficamente:

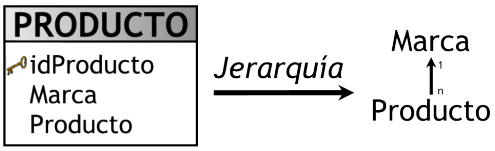


Figura 5.38: “PRODUCTO”, relación padre-hijo.

- Se definió la jerarquía “Jerarquía Fechas”, que se aplicará sobre los atributos recientemente creados, “Años”, “Trimestres” y “Meses”, en donde:
  - Un mes del año pertenece solo a un trimestre del año. Un trimestre del año tiene uno o más meses del año.
  - Un trimestre del año pertenece solo a un año. Un año tiene uno o más trimestres del año.

Gráficamente:

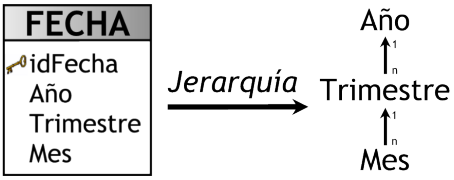


Figura 5.39: "FECHA", relación padre-hijo.

Entonces, el cubo quedaría conformado de la siguiente manera:

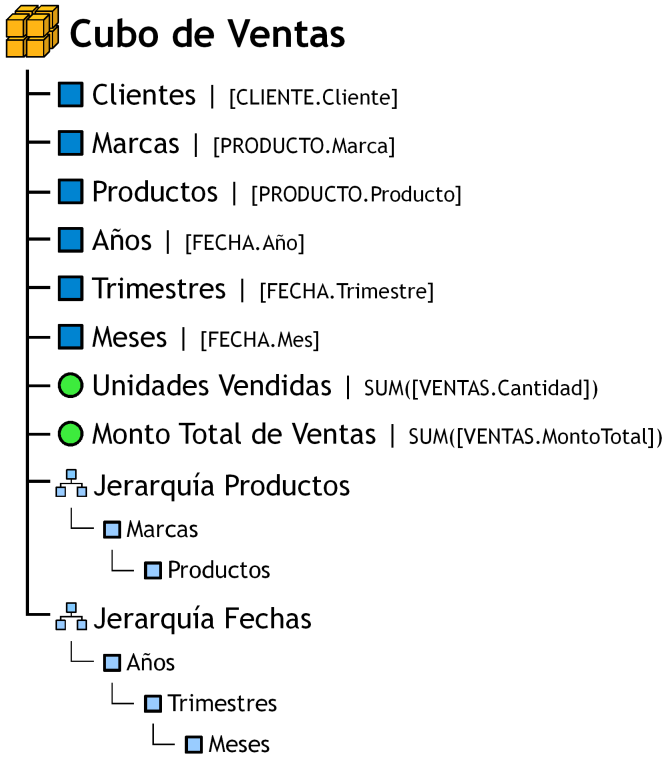


Figura 5.40: Cubo ejemplo, paso 3.

5.6.4. Otros ejemplos de cubos multidimensionales

A partir del modelo lógico planteado, podrían haberse creado una gran cantidad de cubos, cada uno de los cuales estaría orientado a un tipo de análisis en particular. Tal y como se explicó antes, estos cubos pueden coexistir sin ningún inconveniente.

A continuación se expondrán una serie de cubos de ejemplo:

- Cubo 1:

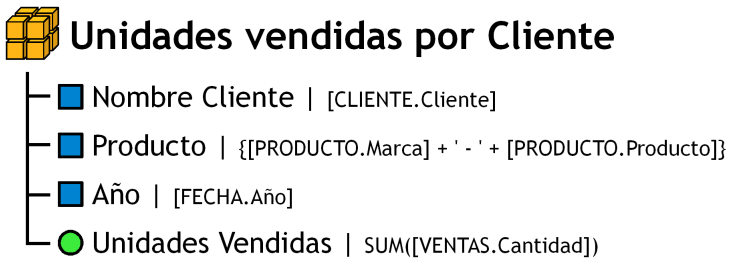


Figura 5.41: Cubo 1, ejemplo.

- Cubo 2:

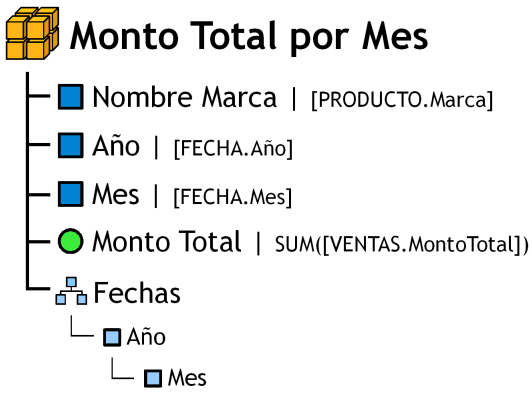


Figura 5.42: Cubo 2, ejemplo.

- Cubo 3:

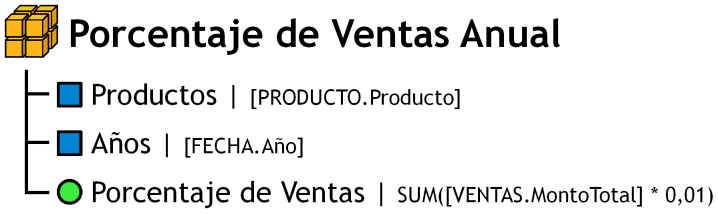


Figura 5.43: Cubo 3, ejemplo.

Capítulo 6

CONSIDERACIONES DE DISEÑO

6.1. Tamaño del DW

Dependiendo del negocio, el volumen de datos y el alcance del proyecto, el tamaño del DW puede variar considerablemente, por lo cual, es una buena práctica tener esto en cuenta al momento de diseñar el depósito y al determinar los recursos físicos, los tiempos de desarrollo y los respectivos costos inherentes.

De acuerdo al tamaño del depósito de datos, se lo puede clasificar como:

- Personal: si su tamaño es menor a 1 Gigabyte.

$$DW < 1\ GB$$

- Pequeño: si su tamaño es mayor a 1 Gigabyte y menor a 50 Gigabyte.

$$1\ GB < DW < 50\ GB$$

- Mediano: si su tamaño es mayor a 50 Gigabyte y menor a 100 Gigabyte.

$$50\ GB < DW < 100\ GB$$

- Grande: si su tamaño es mayor a 100 Gigabyte y menor a 1 Terabyte.

$$100\ GB < DW < 1\ TB$$

- Muy grande: si su tamaño es mayor a 1 Terabyte.

$$DW > 1\ TB$$

6.2. Tiempo de construcción

Divers@s autor@s resaltan la importancia del factor tiempo en la construcción de un DW, por lo cual se ha considerado interesante exponer tres frases seleccionadas al respecto:

- “El 70 % del tiempo total dedicado al proyecto se insume en definir el problema y en preparar la tabla de datos”.
- “Estime el tiempo necesario, multiplíquelo por dos y agregue una semana de resguardo”.
- ”Regla 90 – 90”: el primer 90 % de la construcción de un sistema absorbe el 90 % del tiempo y esfuerzo asignados; el último 10 % se lleva el otro 90 % del tiempo y esfuerzo asignado.

6.3. Implementación

Las implementaciones de los depósitos de datos varían entre sí de forma considerable, teniendo en cuenta las herramientas de software que se empleen, los modelos que se utilicen, recursos disponibles, SGBD que lo soporten, herramientas de análisis y consulta, entre otros.

6.4. Performance

Cuando se diseñan los ETLs, es muy importante que los mismos sean lo más eficientes posible, ya que una vez que se tenga un gran volumen de datos, el espacio en disco se volverá fundamental y los tiempos incurridos en el procesamiento y acceso a la información serán esenciales, y más aún si el DWH es considerado o tomado como un sistema de misión crítica.

También es muy importante configurar correctamente el SGBD en el que se almacene y mantenga el DW, así como lo es elegir las mejores estrategias para modelar las diferentes estructuras de datos que se utilizarán.

Para mejorar la performance del DWH, se pueden llevar a cabo las siguientes acciones sobre el DW y las estructuras de datos (cubos multidimensionales, Business Models, etc):

- Prestar especial atención a los tipos de datos utilizados, por ejemplo, para valores enteros pequeños conviene utilizar tinyint o smallint en lugar de int, con el fin de no asignar tamaños de datos mayores a los necesarios. Esto toma vital importancia cuando se aplica en las claves primarias, debido a que formarán parte de la tabla de hechos que es la que contiene el volumen del almacén de datos.
- Utilizar Claves Subrogadas[1].
- Utilizar técnicas de indexación.
- Utilizar técnicas de particionamiento.
- Crear diferentes niveles de sumarización.
- Crear vistas materializadas.

[1] Ver sección 6.13, en la página 124.

- Utilizar técnicas de administración de datos en memoria caché.
- Utilizar técnicas de multiprocesamiento, con el objetivo de agilizar la obtención de resultados, a través de la realización de procesos en forma concurrente.

6.5. Mantenimiento

Un punto muy importante es mantener en correcto funcionamiento al DW, ya que a medida que pase el tiempo, este tenderá a crecer significativamente, y surgirán cambios, tanto en los requerimientos como en las fuentes de datos.

6.6. Impactos

Al implementar un DWH, es fundamental que l@s usuari@s del mismo participen activamente durante todo su desarrollo, debido a que son ell@s l@s que conocen en profundidad su negocio y saben cuáles son los resultados que se desean obtener. Además, es precisamente en base a la utilización que se le de, que el depósito de datos madurará y se adaptará a las situaciones cambiantes por las que atraviese la empresa. L@s usuari@s, al trabajar junto a l@s desarrollador@s y analistas podrán comprender más en profundidad sus propios sistemas operacionales, con todo lo que esto implica.

Con la implementación del DWH, los procesos de toma de decisiones serán optimizados, al obtener información correcta al instante en que se necesita, evitando perdidas de tiempo y anomalías en los datos. Al contar con esta información, l@s usuari@s tendrán más confianza en las decisiones que tomarán y en adición a ello, poseerán una base sustentable para justificarlas.

Usualmente, los DW integrarán fuentes de datos de diversas áreas y sectores de la empresa, esto tendrá como beneficio contar con una sola fuente de información centralizada y común para tod@s l@s usuari@s. Esto posibilitará que en las diferentes áreas se compartan los mismos datos, lo cual conducirá a un mayor entendimiento, comunicación, confianza y cooperación entre las mismas.

El DWH introducirá nuevos conceptos tecnológicos y de inteligencia de negocios, lo cual requerirá que se aprendan nuevas técnicas, herramientas, métodos, destrezas, formas de trabajar, etc.

6.7. DM como sub proyectos

Al diseñar e implementar DM como partes de un proyecto DW, se debe tener en cuenta que el análisis que se efectuará, los modelos que intervendrán y el alcance, deben ser globales, con el fin de determinar, por ejemplo, tablas de dimensiones comunes entre las diferentes áreas de trabajo. Esto evitará que se realicen tareas repetidas, ahorrando tiempos y enfocándose en la consolidación, unificación y centralización de la información de los diferentes sectores.

6.8. Teoría de grafos

Para evaluar la validez de la estructura lógica del depósito de datos, puede emplearse la teoría de grafos, la cual afirma que su estructura será correcta sí y solo sí está confor-

mada únicamente por trayectorias acíclicas.

Si se encuentran trayectorias cíclicas, deberán ser transformadas para que las consultas al DW sean válidas y confiables.

Una trayectoria acíclica, es aquella que sólo tiene una forma de recorrido (en un solo sentido). Por ejemplo, en la siguiente figura se puede apreciar que existe una sola manera de recorrer las tablas de dimensiones.

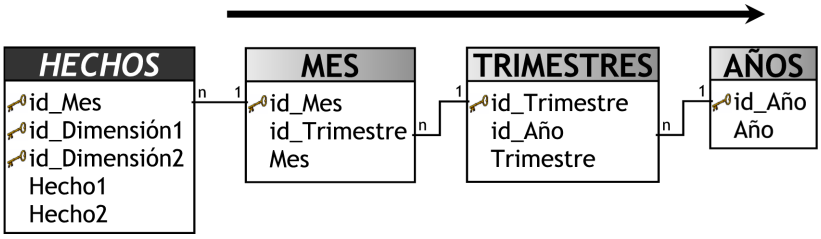


Figura 6.1: Trayectoria acíclica.

Una trayectoria cíclica, es aquella que se puede recorrer en dos o más secuencias diferentes. Por ejemplo, en la siguiente imagen se pueden distinguir dos sentidos por los cuales recorrer las tablas de dimensiones.

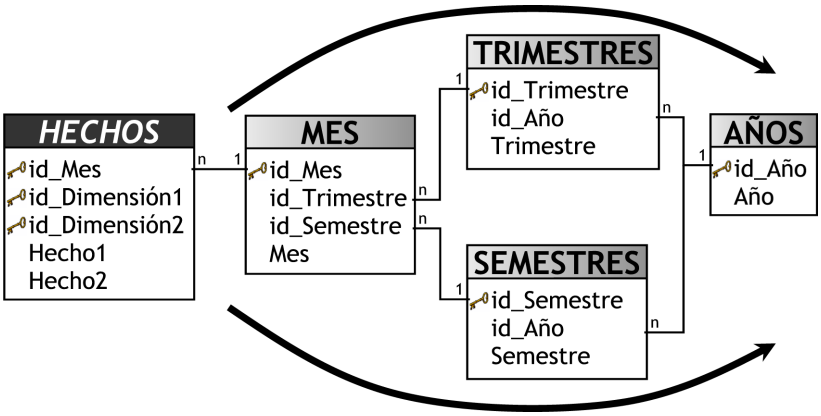


Figura 6.2: Trayectoria cíclica.

6.9. Elección de columnas

Cuando se seleccionan los campos que integrarán el DW, se debe tener en cuenta lo siguiente:

- Se deben descartar aquellos campos cuyos valores tengan muy poca variabilidad.
- Se deben descartar los campos que tengan valores diferentes para cada objeto, por ejemplo el número de D.N.I. cuando se analizan personas.

- En los casos en que no existan jerarquías dentro de alguna tabla de dimensión, en la cual la cantidad de registros que posee la misma son demasiados, es conveniente, conjuntamente con l@s usuari@s, definirlas. Pero, si llegase a suceder que no se encontrase ningún criterio por el cual jerarquizar los campos, es una buena práctica crear jerarquías propias. El objetivo de llevar a cabo esta acción, es la de poder dividir los registros en grupos, propiciando de esta manera una exploración más amena y controlable. Para ejemplificar este punto, se utilizará como referencia la tabla de dimensión de la siguiente figura. La misma no posee ninguna jerarquía definida y la cantidad de registros con que cuenta son cientos:

Figura 6.3: Tabla de dimensión ”PRODUCTO”.

Entonces, lo que se realizará será crear una nueva jerarquía a partir de los campos disponibles:

  - Se añadirá a la tabla un nuevo campo (“Letra”), el mismo estará formado por la primera letra del atributo “Producto” que lo acompaña. Por ejemplo, si el valor de “Producto” es “Lapicera”, “Letra” será “L”; si es “Cartuchera” será “C”, etc.

El resultado será el siguiente:

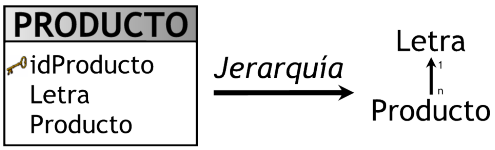


Figura 6.4: Jerarquía de ”PRODUCTO”.

Además, se pueden aplicar algunas de las acciones que se expondrán a continuación sobre los valores de los campos que se incluirán en el depósito de datos:

- Factorizar: se utiliza para descomponer un valor en dos o más componentes. Por ejemplo, el campo “código” perteneciente a un producto está formado por tres identificadores separados por guiones medios, que representan su rubro, marca y tipo (“idRubro-idMarca-idTipo”), entonces este campo puede factorizarse y separarse en tres valores independientes (“idRubro”, “idMarca” e “idTipo”).

- Estandarizar: se utiliza para ajustar valores a un tipo de formato o norma preestablecida. Por ejemplo, se puede emplear esté método cuando se desea que todos lo campos del tipo texto sean convertidos a mayúscula.

- Codificar: es utilizado para representar valores a través de las reglas de un código preestablecido. Por ejemplo, en el campo “estado” se pueden codificar sus valores, “0” y “1”, para transformarlos en “Apagado” y “Encendido” respectivamente.

- Discretizar: es empleado para convertir un conjunto continuo de valores en uno discreto. Por ejemplo, cuando se especificaron los tamaños del DW se realizó está operación.

6.10. Claves primarias en tablas de Dimensiones

Al momento de añadir la clave principal a una tabla de dimensión, se puede establecer:

1. Una única columna que sea clave primaria e identifique unívocamente cada registro.
2. Varias columnas que sean clave primaria e identifiquen en conjunto, unívocamente cada registro.

La primera opción requiere menos espacio de almacenamiento en el DW y permite que las consultas SQL sean más sencillas. La segunda opción requiere más espacio de almacenamiento en el DW, provoca que las consultas SQL sean más complejas y por consiguiente hace que se demore más tiempo en procesar los resultados. Sin embargo, esta última alternativa hace que los procesos ETL sean menos complejos y más eficientes.

Más allá de estas dos grandes opciones, es totalmente recomendable la utilización de Claves Subrogadas[2].

6.11. Balance de diseño

El siguiente gráfico muestra los tres puntos más importantes que se deben balancear al momento de diseñar y construir el modelo lógico de datos del DW:

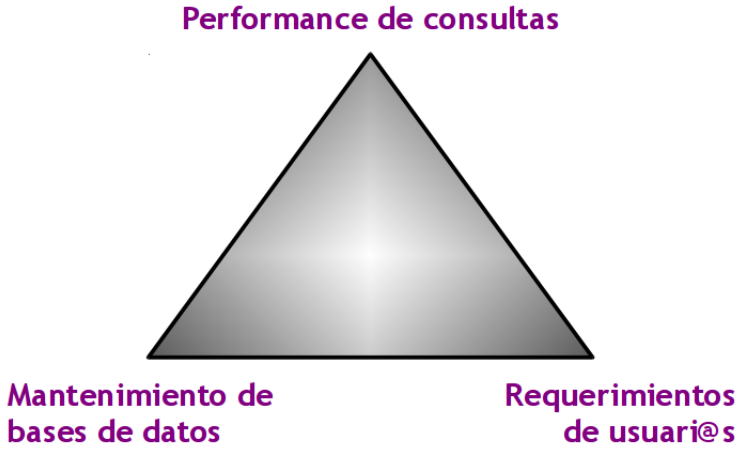


Figura 6.5: Balance de diseño.

Estas tres características están fuertemente relacionadas y condicionadas entre sí, por lo cual, el valor que adopte cada una de ellas, afectará a las otras de manera significativa.

[2]Ver sección 6.13, en la página 124.

Por ejemplo, si se enfoca la atención en los requerimientos de l@s usuari@s, se obtendrá un DW muy complejo que cubrirá todas las necesidades de análisis. Sin embargo, traerá como contrapartida una disminución en la performance de las consultas y un aumento del mantenimiento de las bases de datos.

6.12. Relación muchos a muchos

Siempre que sea posible, se debe evitar mantener en el DW tablas de dimensiones con relaciones muchos a muchos entre ellas, ya que esta situación puede, entre otros inconvenientes, provocar la pérdida de la capacidad analítica de la información y conducir a una sumarización incorrecta de los datos.

Para explicar esta problemática, se tomará como ejemplo la relación existente entre ríos y provincias, es decir:

- Una provincia tiene uno o más ríos, y un río pertenece a una o más provincias.

Además, se tomará como referencia las siguientes tablas pertenecientes a un OLTP, que contienen básicamente los datos relacionados a ríos y provincias:

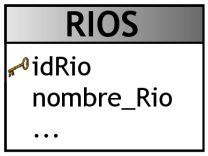


Figura 6.6: Tabla ”RIOS”.

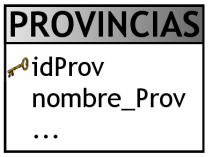


Figura 6.7: Tabla ”PROVINCIAS”.

Cuando existe este tipo de relación (muchos a muchos) entre dos o más tablas, se pueden realizar diferentes acciones para solventar esta situación. Una posible solución, sería llevar a cabo los siguientes pasos:

1. Crear una tabla de dimensión por cada entidad que pertenece a la relación. Cada una de estas tablas no debe incluir ninguna correspondencia a las demás. *En este caso se crearán dos tablas de dimensiones, DIM_RIOS (correspondiente a la entidad “RIOS”) y DIM_PROV (correspondiente a la entidad “PROVINCIAS”).*
2. Crear otra tabla de dimensión *(en este caso DIM_RELACION)*, que sea hija de las tablas de dimensiones recientemente confeccionadas *(en este caso DIM_RIOS y DIM_PROV)*, que estará compuesta de los siguientes campos:
   - Clave principal: dato autonumérico o autoincrementable *(en este caso “id_dim_Relacion”)*.

- Claves foráneas: se deben añadir cada una de las columnas que representan la clave principal de las tablas de dimensiones en cuestión *(en este caso "id_dim_Rio" y "id_dim_Prov")*.
- Otros campos de información adicional.

3. Incluir el campo clave principal creado en el paso anterior *(en este caso "id_dim_Relacion")* en la tabla de hechos.

Gráficamente, el resultado sería el siguiente:

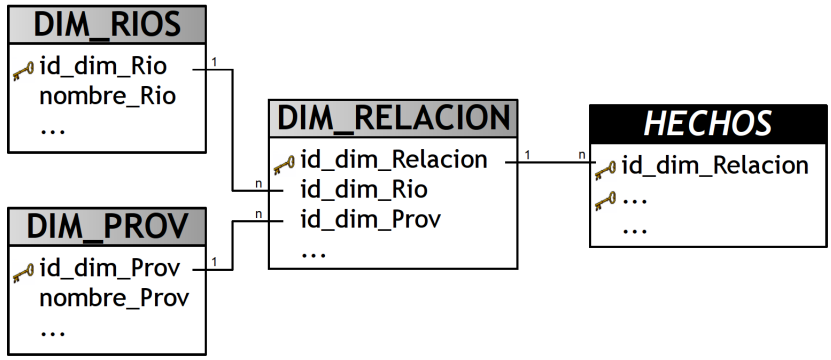


Figura 6.8: Posible solución al modelado de la relación muchos a muchos.

Otra posible solución sería agregar las dos claves primarias de las tablas de dimensiones DIM_RIOS y DIM_PROV en la tabla de hechos.

Existen otras soluciones para solventar esta brecha, pero la primera propuesta posee mucha performance, ya que:

- Elimina la relación muchos a muchos.
- Solo se necesita un campo clave en la tabla de Hechos.
- Las relaciones entre las tablas resultantes es simple y fácil de visualizar.

La única desventaja es en cuanto a los procesos ETL, ya que se aumenta su complejidad y tiempo de proceso.

6.13. Claves Subrogadas

Las claves existentes en los OLTP se denominan claves naturales; en cambio, las claves subrogadas son aquellas que se definen artificialmente, son de tipo numérico secuencial, no tienen relación directa con ningún dato y no poseen ningún significado en especial.

Lo anterior, es solo una de las razones por las cuales utilizar claves subrogadas en el DW, pero se pueden definir una serie de ventajas más:

- Ocupan menos espacio y son más performantes que las tradicionales claves naturales, y más aún si estas últimas son de tipo texto.
- Son de tipo numérico entero (autonumérico o secuencial).

- Permiten que la construcción y mantenimiento de índices sea una tarea sencilla.
- El DW no dependerá de la codificación interna del OLTP.
- Si se modifica el valor de una clave en el OLTP, el DW lo tomará como un nuevo elemento, permitiendo de esta manera, almacenar diferentes versiones del mismo dato.
- Permiten la correcta aplicación de técnicas SCD (Dimensiones lentamente cambiantes)[3].

Esta clave subrogada debe ser el único campo que sea clave principal de cada tabla de dimensión.

Una forma de implementación sería, a través de la utilización de herramientas ETL, mantener una tabla que contenga la clave primaria de la tabla del OLTP y la clave subrogada correspondiente a la dimensión del DW.

En la tabla de dimensión Tiempo, es conveniente hacer una excepción y mantener un formato tal como "yyyymmdd", ya que esto provee dos grandes beneficios:

- Se simplifican los procesos ETL.
- Brinda la posibilidad de realizar particiones de la tabla de hechos a través de ese campo.

6.14. Dimensiones lentamente cambiantes

Las dimensiones lentamente cambiantes o SCD (Slowly Changing Dimensions) son dimensiones en las cuales sus datos tienden a modificarse a través del tiempo, ya sea de forma ocasional o constante, o implique a un solo registro o la tabla completa.

Cuando ocurren estos cambios, se puede optar por seguir alguna de estas dos grandes opciones:

- Registrar el historial de cambios.
- Reemplazar los valores que sean necesarios.

Inicialmente Ralph Kimball planteó tres estrategias a seguir cuando se tratan las SCD: tipo 1, tipo 2 y tipo 3; pero a través de los años la comunidad de personas que se encargaba de modelar bases de datos profundizó las definiciones iniciales e incluyó varios tipos SCD más, por ejemplo: tipo 4 y tipo 6.

A continuación se detallará cada tipo de estrategia SCD:

- SCD Tipo 1: Sobreescribir.
- SCD Tipo 2: Añadir fila.
- SCD Tipo 3: Añadir columna.
- SCD Tipo 4: Tabla de Historia separada.
- SCD Tipo 6: Híbrido.

[3] Ver sección 6.14, en la página 125.

Cabe destacar que existe un SCD Tipo 0, que representa el no tener en cuenta los cambios que pudieran llegar a suceder en los datos de las dimensiones y por consiguiente no tomar medidas al caso.

De acuerdo a la naturaleza del cambio se debe seleccionar qué Tipo SCD se utilizará, en algunos casos resultará conveniente combinar varias técnicas.

Es importante señalar que si bien hay diferentes maneras de implementar cada técnica, es indispensable contar con claves subrogadas en las tablas de dimensiones para aplicar poder aplicar dichas técnicas.

> Al aplicar las diferentes técnicas SCD, en muchos casos se deberá modificar la estructura de la tabla de dimensión con la que se este trabajando, por lo cual estas modificaciones son recomendables hacerlas al momento de modelar la tabla; aunque también puede hacerse una vez que ya se ha modelado y contiene datos, para lo cual al añadir por ejemplo una nueva columna se deberá especificar los valores por defecto que adoptarán los registros de la tabla.
>
> *NOTA: para todos los ejemplos a continuación, "id_Producto" es una clave subrogada que es clave principal de la tabla utilizada.*

6.14.1. SCD Tipo 1: Sobreescribir

Este tipo es el más básico y sencillo de implementar, ya que si bien no guarda los cambios históricos, tampoco requiere ningún modelado especial y no necesita que se añadan nuevos registros a la tabla.

En este caso cuando un registro presente un cambio en alguno de los valores de sus campos, se debe proceder simplemente a actualizar el dato en cuestión, sobreescribiendo el antiguo.

Para ejemplificar este caso, se tomará como referencia la siguiente tabla:

id_Producto	Rubro	Tipo	Producto
1	Rubro 1	Tipo 1	Producto 1

Figura 6.9: SCD Tipo 1: Tabla ejemplo.

Ahora, se supondrá que este producto ha cambiado de Rubro, y ahora a pasado a ser "Rubro 2", entonces se obtendrá lo siguiente:

id_Producto	Rubro	Tipo	Producto
1	**Rubro 2**	Tipo 1	Producto 1

Figura 6.10: SCD Tipo 1: Aplicación.

Usualmente este tipo es utilizado en casos en donde la información histórica no sea importante de mantener, tal como sucede cuando se debe modificar el valor de un registro porque tiene errores de ortografía.

El ejemplo planteado es solo a fines prácticos, ya que con esta técnica, todos los movimientos realizados de "Producto 1", que antes pertenecían al "Rubro 1", ahora pasarán a ser del "Rubro 2", lo cual creará una gran inconsistencia en el DW.

6.14.2. SCD Tipo 2: Añadir fila

Esta estrategia requiere que se agreguen algunas columnas adicionales a la tabla de dimensión, para que almacenen el historial de cambios.

Las columnas que suelen agregarse son:

- FechaInicio: fecha desde que entró en vigencia el registro actual. Por defecto suele utilizarse una fecha muy antigua, ejemplo: "01/01/1000".
- FechaFin: fecha en la cual el registro actual dejó de estar en vigencia. Por defecto suele utilizarse una fecha muy futurista, ejemplo: "01/01/9999".
- Versión: número secuencial que se incrementa cada nuevo cambio. Por defecto suele comenzar en "1".
- Versión actual: especifica si el campo actual es el vigente. Este valor puede ser en caso de ser verdadero: "true" o "1"; y en caso de ser falso: "flase" o "0".

Entonces, cuando ocurra algún cambio en los valores de los registros, se añadirá una nueva fila y se deberá completar los datos referidos al historial de cambios.

Para ejemplificar este caso, se tomará como referencia la siguiente tabla:

id_Producto	Rubro	Tipo	Producto
1	Rubro 1	Tipo 1	Producto 1

Figura 6.11: SCD Tipo 2: Tabla ejemplo.

A continuación se añadirán las columnas que almacenarán el historial:

id_Producto	Rubro	Tipo	Producto	FechaInicio	FechaFin	Version	VersionActual
1	Rubro 1	Tipo 1	Producto 1	**01/01/1000**	**01/01/9999**	**1**	**true**

Figura 6.12: SCD Tipo 2: Agregación de columnas.

Ahora, se supondrá que este producto ha cambiado de Rubro, y ahora a pasado a ser "Rubro 2", entonces se obtendrá lo siguiente:

id_Producto	Rubro	Tipo	Producto	FechaInicio	FechaFin	Version	VersionActual
1	Rubro 1	Tipo 1	Producto 1	01/01/1000	**06/11/2009**	1	**false**
2	**Rubro 2**	Tipo 1	Producto 1	**07/11/2009**	01/01/9999	**2**	**true**

Figura 6.13: SCD Tipo 2: Aplicación.

Como puede observarse, se lleva a cabo el siguiente proceso:

- Se añade una nueva fila con su correspondiente clave subrogada ("id_Producto").
- Se registra la modificación ("Rubro").
- Se actualizan los valores de "FechaInicio" y "FechaFin", tanto de la fila nueva, como la antigua (la que presentó el cambio).
- Se incrementa en uno el valor del campo "Version" que posee la fila antigua.
- Se actualizan los valores de "VersionActual", tanto de la fila nueva, como la antigua; dejando a la fila nueva como el registro vigente (true).

Esta técnica permite guardar ilimitada información de cambios.

6.14.3. SCD Tipo 3: Añadir columna

Esta estrategia requiere que se agregue a la tabla de dimensión una columna adicional por cada columna cuyos valores se desea mantener un historial de cambios.

Para ejemplificar este caso, se tomará como referencia la siguiente tabla:

id_Producto	Rubro	Tipo	Producto
1	Rubro 1	Tipo 1	Producto 1

Figura 6.14: SCD Tipo 3: Tabla ejemplo.

A continuación se añadirá una columna para mantener el histórico de cambios sobre los datos de la columna "Rubro":

id_Producto	Rubro	RubroAnterior	Tipo	Producto
1	Rubro 1	-	Tipo1	Producto 1

Figura 6.15: SCD Tipo 3: Agregación de columna.

Ahora, se supondrá que este producto ha cambiado de Rubro, y ahora a pasado a ser "Rubro 2", entonces se obtendrá lo siguiente:

id_Producto	Rubro	RubroAnterior	Tipo	Producto
1	**Rubro 2**	**Rubro 1**	Tipo 1	Producto 1

Figura 6.16: SCD Tipo 3: Aplicación.

Como puede observarse, se lleva a cabo el siguiente proceso:

- En la columna "RubroAnterior" se coloca el valor antiguo.
- En la columna "Rubro" se coloca el nuevo valor vigente.

Esta técnica permite guardar una limitada información de cambios.

6.14.4. SCD Tipo 4: Tabla de Historia separada

Esta técnica se utiliza en combinación con alguna otra y su función básica es almacenar en una tabla adicional los detalles de cambios históricos realizados en una tabla de dimensión.

Esta tabla histórica indicará por ejemplo que tipo de operación se ha realizado (Insert, Update, Delete), sobre que campo y en que fecha.

El objetivo de mantener esta tabla es el de contar con un detalle de todos los cambios, para luego analizarlos y poder tomar decisiones acerca de cuál técnica SCD podría aplicarse mejor.

Por ejemplo, la siguiente tabla histórica registra los cambios de la tabla de dimensión "Productos", la cual supondremos emplea el SCD Tipo 2:

id_Producto	Rubro_Cambio	Tipo_Cambio	Producto_Cambio	FechaDeCambio
1	Insert	-	-	05/06/2000
2	Insert	Insert	-	25/10/2002
3	-	Insert	-	17/01/2005
4	-	-	Insert	18/12/2009

Figura 6.17: SCD Tipo 4: Aplicación.

Tomando como ejemplo el primer registro de esta tabla, la información allí guardada indica lo siguiente:

- El día "05/06/2000", el registro de la tabla de dimensión "Productos" con "id_Producto" igual a "1" sufrió un cambio de "Rubro", por lo cual se debió insertar ("Insert") una nueva fila con los valores vigentes.

6.14.5. SCD Tipo 6: Híbrido

Esta técnica combina las SCD Tipo 1, 2 y 3.

Se denomina SCD Tipo "6", simplemente porque: *6 = 1 + 2 +3.*

6.15. Dimensiones Degeneradas

El término Dimensión Degenerada, hace referencia a un campo que será utilizado como criterio de análisis y que es almacenado en la tabla de hechos.

Esto sucede cuando un campo que se utilizará como criterio de análisis posee el mismo nivel de granularidad que los datos de la tabla de hechos, y que por lo tanto no se pueden realizar agrupaciones o sumarizaciones a través de este campo. Los "números de orden", "números de ticket", "números de transacción", etc, son algunos ejemplos de dimensiones degeneradas

La inclusión de estos campos en las tablas de hechos, se lleva a cabo para reducir la duplicación y simplificar las consultas.

Se podría plantear la opción de simplemente incluir estos campos en una tabla de dimensión, pero en este caso estaríamos manteniendo una fila de esta dimensión por cada fila en la tabla de hechos, por consiguiente obtendríamos la duplicación de información y complejidad, que precisamente se pretende evitar.

6.16. Dimensiones Clustering

Las dimensiones Clustering, son aquellas que están relacionadas a dos o más dimensiones y que brindan información diferente a cada una de ellas.

Por ejemplo, en el siguiente esquema, se puede apreciar que dos tablas de dimensiones (“CLIENTES” y “PROVEEDORES”) comparten otra en común (“CIUDADES”), además esta última provee diferente información dependiendo de la tabla de dimensión que la consulte, es decir, devuelve el nombre de la ciudad de l@s client@s o bien la de l@s proveedor@s. En este caso y debido a lo dicho anteriormente, la dimensión ”CIUDADES”, es una dimensión Clustering.

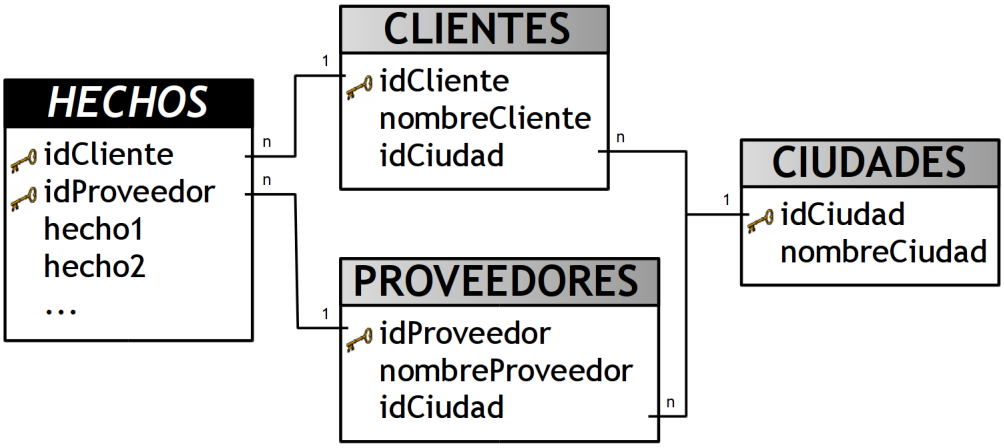


Figura 6.18: Dimensión Clustering: ”CIUDADES”.

Obviamente no se puede mantener este esquema si se pretende analizar los hechos de acuerdo a la ciudad de l@s proveedor@s y de l@s client@s simultáneamente.

Para solucionar esta situación pueden llevarse a cabo diferentes estrategias, cada una de las cuales trae aparejadas sus ventajas y desventajas, por lo cual dependiendo cual sea el contexto se elegirá entre una y otra.

A continuación se destacarán algunas soluciones a esta situación:

1. Se pueden incluir todos los campos de la dimensión Clustering en cada tabla de dimensión con que se relacione y eliminar luego la dimensión Clustering. En este caso:
   - Agregar el campo “nombreCiudad” de la dimensión Clustering “CIUDADES” a la tabla de dimensión ”CLIENTES”.
   - Agregar el campo “nombreCiudad” de la dimensión Clustering “CIUDADES” a la tabla de dimensión ”PROVEEDORES”.

- Eliminar la dimensión Clustering ”CIUDADES”.

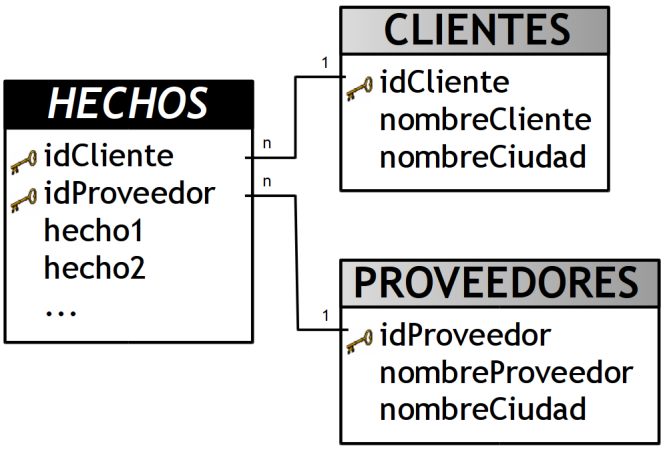


Figura 6.19: Dimensión Clustering: primera posible solución.

- Ventajas: Elimina los JOINs entre las tablas.
- Desventajas: Ante cualquier cambio en los nombres de las ciudades se debe modificar/actualizar todas las dimensiones implicadas.

2. Se puede crear una nueva tabla de dimensión basada en la dimensión Clustering por cada tabla que se relacione con esta y luego eliminar la dimensión Clustering. En este caso:

   - Crear la tabla de dimensión “CIUDADES_CLI”, esta estará basada en la dimensión Clustering ”CIUDADES”.
   - Crear la tabla de dimensión “CIUDADES_PROV”, esta estará basada en la dimensión Clustering ”CIUDADES”.
   - Eliminar la dimensión Clustering ”CIUDADES”.

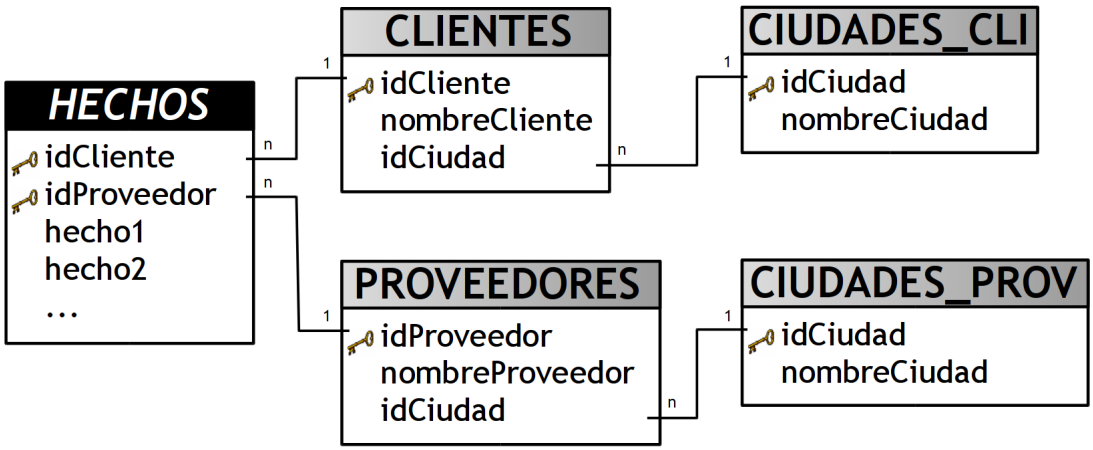


Figura 6.20: Dimensión Clustering: segunda posible solución.

- Ventajas: Ante cualquier cambio en los nombres de las ciudades, solo se deben modificar/actualizar las nuevas dimensiones que están basadas en la dimensión Clustering.
- Desventajas: Mantiene los JOINs entre las tablas.

Apéndice A

Descripción de la empresa

A.1. Identificación de la empresa

La empresa analizada, desarrolla las actividades comerciales de mayorista y minorista de artículos de limpieza, en un ambiente geográfico de alcance nacional. De acuerdo a su volumen de operaciones, se la puede considerar de tamaño mediano.

Con respecto a su clasificación, es una sociedad de responsabilidad limitada con fines de lucro.

Su estructura está formalizada y posee características de una organización funcional.

A.2. Objetivos

Su objetivo principal es el de maximizar sus ganancias. Pero también, se puede adicionar el objetivo de expandirse a un nuevo nivel de mercado, con el fin de conseguir una mayor cantidad de client@s y posicionarse competitivamente por sobre sus rivales.

Otra meta que persigue, pero que aún no está definida como tal, es la de incursionar en otros rubros para lograr diversificarse.

A.3. Políticas

La empresa posee escasos grandes clientes con un gran poder adquisitivo, y son precisamente estos, los que adquieren el volumen de los productos que se comercializan. Debido a ello, la política que se utiliza para cubrir los objetivos antes mencionados, es la de satisfacer ampliamente las necesidades de sus client@s, brindándoles confianza y promoviendo un ambiente familiar entre l@s mism@s. Esta acción se realiza con el fin de mantener l@s client@s actuales y para que nuev@s se interesen en su forma de operar.

Existe otra política que es implícita, por lo cual, no está definida tan estrictamente como la anterior, y es la de mejorar continuamente, con el objetivo de sosegar las exigencias y cambios en el mercado en el que actúa y para conseguir una mejor posición respecto a sus competidor@s.

A.4. Estrategias

Dentro de las estrategias existentes, se han destacado dos por considerarse más significativas, ellas son:

- Expandir el ámbito geográfico, creando varias sucursales en puntos estratégicos del país.
- Añadir nuevos rubros a su actividad de comercialización.

A.5. Organigrama

A continuación, se expondrá un organigrama que fue confeccionado a partir de los datos suministrados en la empresa, ya que no existía ninguno previamente predefinido.

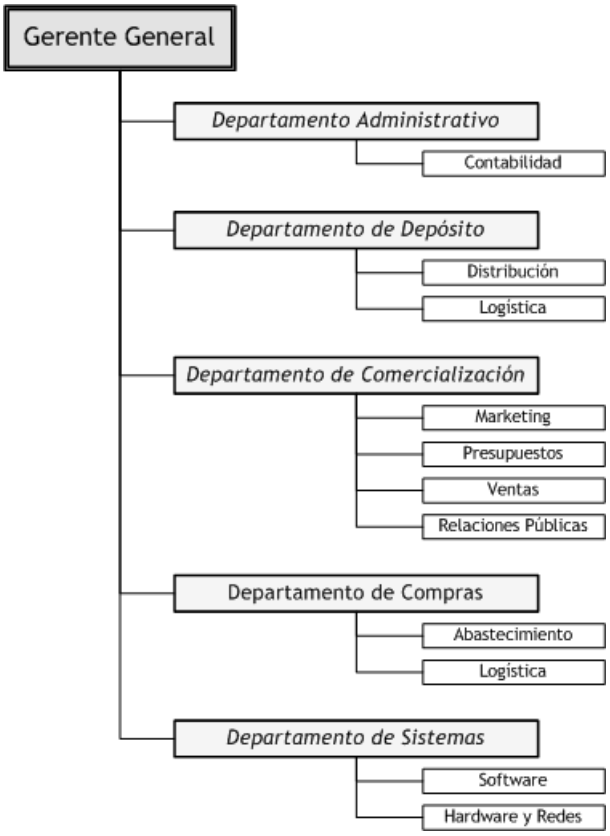


Figura A.1: Organigrama.

A.6. Datos del entorno específico

L@s client@s con que cuenta son bastantes variad@s y cubren un amplio margen. L@s mism@s son tanto provinciales, como nacionales, con diferentes tipos de poder adquisitivo.

Con respecto a sus proveedor@s, la empresa posee en algunos rubros diversas opciones de las cuales puede elegir y comparar, pero en otros solo cuenta con pocas alternativas.

Además, tiene como rivales a nivel de mayoreo, vari@s competidor@s importantes y ya consolidad@s en el mercado, pero, a nivel minorista aventaja por su tamaño y volumen de actividades a sus principales competidor@s.

A.7. Relación de las metas de la organización con las del DWH

El DWH coincide con la metas de la empresa, ya que esta necesita mejorar su eficiencia en la toma de decisiones y contar con información detallada a tal fin. Esto es vital, ya que es muy importante para procurar una mayor ventaja competitiva conocer cuáles son los factores que inciden directamente sobre su rentabilidad, como así también, analizar su relación con otros factores y sus respectivos por qué.

El DWH aportará un gran valor a la empresa, entre las principales ventajas e inconvenientes que solucionará se pueden mencionar los siguientes:

- Permitirá a l@s usuari@s tener una visión general del negocio.
- Transformará datos operativos en información analítica, enfocada a la toma de decisiones.
- Se podrán generar reportes dinámicos, ya que actualmente son estáticos y no ofrecen ninguna facilidad de análisis.
- Soportará la estrategia de la empresa.
- Aportará a la mejora continua de la estructura de la empresa.

A.8. Procesos

Los principales procesos que se llevan a cabo son los siguientes:

- Venta:
  - Minorista: es la que se le realiza a l@s client@s particulares que se acercan hasta la empresa para adquirir los productos que requieren.
  - Mayorista: es la que se le efectúa a l@s grandes client@s, ya sea por medio de comunicaciones telefónicas, o a través de visitas o reuniones.
  - Al realizarse una venta, el departamento de Depósito se encarga de controlar el stock, realizar encargos de mercadería en caso de no cubrir lo solicitado, armar el pedido y enviarlo por medio de transporte propio o de tercer@s al destino correspondiente.
- Compra:
  - El departamento de Compras, al recibir del departamento de Depósito las necesidades de mercadería, realiza una comparación de los productos ofrecidos por sus diferentes proveedor@s en cuestión de precio, calidad y confianza. Posteriormente, se efectúa el pedido correspondiente.

Apéndice B

Licencia de Documentación Libre de GNU

Versión 1.2, Noviembre 2002

This is an unofficial translation of the GNU Free Documentation License into Spanish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documentation that uses the GNU FDL – only the original English text of the GNU FDL does that. However, we hope that this translation will help Spanish speakers understand the GNU FDL better.

Ésta es una traducción no oficial de la GNU Free Document License a Español (Castellano). No ha sido publicada por la Free Software Foundation y no establece legalmente los términos de distribución para trabajos que usen la GFDL (sólo el texto de la versión original en Inglés de la GFDL lo hace). Sin embargo, esperamos que esta traducción ayude los hispanohablantes a entender mejor la GFDL. La versión original de la GFDL esta disponible en la Free Software Foundation.

Esta traducción está basada en una de la versión 1.1 de Igor Támara y Pablo Reyes. Sin embargo la responsabilidad de su interpretación es de Joaquín Seoane.

B.1. Preámbulo

El propósito de esta Licencia es permitir que un manual, libro de texto, u otro documento escrito sea *libre* en el sentido de libertad: asegurar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, de manera comercial o no. En segundo término, esta Licencia proporciona al autor y al editor[2] una manera de

[1] Ésta es la traducción del Copyright de la Licencia, no es el Copyright de esta traducción no autorizada.

[2] La licencia original dice *publisher*, que es, estrictamente, quien publica, diferente de *editor*, que es más bien quien prepara un texto para publicar. En castellano *editor* se usa para ambas cosas.

obtener reconocimiento por su trabajo, sin que se le considere responsable de las modificaciones realizadas por otros.

Esta Licencia es de tipo *copyleft*, lo que significa que los trabajos derivados del documento deben a su vez ser libres en el mismo sentido. Complementa la Licencia Pública General de GNU, que es una licencia tipo copyleft diseñada para el software libre.

Hemos diseñado esta Licencia para usarla en manuales de software libre, ya que el software libre necesita documentación libre: un programa libre debe venir con manuales que ofrezcan la mismas libertades que el software. Pero esta licencia no se limita a manuales de software; puede usarse para cualquier texto, sin tener en cuenta su temática o si se publica como libro impreso o no. Recomendamos esta licencia principalmente para trabajos cuyo fin sea instructivo o de referencia.

B.2. Aplicabilidad y definiciones

Esta Licencia se aplica a cualquier manual u otro trabajo, en cualquier soporte, que contenga una nota del propietario de los derechos de autor que indique que puede ser distribuido bajo los términos de esta Licencia. Tal nota garantiza en cualquier lugar del mundo, sin pago de derechos y sin límite de tiempo, el uso de dicho trabajo según las condiciones aquí estipuladas. En adelante la palabra *Documento* se referirá a cualquiera de dichos manuales o trabajos. Cualquier persona es un licenciatario y será referido como *Usted*. Usted acepta la licencia si copia. modifica o distribuye el trabajo de cualquier modo que requiera permiso según la ley de propiedad intelectual.

Una *Versión Modificada* del Documento significa cualquier trabajo que contenga el Documento o una porción del mismo, ya sea una copia literal o con modificaciones y/o traducciones a otro idioma.

Una *Sección Secundaria* es un apéndice con título o una sección preliminar del Documento que trata exclusivamente de la relación entre los autores o editores y el tema general del Documento (o temas relacionados) pero que no contiene nada que entre directamente en dicho tema general (por ejemplo, si el Documento es en parte un texto de matemáticas, una Sección Secundaria puede no explicar nada de matemáticas). La relación puede ser una conexión histórica con el tema o temas relacionados, o una opinión legal, comercial, filosófica, ética o política acerca de ellos.

Las *Secciones Invariantes* son ciertas Secciones Secundarias cuyos títulos son designados como Secciones Invariantes en la nota que indica que el documento es liberado bajo esta Licencia. Si una sección no entra en la definición de Secundaria, no puede designarse como Invariante. El documento puede no tener Secciones Invariantes. Si el Documento no identifica las Secciones Invariantes, es que no las tiene.

Los *Textos de Cubierta* son ciertos pasajes cortos de texto que se listan como Textos de Cubierta Delantera o Textos de Cubierta Trasera en la nota que indica que el documento es liberado bajo esta Licencia. Un Texto de Cubierta Delantera puede tener como mucho 5 palabras, y uno de Cubierta Trasera puede tener hasta 25 palabras.

Una copia *Transparente* del Documento, significa una copia para lectura en máquina, representada en un formato cuya especificación está disponible al público en general, apto para que los contenidos puedan ser vistos y editados directamente con editores de texto genéricos o (para imágenes compuestas por puntos) con programas genéricos

de manipulación de imágenes o (para dibujos) con algún editor de dibujos ampliamente disponible, y que sea adecuado como entrada para formateadores de texto o para su traducción automática a formatos adecuados para formateadores de texto. Una copia hecha en un formato definido como Transparente, pero cuyo marcaje o ausencia de él haya sido diseñado para impedir o dificultar modificaciones posteriores por parte de los lectores no es Transparente. Un formato de imagen no es Transparente si se usa para una cantidad de texto sustancial. Una copia que no es *Transparente* se denomina *Opaca*.

Como ejemplos de formatos adecuados para copias Transparentes están ASCII puro sin marcaje, formato de entrada de Texinfo, formato de entrada de LaTeX, SGML o XML usando una DTD disponible públicamente, y HTML, PostScript o PDF simples, que sigan los estándares y diseñados para que los modifiquen personas. Ejemplos de formatos de imagen transparentes son PNG, XCF y JPG. Los formatos Opacos incluyen formatos propietarios que pueden ser leídos y editados únicamente en procesadores de palabras propietarios, SGML o XML para los cuáles las DTD y/o herramientas de procesamiento no estén ampliamente disponibles, y HTML, PostScript o PDF generados por algunos procesadores de palabras sólo como salida.

La *Portada* significa, en un libro impreso, la página de título, más las páginas siguientes que sean necesarias para mantener legiblemente el material que esta Licencia requiere en la portada. Para trabajos en formatos que no tienen página de portada como tal, *Portada* significa el texto cercano a la aparición más prominente del título del trabajo, precediendo el comienzo del cuerpo del texto.

Una sección *Titulada XYZ* significa una parte del Documento cuyo título es precisamente XYZ o contiene XYZ entre paréntesis, a continuación de texto que traduce XYZ a otro idioma (aquí XYZ se refiere a nombres de sección específicos mencionados más abajo, como *Agradecimientos*, *Dedicatorias* , *Aprobaciones* o *Historia*. *Conservar el Título* de tal sección cuando se modifica el Documento significa que permanece una sección *Titulada XYZ* según esta definición[3].

El Documento puede incluir Limitaciones de Garantía cercanas a la nota donde se declara que al Documento se le aplica esta Licencia. Se considera que estas Limitaciones de Garantía están incluidas, por referencia, en la Licencia, pero sólo en cuanto a limitaciones de garantía: cualquier otra implicación que estas Limitaciones de Garantía puedan tener es nula y no tiene efecto en el significado de esta Licencia.

B.3. Copia literal

Usted puede copiar y distribuir el Documento en cualquier soporte, sea en forma comercial o no, siempre y cuando esta Licencia, las notas de copyright y la nota que indica que esta Licencia se aplica al Documento se reproduzcan en todas las copias y que usted no añada ninguna otra condición a las expuestas en esta Licencia. Usted no puede usar medidas técnicas para obstruir o controlar la lectura o copia posterior de las copias que usted haga o distribuya. Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias también deberá seguir las condiciones de la sección 3.

Usted también puede prestar copias, bajo las mismas condiciones establecidas anteriormente, y puede exhibir copias públicamente.

[3] En sentido estricto esta licencia parece exigir que los títulos sean exactamente *Acknowledgements*, *Dedications*, *Endorsements* e *History*, en inglés.

B.4. Copiado en cantidad

Si publica copias impresas del Documento (o copias en soportes que tengan normalmente cubiertas impresas) que sobrepasen las 100, y la nota de licencia del Documento exige Textos de Cubierta, debe incluir las copias con cubiertas que lleven en forma clara y legible todos esos Textos de Cubierta: Textos de Cubierta Delantera en la cubierta delantera y Textos de Cubierta Trasera en la cubierta trasera. Ambas cubiertas deben identificarlo a Usted clara y legiblemente como editor de tales copias. La cubierta debe mostrar el título completo con todas las palabras igualmente prominentes y visibles. Además puede añadir otro material en las cubiertas. Las copias con cambios limitados a las cubiertas, siempre que conserven el título del Documento y satisfagan estas condiciones, pueden considerarse como copias literales.

Si los textos requeridos para la cubierta son muy voluminosos para que ajusten legiblemente, debe colocar los primeros (tantos como sea razonable colocar) en la verdadera cubierta y situar el resto en páginas adyacentes.

Si Usted publica o distribuye copias Opacas del Documento cuya cantidad exceda las 100, debe incluir una copia Transparente, que pueda ser leída por una máquina, con cada copia Opaca, o bien mostrar, en cada copia Opaca, una dirección de red donde cualquier usuario de la misma tenga acceso por medio de protocolos públicos y estandarizados a una copia Transparente del Documento completa, sin material adicional. Si usted hace uso de la última opción, deberá tomar las medidas necesarias, cuando comience la distribución de las copias Opacas en cantidad, para asegurar que esta copia Transparente permanecerá accesible en el sitio establecido por lo menos un año después de la última vez que distribuya una copia Opaca de esa edición al público (directamente o a través de sus agentes o distribuidores).

Se solicita, aunque no es requisito, que se ponga en contacto con los autores del Documento antes de redistribuir gran número de copias, para darles la oportunidad de que le proporcionen una versión actualizada del Documento.

B.5. Modificaciones

Puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que usted libere la Versión Modificada bajo esta misma Licencia, con la Versión Modificada haciendo el rol del Documento, por lo tanto dando licencia de distribución y modificación de la Versión Modificada a quienquiera posea una copia de la misma. Además, debe hacer lo siguiente en la Versión Modificada:

- A) Usar en la Portada (y en las cubiertas, si hay alguna) un título distinto al del Documento y de sus versiones anteriores (que deberían, si hay alguna, estar listadas en la sección de Historia del Documento). Puede usar el mismo título de versiones anteriores al original siempre y cuando quien las publicó originalmente otorgue permiso.
- B) Listar en la Portada, como autores, una o más personas o entidades responsables de la autoría de las modificaciones de la Versión Modificada, junto con por lo menos cinco de los autores principales del Documento (todos sus autores principales, si hay menos de cinco), a menos que le eximan de tal requisito.
- C) Mostrar en la Portada como editor el nombre del editor de la Versión Modificada.
- D) Conservar todas las notas de copyright del Documento.

- E) Añadir una nota de copyright apropiada a sus modificaciones, adyacente a las otras notas de copyright.
- F) Incluir, inmediatamente después de las notas de copyright, una nota de licencia dando el permiso para usar la Versión Modificada bajo los términos de esta Licencia, como se muestra en la Adenda al final de este documento.
- G) Conservar en esa nota de licencia el listado completo de las Secciones Invariantes y de los Textos de Cubierta que sean requeridos en la nota de Licencia del Documento original.
- H) Incluir una copia sin modificación de esta Licencia.
- I) Conservar la sección Titulada *Historia*, conservar su Título y añadirle un elemento que declare al menos el título, el año, los nuevos autores y el editor de la Versión Modificada, tal como figuran en la Portada. Si no hay una sección Titulada *Historia* en el Documento, crear una estableciendo el título, el año, los autores y el editor del Documento, tal como figuran en su Portada, añadiendo además un elemento describiendo la Versión Modificada, como se estableció en la oración anterior.
- J) Conservar la dirección en red, si la hay, dada en el Documento para el acceso público a una copia Transparente del mismo, así como las otras direcciones de red dadas en el Documento para versiones anteriores en las que estuviese basado. Pueden ubicarse en la sección *Historia*. Se puede omitir la ubicación en red de un trabajo que haya sido publicado por lo menos cuatro años antes que el Documento mismo, o si el editor original de dicha versión da permiso.
- K) En cualquier sección Titulada *Agradecimientos* o *Dedicatorias*, Conservar el Título de la sección y conservar en ella toda la sustancia y el tono de los agradecimientos y/o dedicatorias incluidas por cada contribuyente.
- L) Conservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Números de sección o el equivalente no son considerados parte de los títulos de la sección.
- M) Borrar cualquier sección titulada *Aprobaciones*. Tales secciones no pueden estar incluidas en las Versiones Modificadas.
- N) No cambiar el título de ninguna sección existente a *Aprobaciones* ni a uno que entre en conflicto con el de alguna Sección Invariante.
- O) Conservar todas las Limitaciones de Garantía.

Si la Versión Modificada incluye secciones o apéndices nuevos que califiquen como Secciones Secundarias y contienen material no copiado del Documento, puede opcionalmente designar algunas o todas esas secciones como invariantes. Para hacerlo, añada sus títulos a la lista de Secciones Invariantes en la nota de licencia de la Versión Modificada. Tales títulos deben ser distintos de cualquier otro título de sección.

Puede añadir una sección titulada *Aprobaciones*, siempre que contenga únicamente aprobaciones de su Versión Modificada por otras fuentes –por ejemplo, observaciones de peritos o que el texto ha sido aprobado por una organización como la definición oficial de un estándar.

Puede añadir un pasaje de hasta cinco palabras como Texto de Cubierta Delantera y un pasaje de hasta 25 palabras como Texto de Cubierta Trasera en la Versión Modificada. Una entidad solo puede añadir (o hacer que se añada) un pasaje al Texto de Cubierta Delantera y uno al de Cubierta Trasera. Si el Documento ya incluye un textos de cubiertas

añadidos previamente por usted o por la misma entidad que usted representa, usted no puede añadir otro; pero puede reemplazar el anterior, con permiso explícito del editor que agregó el texto anterior.

Con esta Licencia ni los autores ni los editores del Documento dan permiso para usar sus nombres para publicidad ni para asegurar o implicar aprobación de cualquier Versión Modificada.

B.6. Combinación de documentos

Usted puede combinar el Documento con otros documentos liberados bajo esta Licencia, bajo los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, listadas todas como Secciones Invariantes del trabajo combinado en su nota de licencia. Así mismo debe incluir la Limitación de Garantía.

El trabajo combinado necesita contener solamente una copia de esta Licencia, y puede reemplazar varias Secciones Invariantes idénticas por una sola copia. Si hay varias Secciones Invariantes con el mismo nombre pero con contenidos diferentes, haga el título de cada una de estas secciones único añadiéndole al final del mismo, entre paréntesis, el nombre del autor o editor original de esa sección, si es conocido, o si no, un número único. Haga el mismo ajuste a los títulos de sección en la lista de Secciones Invariantes de la nota de licencia del trabajo combinado.

En la combinación, debe combinar cualquier sección Titulada *Historia* de los documentos originales, formando una sección Titulada *Historia*; de la misma forma combine cualquier sección Titulada *Agradecimientos*, y cualquier sección Titulada *Dedicatorias*. Debe borrar todas las secciones tituladas *Aprobaciones*.

B.7. Colecciones de documentos

Puede hacer una colección que conste del Documento y de otros documentos liberados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en todos los documentos por una sola copia que esté incluida en la colección, siempre que siga las reglas de esta Licencia para cada copia literal de cada uno de los documentos en cualquiera de los demás aspectos.

Puede extraer un solo documento de una de tales colecciones y distribuirlo individualmente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los demás aspectos relativos a la copia literal de dicho documento.

B.8. Agregación con trabajos independientes

Una recopilación que conste del Documento o sus derivados y de otros documentos o trabajos separados e independientes, en cualquier soporte de almacenamiento o distribución, se denomina un *agregado* si el copyright resultante de la compilación no se usa para limitar los derechos de los usuarios de la misma más allá de lo que los de los trabajos individuales permiten. Cuando el Documento se incluye en un agregado, esta Licencia no se aplica a otros trabajos del agregado que no sean en sí mismos derivados

del Documento.

Si el requisito de la sección 3 sobre el Texto de Cubierta es aplicable a estas copias del Documento y el Documento es menor que la mitad del agregado entero, los Textos de Cubierta del Documento pueden colocarse en cubiertas que enmarquen solamente el Documento dentro del agregado, o el equivalente electrónico de las cubiertas si el documento está en forma electrónica. En caso contrario deben aparecer en cubiertas impresas enmarcando todo el agregado.

B.9. Traducción

La Traducción es considerada como un tipo de modificación, por lo que usted puede distribuir traducciones del Documento bajo los términos de la sección 4. El reemplazo las Secciones Invariantes con traducciones requiere permiso especial de los dueños de derecho de autor, pero usted puede añadir traducciones de algunas o todas las Secciones Invariantes a las versiones originales de las mismas. Puede incluir una traducción de esta Licencia, de todas las notas de licencia del documento, así como de las Limitaciones de Garantía, siempre que incluya también la versión en Inglés de esta Licencia y las versiones originales de las notas de licencia y Limitaciones de Garantía. En caso de desacuerdo entre la traducción y la versión original en Inglés de esta Licencia, la nota de licencia o la limitación de garantía, la versión original en Inglés prevalecerá.

Si una sección del Documento está Titulada *Agradecimientos*, *Dedicatorias* o *Historia* el requisito (sección 4) de Conservar su Título (Sección 1) requerirá, típicamente, cambiar su título.

B.10. Terminación

Usted no puede copiar, modificar, sublicenciar o distribuir el Documento salvo por lo permitido expresamente por esta Licencia. Cualquier otro intento de copia, modificación, sublicenciamiento o distribución del Documento es nulo, y dará por terminados automáticamente sus derechos bajo esa Licencia. Sin embargo, los terceros que hayan recibido copias, o derechos, de usted bajo esta Licencia no verán terminadas sus licencias, siempre que permanezcan en total conformidad con ella.

B.11. Revisiones futuras de esta licencia

De vez en cuando la Free Software Foundation puede publicar versiones nuevas y revisadas de la Licencia de Documentación Libre GNU. Tales versiones nuevas serán similares en espíritu a la presente versión, pero pueden diferir en detalles para solucionar nuevos problemas o intereses. Vea http://www.gnu.org/copyleft.

Cada versión de la Licencia tiene un número de versión que la distingue. Si el Documento especifica que se aplica una versión numerada en particular de esta licencia o *cualquier versión posterior*, usted tiene la opción de seguir los términos y condiciones de la versión especificada o cualquiera posterior que haya sido publicada (no como borrador) por la Free Software Foundation. Si el Documento no especifica un número de versión de esta Licencia, puede escoger cualquier versión que haya sido publicada (no como borrador) por la Free Software Foundation.

B.12. Adenda: cómo usar esta Licencia en sus documentos

Para usar esta licencia en un documento que usted haya escrito, incluya una copia de la Licencia en el documento y ponga el siguiente copyright y nota de licencia justo después de la página de título:

> Copyright ©AÑO SU NOMBRE. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni Textos de Cubierta Delantera ni Textos de Cubierta Trasera. Una copia de la licencia está incluida en la sección titulada *GNU Free Documentation License*.

Si tiene Secciones Invariantes, Textos de Cubierta Delantera y Textos de Cubierta Trasera, reemplace la frase *sin ... Trasera* por esto:

- siendo las Secciones Invariantes LISTE SUS TÍTULOS, siendo los Textos de Cubierta Delantera LISTAR, y siendo sus Textos de Cubierta Trasera LISTAR.

Si tiene Secciones Invariantes sin Textos de Cubierta o cualquier otra combinación de los tres, mezcle ambas alternativas para adaptarse a la situación.

Si su documento contiene ejemplos de código de programa no triviales, recomendamos liberar estos ejemplos en paralelo bajo la licencia de software libre que usted elija, como la Licencia Pública General de GNU (*GNU General Public License*), para permitir su uso en software libre.

Bibliografía

[1] Laboratorios, prácticos, apuntes y bibliografía de la materia MOTORES DE BASE DE DATOS – Ing. Mauricio Rizzi, Ing. Mariano García Mattío – Instituto Universitario Aeronáutico (IUA) - Año 2006.

[2] Laboratorios, prácticos, apuntes y bibliografía del curso SISTEMAS AVANZADOS DE BASE DE DATOS CON SOPORTE PARA LA TOMA DE DECISIONES – Ing. Mauricio Rizzi – Universidad Católica de Córdoba (UCC) – Año 2006.

[3] ESTRATEGIA COMPETITIVA, Técnicas para el Análisis de los Sectores Industriales y de la Competencia – Michael E. Porter – Año 2000 – Vigésima séptima reimpresión.

[4] EL NUEVO DIRECTIVO RACIONAL, Análisis de problemas y toma de decisiones – Charles H. Kepner, Benjamin B. Tregoe – Ed. McGraw-Hill – Año 1992.

[5] Ingeniería del Software, Un enfoque práctico – Roger S. Pressman. MacGraw-Hill – Año 2001 – 5ta Edición.

[6] SISTEMAS DE BASES DE DATOS, Un enfoque práctico para diseño, implementación y gestión – Thomas M. Connolly, Carolyn E. Begg – Addison-Wesley – Año 2005 – 4ta Edición.

[7] BI-FLOSS: Business Intelligence – Free/Libre Open Source Software [http://bifloss.blogspot.com] – Ing. de Almeida Rodrigo, Ing. Heredia Mariano – Abril de 2008.

[8] CUADRO DE MANDO INTEGRAL (The Balanced Scorecard) – Robert S. Kaplan, David P. Norton – Ed. Gestión 2000 – Año 1992.

[9] MASTERING DATA WAREHOUSE DESIGN, Relational and Dimensional Techniques – Claudia Imhoff, Nicholas Galemmo, Jonathan G. Geiger – Ed. WILEY – Año 2003.

[10] Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL – Roland Bouman, Jos van Dongen – Ed. WILEY – Año 2010.

[11] Sitios web:

- [http://wikipedia.org] Wikipedia.
- [http://sistemasdecisionales.blogspot.com] Sistemas Decisionales, algo más que Business Intelligence – Jorge Fernández González.
- [http://informationmanagement.wordpress.com] Information Management – Josep Curto Díaz.
- [http://todobi.blogspot.com] Todo BI: Business Intelligence, Data Warehouse, CRM y mucho más....
- [http://www.intelineg.com] Inteligencia de Negocio – Rémi Grossat.

- [http://www.beyeblogs.com/karthikonbi] Business Intelligence - A Practitioner's Thoughts – Karthikeyan Sankaran.
- [http://magm3333.googlepages.com] Programación, Base de Datos e IT en general – Ing. Mariano Alberto García Mattío.
- [http://estudiandobi.blogspot.com] Estudiando Business Intelligence – cduque.
- [http://www.beyenetwork.es/articles] BeyeNETWORK España: Articles.
- [http://analisisbi.blogspot.com] Análisis BI – Diego Arenas C.
- [http://www.dataprix.com] Dataprix: Transformando datos en conocimiento – Ing. Carlos Fernandez.
- [http://www.roberto-espinosa.es] El Rincón del BI: Descubriendo el Business Intelligence – Ing. Roberto Espinosa Milla.

Índice alfabético

Almacenamiento intermedio, 22
Almacén de Datos Corporativo, 78
Análisis de requerimientos, 89
Atributos, 35, 113

Base de datos multidimensional, 28
Bottom-Up, 74
Business Intelligence, 5
Business Models, 76

Carga, 25
Carga Inicial (Inicial Load), 25
Carga Total (Full Load), 26
Claves Subrogadas, 124
Codificación, 22
Codificar, 121
Consultas, 66
Convenciones de nombramiento, 24
Correspondencias, 93
Cubo Multidimensional, 112
Cubo Multidimensional: creación y ejemplificación, 45
Cubo Multidimensional: introducción, 33

Dashboards, 67
Data Mart, 73, 79, 119
Data Mining, 68
Data Warehouse, 9
Data Warehouse Manager, 27
Data Warehousing, 9
Dato agregado, 31
Datos altamente resumidos, 17
Datos Anómalos (Outliers), 24
Datos de referencia, 29
Datos Faltantes (Missing Values), 25
Datos ligeramente resumidos, 17
Desnormalización, 38
Detalle de datos actuales, 17
Detalle de datos históricos, 17
Detección de Desviación, 70
Dimensiones Clustering, 130
Dimensiones Degeneradas, 129
Dimensiones lentamente cambiantes, 125
Discretizar, 25, 122
Drill-across, 57
Drill-down, 53
Drill-through, 63
Drill-up, 55
Día Juliano, 30

EIS, 70
Esquema Constelación, 40
Esquema Copo de Nieve, 39
Esquema en Estrella, 37
Estandarizar, 121
ETL, 11, 21, 118
Extracción, 22

Factorizar, 121
Fuentes múltiples, 24

Granularidad, 37

Hechos, 30
HEFESTO, 85
Herramientas de Consulta y Análisis, 64
HOLAP, 44

Indicadores, 34, 90, 112
Integración de Datos, 11, 21, 105
Integrada, 11

Jerarquías, 35, 114

Limpieza de Datos (Data Cleansing), 24
Load Manager, 21

Mapping, 50
Medida de atributos, 23
Metadatos, 18, 49
Modelo Conceptual, 91
Modelo Conceptual ampliado, 98
Modelo Lógico, 99
MOLAP, 43

No volátil, 13
Normalización, 38

OLAP, 66
OLTP, 20
Operational Data Store, 78
Orientada al negocio, 10

Page, 60
Particionamiento, 76
Perspectivas, 90
Pivot, 59
Programación Genética, 69

Query Manager, 51

Redes Neuronales, 69
Redundancia, 16
Relación, 36
Relación muchos a muchos, 123
Reportes, 66
ROLAP, 42
Roll-across, 58

SCD, 125
SCD Tipo 1: Sobreescribir, 126
SCD Tipo 2: Añadir fila, 127
SCD Tipo 3: Añadir columna, 128
SCD Tipo 4: Tabla de Historia separada, 129
SCD Tipo 6: Híbrido, 129
Sello de tiempo, 12
SGBD, 75
Sistema de Misión Crítica, 73, 118
Sistemas Expertos, 69
Staging Area, 77

Tabla de Dimensión Tiempo, 29
Tablas de Dimensiones, 28, 99
Tablas de Hechos, 30, 101
Tablas de hechos agregadas, 33
Tablas de hechos preagregadas, 33
Top-Down, 74
Transformación, 22

Uniones, 104
Usuarios, 71

Variante en el tiempo, 12

Árboles de Decisión, 70
Áreas de Datos, 77