UNIVERSIDAD NACIONAL AUTÓNOMA

DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
CAMPUS ARAGÓN

“DISEÑANDO UN DATA WAREHOUSE PARA LA EMPRESA DE SEGUROS”

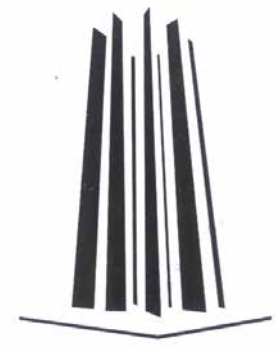
T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

SILVA VARGAS JOSÉ LUIS



ASESOR: M. EN E. FLORES DÍAZ IMELDA DE LA LUZ

SAN JUAN DE ARAGÓN, ESTADO DE MÉXICO 2007

A mis padres, Ada y José Luis.

Cuyo esfuerzo y amor, han hecho posible la culminación de este proyecto.

“Diseñando un Data Warehouse Para la empresa de seguros”

Antecedentes

El Data Warehouse (DW) es un concepto que ha ido madurando a lo largo de la última década, pasando a ser una tecnología esencial que suministra información empresarial integrada, nos ayuda a comprender, por ejemplo, los patrones de compra de nuestros clientes, identificar nichos de mercado y obtener beneficios que se traducen en oportunidades de crecimiento.

Un almacén de datos o DW, es una colección de datos y procesos orientados a temas, integrados y no volátiles[1], que en conjunto nos apoyan a tomar mejores decisiones, con información adecuada, en el momento y lugar correctos.

Una arquitectura de soporte a la toma de decisiones debe ser capaz de dar respuesta a preguntas de negocio, como por ejemplo en la industria del seguro, volumen de pólizas emitidas, siniestralidad en pólizas, número de reclamaciones, etc. Por todo esto, las aplicaciones para DW son sistemas de soporte a la toma de decisiones.

Lo más importante al implantar soluciones para DW es aprovechar estas herramientas en todos los niveles de la empresa, permitiendo que todos los usuarios, independientemente de su experiencia y nivel de conocimientos técnicos, sean capaces de analizar la información y generar indicadores que les sean relevantes.

EL DW provee métodos de extracción, transformación y carga, así como un repositorio de datos, que contiene información optimizada y libre de errores de integridad. De manera típica, la principal fuente de información de un DW es un sistema de información operacional; por ejemplo el sistema de administración y emisión de pólizas de seguros; sin embargo, también puede ser alimentado con datos provenientes de fuentes externas, por ejemplo, el tipo de cambio mensual que publica El Banco de México, el índice nacional de precios al consumidor (INPC), etc.

Desde hace algún tiempo el DW se ha venido utilizando en diversas industrias con resultados realmente buenos:

Ventas	Gobierno
Estudios de mercado	Planeación fuerza laboral
Fidelidad del cliente	Control de gasto
Finanzas	Aerolíneas
Manejo del riesgo	Mejora de las rutas
Detección de fraudes	Manejo de hangares
Manufactura	**Seguros**
Reducción de costos	**Soporte a la cobranza**
Logística	**Manejo de la siniestralidad**

El presente trabajo se enfoca en el diseño y las aplicaciones de un DW para la industria del Seguro, en particular una empresa de seguros en México. Por ello, poniéndonos en contexto, podemos comentar

[1] William H. Inmon, Building the Data Warehouse. John Wiley & Sons, U.S.A. 1992. Pág. 31.

que la principal función de una empresa de seguros, es administrar el riesgo, convirtiéndolo en una actividad lucrativa, pero con una importante misión social. Por lo que el uso de un DW en una empresa de seguros, aporta grandes beneficios en los rubros: gestión de la cobranza, administración del riesgo, así como detectar tanto reclamaciones fraudulentas como los negocios más rentables.

Para tener una visión más detallada de lo que son los seguros, puede consultar el capítulo dos.

Motivación

En la actualidad, las empresas de seguros se desarrollan en un medio en el cuál la fidelidad del cliente es cuestionable debido a la gran presión que ejerce la competencia. Esto aunado a la reducción de las ganancias, las reclamaciones fraudulentas, así como la nueva tendencia de la banca a incursionar en el mercado de seguros en México, está forzando a las empresas de seguros a tomar algunas decisiones realmente difíciles para mantenerse competitivas, y en ciertos casos, incluso para asegurar su permanencia en el mercado.

Conscientes de esta situación las empresas de seguros están más preocupadas por recolectar datos sobre sus clientes y transformarlos en información útil; es en este punto cuando las decisiones basadas en información de calidad, se convierten en una ventaja competitiva.

Como respuesta a estas necesidades, se han desarrollado una nueva categoría de sistemas de información, conocidos como sistemas de soporte a la toma de decisiones o Data Warehouse (DW).

El presente trabajo aborda el diseño de un Data Warehouse, específicamente enfocado a una empresa de seguros en México; tomando como punto de partida el modelo lógico de la base de datos, del sistema de información que da soporte a la producción dentro de la empresa, y que alimentará al DW.

Cabe mencionar que en este trabajo, se incorporan conceptos básicos sobre seguros, DW, nociones del modelo relacional de datos, modelo multidimensional (modelo estrella), así como una guía rápida a la metodología de diseño para DW que ha probado su efectividad.

Objetivos

El objetivo primordial del presente trabajo de tesis, es ofrecer al diseñador principiante, una guía rápida que le asista en su importante tarea. Dicha guía estará encaminada a resolver los problemas más comunes a los que debemos enfrentarnos cuando se inicia el diseño de un DW. En particular, el diseño de un DW para una empresa de seguros; que sea capaz de dar respuesta a las preguntas de negocio más típicas que surgen dentro de una empresa de seguros en México.

Para ello tomamos como referencia el ciclo de administración de proyectos, desarrollado a partir de la experiencia ganada a través de numerosos proyectos sobre Data Warehouse. El ciclo puede dividirse en siete etapas agrupadas de forma lógica. Sin embargo debemos mencionar que sólo nos centraremos en las etapas correspondientes al diseño. No es el objetivo de esta tesis ahondar en las etapas de implementación y administración de los recursos empleados para el proyecto.

En el siguiente diagrama (Figura 1.) podemos ver el detalle de las siete etapas que conforman el ciclo de administración de proyectos:

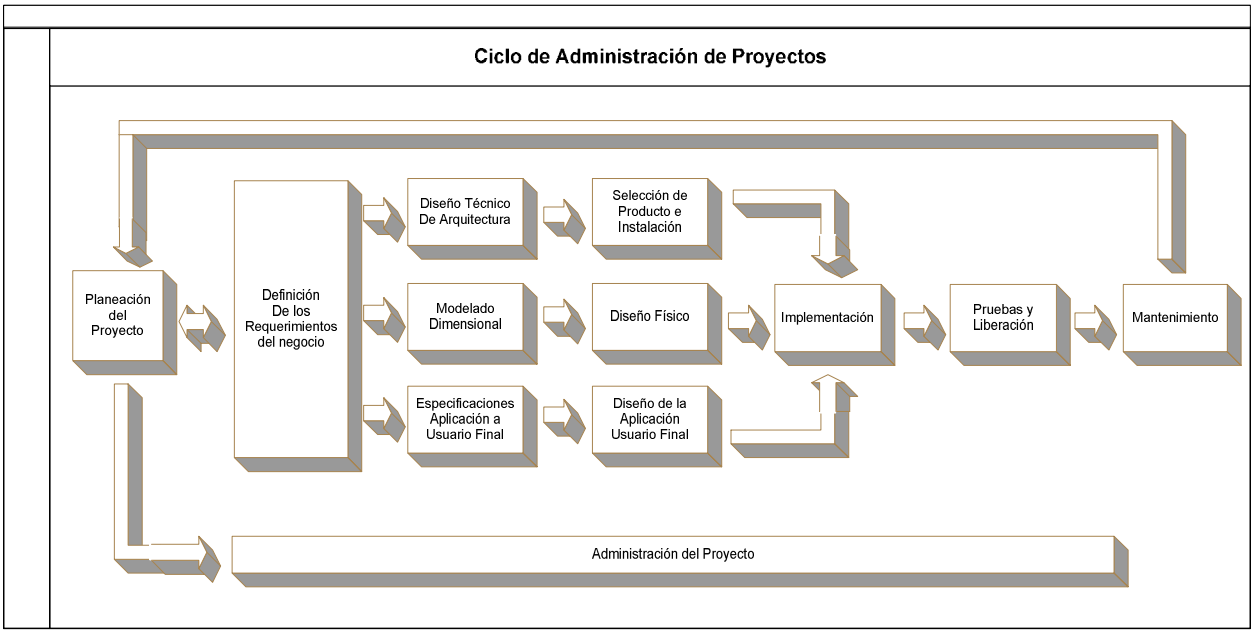


Figura 1. Ciclo de Administración de Proyectos

Las etapas de Planeación del Proyecto y Definición de los Requerimientos de Negocio, están encaminadas a entender y revisar los objetivos del proyecto en relación a los objetivos de la empresa; para de este modo asegurar que el proyecto sea viable, con un fundamento sólido y que además aportará un valor a la empresa.

Posteriormente las etapas de Diseño Técnico, Modelado Dimensional y Especificaciones de Usuario Final, están centradas en generar los fundamentos que servirán como insumo a las etapas posteriores.

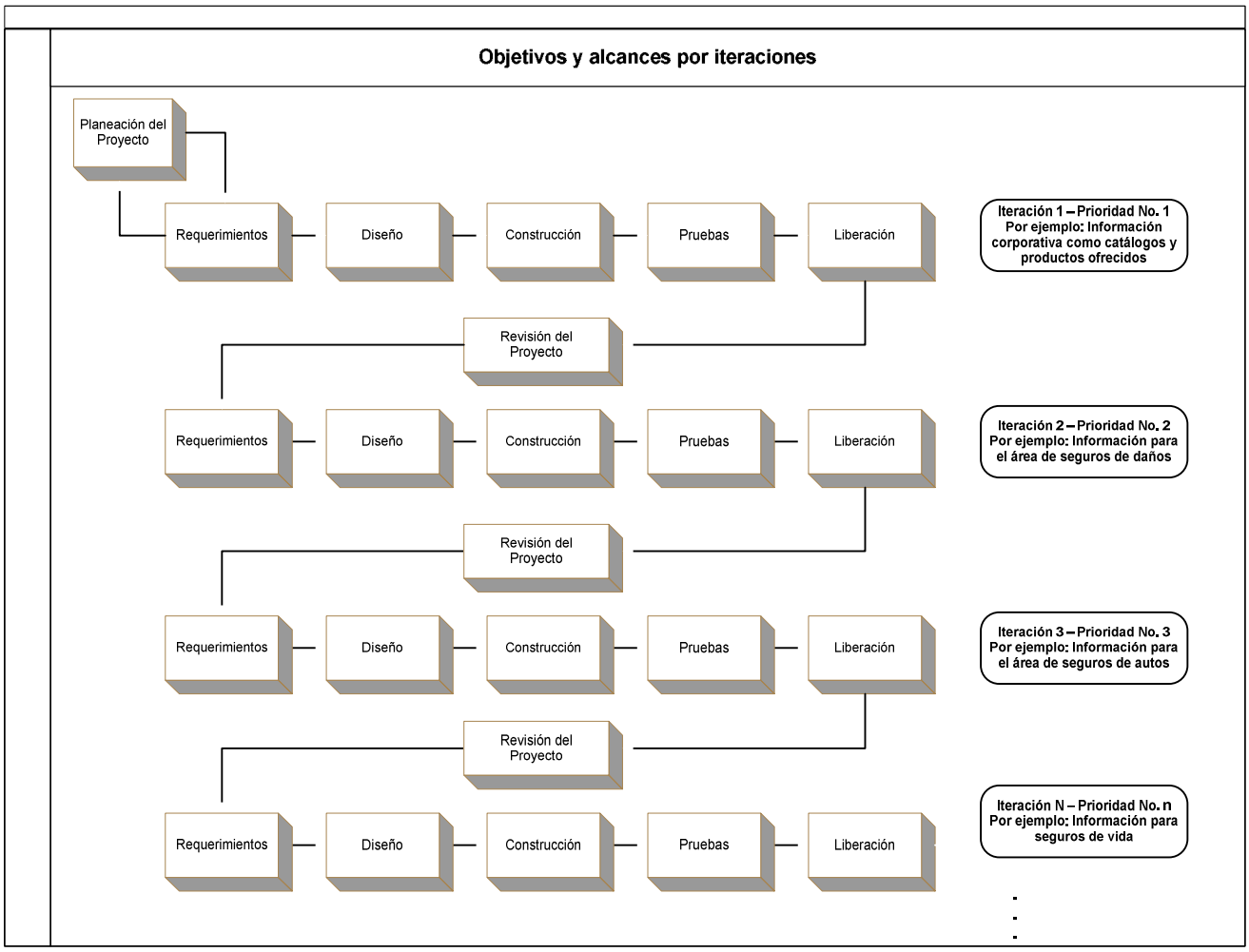
Los conocimientos obtenidos sobre las necesidades y procesos de negocio serán cruciales para el éxito de la etapa de Implementación.

Solución propuesta

Dado que el éxito de un DW depende en gran medida, de un buen diseño, que sea evolutivo y se adapte a las necesidades de información que surgen en la empresa, se hace necesario dividir el diseño en iteraciones. Cada una de estas iteraciones tendrá propósitos y alcances cada vez más específicos, es decir, iniciamos con una etapa troncal que está enfocada a satisfacer las necesidades más generales de información de la empresa, a esto le llamamos DW a nivel corporativo.

Posteriormente, se proponen iteraciones enfocadas a resolver las necesidades particulares de información, para las diversas áreas o departamentos que componen la empresa; en este nivel estaremos hablando de Data Marts.

Cada una de estas iteraciones, será regida por los mismos principios propuestos en el ciclo de administración de proyectos. Para ejemplificarlo, a continuación mostramos una versión resumida del ciclo de administración, con los objetivos y alcances para cada una de las iteraciones.


Objetivos y alcances por iteraciones
Planeación del Proyecto
Requerimientos
Diseño
Construcción
Pruebas
Liberación
Iteración 1 – Prioridad No. 1
Por ejemplo: Información corporativa como catálogos y productos ofrecidos
Revisión del Proyecto
Requerimientos
Diseño
Construcción
Pruebas
Liberación
Iteración 2 – Prioridad No. 2
Por ejemplo: Información para el área de seguros de daños
Revisión del Proyecto
Requerimientos
Diseño
Construcción
Pruebas
Liberación
Iteración 3 – Prioridad No. 3
Por ejemplo: Información para el área de seguros de autos
Revisión del Proyecto
Requerimientos
Diseño
Construcción
Pruebas
Liberación
Iteración N – Prioridad No. n
Por ejemplo: Información para seguros de vida

En este trabajo de tesis, se retoman los aspectos teóricos de la arquitectura DW, pero con un giro especifico para ser aplicados a una empresa de seguros en México. Ofreciendo al diseñador, un punto de referencia para encarar el compromiso de diseñar un DW que sea evolutivo, y sobre todo, un DW que cumpla su propósito.

Contenido del documento

El presente trabajo de tesis está dividido en cuatro capítulos. En el primero se dan algunos antecedentes históricos sobre seguros, revisamos los tipos de seguros que existen, así como sus conceptos básicos.

El capítulo número dos trata el tema Data Warehousing, pasando por los conceptos clave que definen a un DW, la arquitectura básica de un DW, y finalmente la generación de reportes dinámicos que son consideradas herramientas de usuario final.

En el capítulo tres, hacemos una revisión de las diversas tecnologías existentes, que se basan en DW, entre las que podemos encontrar CRM, Work Flow, Balance Score Card, Business Intelligence, etc.

Finalmente el capítulo cuatro trata las aplicaciones que tiene un DW dentro de la empresa de seguros en México.

Capitulo 1

1. Visión general

La industria del seguro tiene como objetivo proteger a la sociedad contra hechos imprevistos que pudieran afectar de manera negativa a las personas y bienes materiales. El manejo de tales imprevistos da lugar a la administración del riesgo. Como consecuencia, las empresas de seguros tienen una importante misión, y hacen de la administración del riesgo una actividad lucrativa.

1.1. Origen del seguro

La historia del seguro es casi tan antigua como las primeras civilizaciones, donde se tenían ciertas prácticas que constituyen los inicios del actual sistema de seguros; probablemente las formas más antiguas de seguros fueron iniciadas por los babilonios e hindis. Estos primeros contratos eran conocidos con el nombre de “Contratos a la Gruesa” y se efectuaban básicamente, entre los banqueros y los propietarios de barcos.

Con frecuencia, el dueño de un barco tomaría prestados los fondos necesarios para comprar carga y financiar un viaje. En el contrato de préstamos a la gruesa se especificaba que si el barco o carga se perdía durante el viaje, el préstamo se entendería como cancelado. Naturalmente, el costo de este contrato era muy elevado, sin embargo, si el banquero financiaba a propietarios cuyas pérdidas resultaban mayores a las esperadas, esto se convertía en un mal negocio.

Los vestigios del seguro de vida se encuentran en Roma, donde era acostumbrado por las asociaciones religiosas, colectar y distribuir fondos entre sus miembros en caso de que alguno de ellos muriera. También en las culturas romana, griega y azteca, podemos encontrar los antecedentes del seguro de retiro; debido a que otorgaban a los ancianos notables algo semejante a una pensión.

Con el crecimiento del comercio durante la edad media tanto en Europa como en el Cercano Oriente, se hizo necesario garantizar la solvencia financiera en caso que ocurriese un desastre de navegación; fue así como Inglaterra se convirtió en el centro marítimo del mundo, y Londres vino a ser la capital aseguradora para el casco y la carga.

El seguro de incendio surgió más tarde en el siglo XVII, después que un incendio destruyó la mayor parte de Londres; a raíz de este suceso se formularon muchos planes, pero la mayoría fracasaron debido a que no constituían reservas adecuadas para enfrentar las pérdidas subsecuentes de los siniestros que ocurrieron.

Las sociedades con objetivo asegurador aparecieron alrededor de 1720, en las etapas iniciales los especuladores y promotores ocasionaron el fracaso financiero de la mayoría de estas nuevas sociedades. Las repercusiones fueron tan serias, que el parlamento inglés restringió las licencias de tal manera que sólo hubo dos compañías autorizadas.

1.2. El contrato

El concepto del contrato de seguro puede resumirse como: "un contrato por el cual una persona se obliga, a cambio de una suma de dinero, a indemnizar a otra, satisfacer una necesidad de esta o entregar a un tercero, dentro de las condiciones convenidas, las cantidades pactadas para compensar las consecuencias de un evento incierto, cuando menos en cuanto al tiempo"[4].

Para otros autores[5] el contrato de seguro "es un contrato sustantivo y oneroso por el cual una persona asume el riesgo de que ocurra un acontecimiento incierto, al menos en cuanto al tiempo, obligándose a realizar una prestación pecuniaria cuando el riesgo se haya convertido en siniestro".

Después de haber analizado un par de definiciones que dan los autores sobre el particular, es necesario proponer un concepto propio, que es el siguiente: El contrato de seguro, es aquel contrato mediante el cual una persona llamada asegurador se obliga, a cambio de una suma de dinero, conocida como prima, a indemnizar a otra llamada asegurado o a la persona que este designe, beneficiario, de un perjuicio o daño que pueda causar un suceso incierto. De tal manera que la suma objeto de indemnización, que fue pactada expresamente, sea pagada cuando ocurra el suceso o riesgo cubierto por el seguro.

1.2.1. Características del contrato

El contrato de seguro presenta las siguientes características:

- Es un acto de comercio.- Efectivamente el contrato de seguro constituye un contrato mercantil, regulado en el Código de Comercio y en otros aspectos supletoriamente por la legislación civil.

- Es un contrato solemne.- El contrato de seguro es solemne, ya que su perfeccionamiento se produce a partir del momento en que el asegurador suscribe la póliza, la firma del asegurador sirve para solemnizar el acuerdo previo de voluntades entre las partes contratantes, respecto a los elementos del seguro.

- Es un contrato bilateral.- En razón de que genera derechos y obligaciones para cada uno de los sujetos contratantes, GARRIGUES[6] al respecto señala: "el tomador de seguros se obliga a pagar la prima y el asegurador se obliga a una prestación pecuniaria: si bien esta prestación esta subordinada a un evento incierto, cual es la realización del siniestro".

- Es un contrato oneroso.- Es oneroso, porque significa para las partes un enriquecimiento y empobrecimiento correlativos. "Por cuanto al tomador del seguro se le impone la obligación de pagar la prima y al asegurador la asunción del riesgo de la que deriva la prestación del pago de la indemnización de la que queda liberado si no se ha pagado la prima antes del siniestro".

- Es un contrato aleatorio.- Es aleatorio porque tanto el asegurado como el asegurador están sometidos a una contingencia que puede representar para uno una utilidad y para el otro una pérdida. Tal contingencia consiste en la posibilidad de que se produzca el siniestro. Al respecto

[4] Montoya Manfredi, Ulises. Derecho Comercial Tomo II. Cultural Cuzco S.A. 1986. Pág.:
[5] Garrigues, Joaquín. Curso de Derecho Mercantil Tomo IV. Editorial Temis 1987.
[6] Garrigues, Joaquín. Curso de Derecho Mercantil Tomo IV. Editorial Temis 1987.

el profesor MONTOYA[7] dice : "El carácter aleatorio del contrato no desaparece por el hecho de que las compañías aseguradoras dispongan de tablas estadísticas que les permite determinar el costo de los riesgos, en función de lo cual fijan el importe de las primas, es decir, que si bien la actividad aseguradora en si es cada vez menos riesgosa en la medida del perfeccionamiento de los medios para determinar la frecuencia de los riesgos, el contrato sigue siendo aleatorio tratándose de cada contrato aislado y respecto del asegurado".

- Es un contrato de ejecución continuada.- Por cuanto los derechos de las partes o los deberes asignados a ellas se van desarrollando en forma continua, a partir de la celebración del contrato hasta su finalización por cualquier causa.

- Es un contrato de adhesión.- El seguro no es un contrato de libre discusión sino de adhesión. Las cláusulas son establecidas por el asegurador, no pudiendo el asegurado discutir su contenido, tan sólo puede aceptar o rechazar el contrato impuesto por el asegurador. Sólo podrá escoger las cláusulas adicionales ofrecidas por el asegurador, pero de ninguna manera podrá variar el contenido del contrato. Pero todo esto dependerá de la voluntad y de la flexibilidad que tenga cada empresa aseguradora.

1.2.2. Interés asegurable

Por interés asegurable se entiende la relación lícita de valor económico sobre un bien. Cuando esta relación se halla amenazada por un riesgo, es un interés asegurable.

Para el profesor MONTOYA[8] el interés es: "la relación por cuya virtud alguien sufre un daño patrimonial por efecto del evento previsto, que no recae en lo que es objeto del seguro, sino en el interés que en el tenga el asegurado"

El interés asegurable es un requisito que debe concurrir en quien desee la cobertura de algún riesgo, reflejado en su deseo verdadero de que el siniestro no se produzca, ya que a consecuencia de él se originaría un perjuicio para su patrimonio.

El principio del interés asegurable se entenderá fácilmente si se tiene en cuenta lo que se esta asegurando, esto quiere decir, el objeto del contrato no es la cosa amenazada por un peligro incierto, sino el interés del asegurado en que el daño no se produzca. El interés asegurable no es solo un simple requisito que imponen los aseguradores, sino una necesidad para velar por la naturaleza de la institución aseguradora. En efecto si tomamos en cuenta estas premisas, tendríamos que la existencia de contratos sin interés asegurable, produciría necesariamente un aumento en la siniestralidad y esto motivaría una elevación de las primas y el verdadero asegurado tendría que pagar un precio superior al que realmente correspondería a su riesgo, perjudicándose así no sólo él, sino también la economía del país, que tendría que soportar una carga económica superior a la debida.

[7] Montoya Manfredi, Ulises. Derecho Comercial Tomo II. Cultural Cuzco S.A. 1986.
[8] Montoya Manfredi, Ulises. Derecho Comercial Tomo II. Cultural Cuzco S.A. 1986.

1.2.3. Riesgo asegurable

Es un evento posible, incierto y futuro, capaz de ocasionar un daño del cual surja una necesidad patrimonial. El acontecimiento debe ser posible, porque de otro modo no existiría inseguridad. Lo imposible no origina riesgo. Debe ser cierto, porque si necesariamente va a ocurrir, nadie asumiría la obligación de repararlo.

Sin riesgo no puede haber seguro, porque al faltar la posibilidad de que se produzca el evento dañoso, ni podrá existir daño ni cabrá pensar en indemnización alguna.

El carácter eventual del riesgo implica la exclusión de la certeza así como de la imposibilidad, abarcando el caso fortuito, sin descartar la voluntad de las partes, siempre y cuando el suceso no se encuentre sometido inevitable y exclusivamente a ella. La incertidumbre no debe tener carácter absoluto sino que debe ser visto desde una perspectiva económica, para lo cual resulta suficiente la incertidumbre del tiempo en que acontecerá, es decir, ya sea en lo que toca a la realización del evento o al momento en que este se producirá.

El riesgo presenta ciertas características que son las siguientes:

- Incierto
- Aleatorio
- Posible
- Concreto
- Licito
- Fortuito
- De contenido económico

En el contrato de seguro el asegurador no puede asumir el riesgo de una manera abstracta, sino que este deber ser debidamente individualizado, ya que no todos los riesgos son asegurables, es por ello que se deben limitarse e individualizarse, dentro de la relación contractual.

1.2.4. Prima

La prima es otro de los elementos indispensables del contrato de seguro, constituye la suma que debe pagar el asegurado a efecto de que el asegurador asuma la obligación de resarcir las pérdidas y daños que ocasione el siniestro, en caso de que se produzca. Este monto se fija proporcionalmente, tomando en cuenta la duración del seguro, el grado de probabilidad de que el siniestro ocurra y la indemnización pactada.

Al respecto RODRIGUEZ[9] señala: "es la cantidad que paga el asegurado como contrapartida de la obligación, resarcitiva e indemnizatoria del asegurador. Es el precio del seguro y un elemento esencial de la institución.

Representa el presupuesto de la relación contractual, por lo que debe cancelarse por adelantado, al emitirse la póliza.

[9] Rodríguez Pastor, Carlos. Derecho de Seguros y Reaseguros. Fundación MJ Bustamante de la Fuente 1987.

Existen distintos tipos de primas:

Prima natural: En los seguros de vida es la prima que depende del cómputo matemático del riesgo. Por esta razón, a mayor riesgo, mayor será la prima natural, y viceversa.

Prima pura: Es la prima de riesgo de los otros ramos de seguros.

Prima comercial: esta es la prima que paga efectivamente el asegurado y se compone de dos partes: la prima natural o pura por un lado y los gastos de explotación y la ganancia del asegurador por el otro. De esos gastos los más importantes son:

- Comisión a favor de los productores que colocan los seguros.
- Comisión de cobranza que se paga a los colaboradores por la percepción de las primas.
- Gastos de administración y propaganda.
- Recargo por fraccionamiento de la prima. La prima puede fraccionarse mediante cuotas periódicas, y ello da origen a un recargo, como suele ocurrir con las ventas a plazo.

Margen de seguridad. Se trata de un recargo para prever cualquier aumento de gastos y en particular la posibilidad de un riesgo mayor.

Prima nivelada: La aplicación simple de la prima natural para el cálculo de la prima comercial haría prohibitivo el seguro de vida, a partir de una determinada edad. En este caso la prima comercial aumentaría de continuo y llegaría un momento en que el asegurado desistiría del contrato dado el alto precio que debería abonar por su seguro.

Por ello ha sido necesario nivelar las primas a fin de que la prima comercial sea la misma, en los seguros de vida, durante toda la vigencia del contrato.

Prima única: es lo que debe abonar el asegurado cuando ello se hace en una sola oportunidad.

Primas periódicas: la prima única se abona con pagos parciales, con lo cual se ofrece al asegurado la posibilidad de decidir la concentración de estas operaciones.

1.2.5. Suma asegurada

Esta obligación constituye otro de los elementos necesarios del contrato de seguro, ya que sino se indica el contrato no surte efecto, resultando ineficaz de pleno derecho.

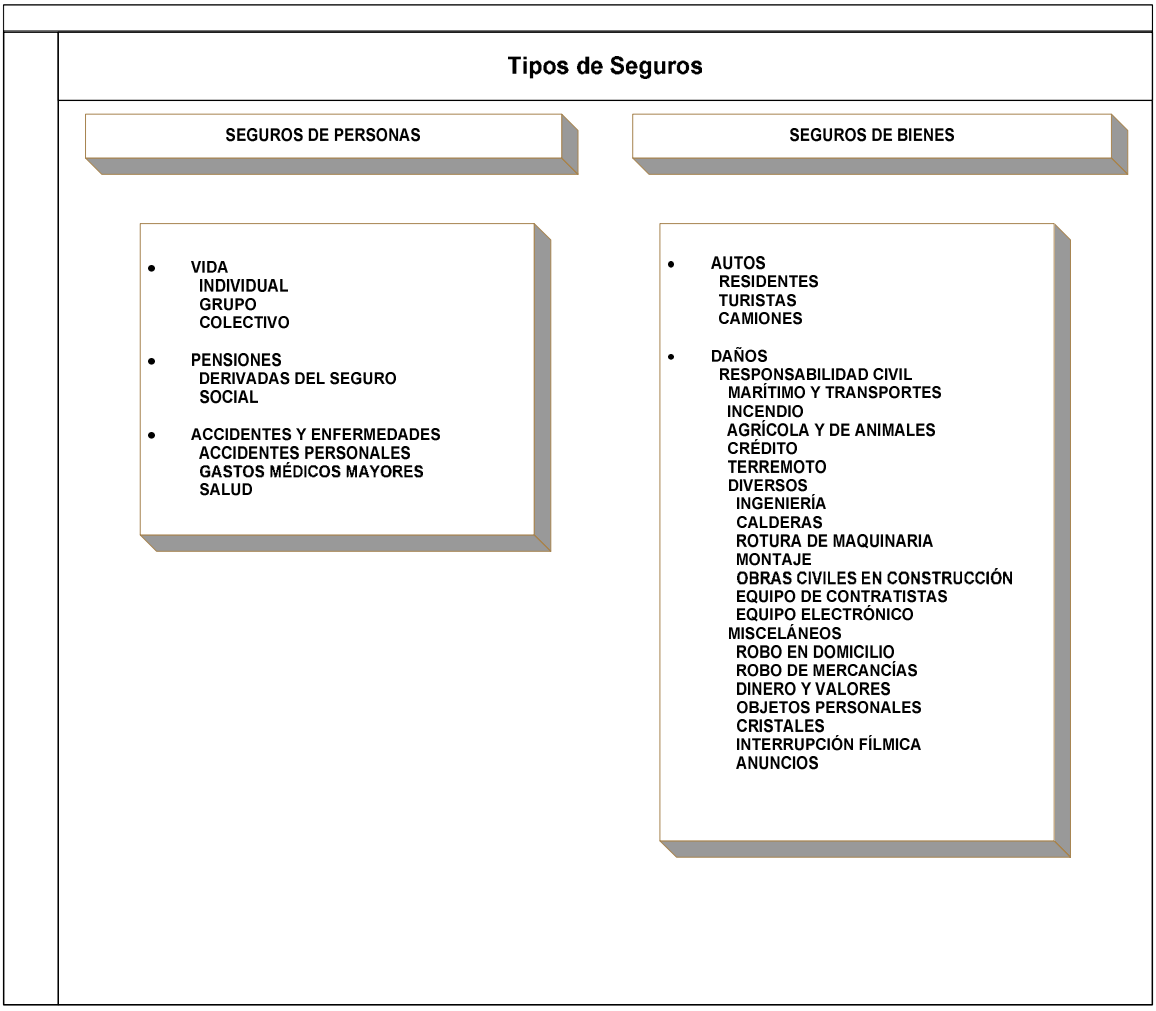
Este elemento resulta trascendente porque representa la causa de la obligación que asume el tomador de pagar la prima correspondiente. Debido a que este se obliga a pagar la prima porque aspira que el asegurador asuma el riesgo y cumpla con pagar la indemnización en caso de que el siniestro ocurra.

Esta obligación depende de la realización del riesgo asegurado. Esto no es sino consecuencia del deber del asegurador de asumir el riesgo asegurable. Y si bien puede no producirse el siniestro, ello no significa la falta del elemento esencial del seguro que ahora nos ocupa, por cuanto este se configura con la asunción del riesgo que hace el asegurador al celebrar el contrato, siendo exigible la prestación indemnizatoria sólo en caso de ocurrir el siniestro.

"La indemnización, es la contraprestación a cargo del asegurador de pagar la cantidad correspondiente al daño causado por el siniestro, en virtud de haber recibido la prima".

1.2.6. Tipos de seguro

Existen diversos tipos de seguros; sin embargo la Asociación Mexicana de Instituciones de Seguros (AMIS)[10] da la clasificación que se resume en el siguiente diagrama (Figura 1.1.):



En sentido estricto, es un seguro sobre la vida de una persona, aplicables en caso de muerte; en un sentido general, son aquellos seguros que cubren un acontecimiento que afecta la salud o integridad física de una persona. Este tipo de seguros tienen una subclasificación, puede aplicarse a vida individual, es decir cuando se cubre la vida de una sola persona; también tenemos aquellos que cubren la vida de un grupo de personas. Y por último tenemos los colectivos, que cubren un grupo mucho más amplio de personas.

Seguro colectivo.- aquel contrato de seguro sobre personas, que se caracteriza por cubrir mediante un solo contrato múltiples asegurados que integran una colectividad homogénea.

Seguro complementario.- aquel que se incorpora a otra con objeto de prestar a la persona asegurada en ambos una nueva garantía o ampliar la cobertura preexistente.

Seguro de accidentes.- aquel que tiene por objeto la prestación de indemnizaciones en caso de accidentes que motiven la muerte o incapacidad del asegurado, a causa de actividades previstas en la póliza.

Seguro de asistencia de viajes.- aquel seguro conducente a resolver las incidencias de diversa naturaleza que le hayan surgido durante un viaje.

Seguro de automóviles.- aquel que tiene por objeto la prestación de indemnizaciones derivadas de accidentes producidos a consecuencia de la circulación de vehículos.

Seguro de enfermedad.- es aquel en virtud, en caso de enfermedad del asegurado, se le entrega una indemnización prevista previamente en la póliza.

Seguro contra incendio.- aquel que garantiza al asegurado la entrega de la indemnización en caso de incendio de sus bienes determinados en la póliza o la reparación o resarcimiento de los mismos.

Seguro de orfandad.- aquel que tiene por objeto la concesión de una pensión temporal a favor de los hijos menores de 18 años en caso de fallecimiento del padre o de la madre de los que dependan económicamente.

Seguro de personas.- aquel que se caracteriza porque el objeto asegurado es la persona humana, tomando en cuenta su existencia, salud e integridad al pago de la prestación.

Seguro contra robos.- aquel en el que el asegurador se compromete a indemnizar al asegurado por las pérdidas sufridas a consecuencia de la desaparición de los objetos asegurados.

Seguro de transportes.- aquel por el que una entidad aseguradora se compromete al pago de determinadas indemnizaciones a consecuencia de los daños sobrevenidos durante el transporte de mercancías.

Seguro de vida.- es aquel en el que el pago por el asegurador de la cantidad estipulada en el contrato se hace dependiendo del fallecimiento o supervivencia del asegurado en una época determinada.

Capitulo 2

2. Data warehousing

El origen de los sistemas de soporte a las decisiones se remonta a los mismos inicios de la computadora y los sistemas de información, resulta curioso que los sistemas de soporte a las decisiones (Decision Support System, DSS), se hayan desarrollado fuera de la larga y compleja cadena evolutiva de la tecnología de la información.

En el inicio de la década de los 60's, el mundo de la computación consistía en crear aplicaciones individuales, que se ejecutaban utilizando archivos maestros, las aplicaciones efectuaban reportes que por lo general estaban desarrollados en COBOL. En esta etapa proliferaron los enormes archivos maestros con gran cantidad de datos redundantes, lo que supuso grandes dificultades para almacenar, sincronizar y actualizar los datos. Además de que los datos eran accedidos de manera secuencial.

Los 70's trajeron consigo nuevas tecnologías de almacenamiento y acceso a datos, con los dispositivos de almacenamiento de acceso directo (Direct Access Storage Device, DASD), ya no era necesario hacer una lectura de los registros 1 + 2 + 3… n, para leer el registro n+1. Esto trajo un nuevo tipo de sistema, los llamados manejadores de base de datos (Data Base Management System, DBMS). Estos sistemas facilitaban al programador el almacenar y acceder datos que residían en un DASD.

A mediados de los 70's el procesamiento de transacciones en línea (On-Line Transaction Processing, OLTP) hicieron aún más rápido el acceso a datos, abriendo todo un nuevo panorama para los negocios. La computadora podía ser usada para una diversidad de tareas que hasta ese entonces no eran posibles.

Para la década de los 80's, nuevas tecnologías como la computadora personal (Personal Computer, PC), y los leguajes de programación de cuarta generación (4GL's) lograron que el usuario final se convirtiera en una figura que tomaba el control de los datos y de los sistemas, un rol que antes sólo ocupaba el analista especializado. Con esto nacen los sistemas de información para la administración/gestión (Management Information System, MIS), los cuales son considerados los predecesores de los DSS, dado que soportaban decisiones de administración/gestión. Hasta este punto los datos y la tecnología sólo soportaban decisiones operacionales, y ninguna base de datos podía servir para el procesamiento transaccional de las operaciones de la empresa y para elaborar análisis estadístico al mismo tiempo.

En la década de los 90's los negocios aumentaron su complejidad, las corporaciones se extendían de manera global, y la competencia se tornó más agresiva; los ejecutivos y tomadores de decisiones comenzaron a tener una necesidad creciente de información, para mantenerse competitivos, mejorar sus servicios, y consecuentemente, aumentar su participación en el mercado.

Si bien los sistemas operacionales proveían información derivada de las operaciones diarias de la empresa, ésta no podría satisfacer a los ejecutivos, quienes requerían información que les llevara a tomar decisiones estratégicas. Ellos necesitaban saber, por ejemplo, qué producto tenía mayor aceptación, en qué lugar podría instalarse otra sucursal, etc. Por esta razón se comenzó a desarrollar un nuevo paradigma específicamente orientado a proveer información estratégica, y fue así como algunas compañías empezaron a tener logros sustanciales al desarrollar sistemas de almacenamiento de datos.

Un almacén de datos o Data Warehouse (DW) es un conjunto de procesos, una colección de datos orientados a darnos una descripción de las condiciones especificas del negocio, en un punto determinado del tiempo; estos datos deberán ser coherentes, integrados y no volátiles; en conjunto nos apoyan para tomar mejores decisiones, con información adecuada y oportuna.

2.1. Objetivos de un data warehouse

Antes de iniciar a detallar el proceso de modelado e implementación, es necesario centrarse en los objetivos primordiales que tiene un DW. En todos los casos, identificamos requerimientos recurrentes, que listaremos a continuación:

El DW debe hacer la información de fácil acceso. El contenido del DW debe ser entendible, los datos deberán ser intuitivos no sólo para el desarrollador, sino para los analistas de negocio. El que sea entendible implica que sea legible, es decir, que los datos sean etiquetados de manera realmente descriptiva. Los analistas de negocio, requieren separar información y luego combinarla, basándose en los criterios que dictan las reglas de negocio. También debemos tomar en cuenta que las herramientas de acceso a los datos que residen en el DW, deben ser fáciles de usar, y entregar resultados en tiempos de espera mínimos.

El DW debe contener información consistente. Los datos contenidos en el DW deben ser veraces, y cuidadosamente obtenidos de diversas fuentes dentro de la organización, vigilando la calidad y limpieza de estos datos. Es así como la información generada debe tener un único significado dentro de la organización, es decir, que las definiciones de los datos en el DW son comunes y están disponibles para todos los usuarios. Información consistente se traduce en información de calidad.

El DW debe ser evolutivo y flexible al cambio. Sabemos de sobra que el cambio es inevitable, las necesidades de los usuarios están sujetas a las dinámicas del mercado y a los adelantos tecnológicos. El diseño del DW debe prever estos cambios, los cuales no deben inhabilitar las aplicaciones existentes, y por supuesto no deberán alterar los datos al momento que las preguntas del negocio cambien.

El DW debe constituir un bastión que proteja los datos. La información de la empresa es invaluable y deberá ser resguardada. Cuando menos, el DW contiene información de quién, qué y en cuánto se venden los productos, lo que es potencialmente dañino en las manos equivocadas. Debe existir un control efectivo de acceso a la información confidencial.

El DW debe ser la base para tomar mejores decisiones. La información contenida en el DW deberá ser correcta y precisa, para dar un soporte efectivo en el proceso de toma de decisiones. Existe una y sólo una verdad una vez que el DW ha emitido sus resultados, y las decisiones que se tomen a parir de éstos determinarán el impacto y valor agregado al negocio.

Y finalmente, para que un DW sea exitoso y cumpla su cometido, éste deberá ganar aceptación dentro de la empresa, y comenzar a ser explotado máximo seis meses después de su liberación y habiendo impartido los cursos de capacitación adecuados. Pero más que otra cosa, la aceptación del sistema depende de la simplicidad de acceso que éste ofrezca.

Como se puede ver, el desarrollar un Data Warehouse exitoso requiere mucho más que ser un DBA o un técnico talentoso. Al tomar la iniciativa de desarrollar un DW tendremos, por un lado, la tecnología

de información (Information Technology, IT) y en el otro las demandas de los usuarios, con las que generalmente no estamos familiarizados; y deberemos aprender a avanzar conjuntamente con ambas.

2.2. Características de un data warehouse

Una vez que hemos revisado los objetivos primordiales de un DW, continuaremos con un listado de las características que lo definen. Basados en estas definiciones, entenderemos más a fondo la naturaleza de los datos que residen en un DW, así como hacernos conscientes de las diferencias entre los datos que residen en los sistemas operacionales, y aquellos que son almacenados en el DW.

2.2.1.1. Orientado a temas

Una de las principales características en un DW es la orientación a temas, es decir que los datos son clasificados en base a los aspectos de mayor interés para la organización. En contraste, los sistemas operacionales están organizados alrededor de las aplicaciones y los procesos, por ejemplo en el caso de la industria de seguros: pólizas emitidas, primas emitidas, contabilidad, siniestros ocurridos, reaseguro, reservas de suficiencia, entre otros.

En toda industria los conjuntos de datos están organizados alrededor de aplicaciones individuales, los sistemas operacionales administran y almacenan estos conjuntos de datos, de tal forma que sólo tienen información relacionada con la aplicación en particular. Por ejemplo, una aplicación de ingreso de pólizas puede acceder a datos sobre clientes, productos y tarifas. La base de datos combina estos elementos en una estructura que se adapta a las necesidades de la aplicación, aunque estos datos hayan sido generados a través de aplicaciones independientes.

En cambio un DW almacena los datos por temas, no por aplicación. Los temas difieren de industria a industria, por ejemplo, algunos de los temas críticos en el área de seguros son: finanzas, emisión, gestión de la cobranza, entre otros.

Para aclarar un poco más el concepto, revisemos el siguiente diagrama (Figura 2.1.):

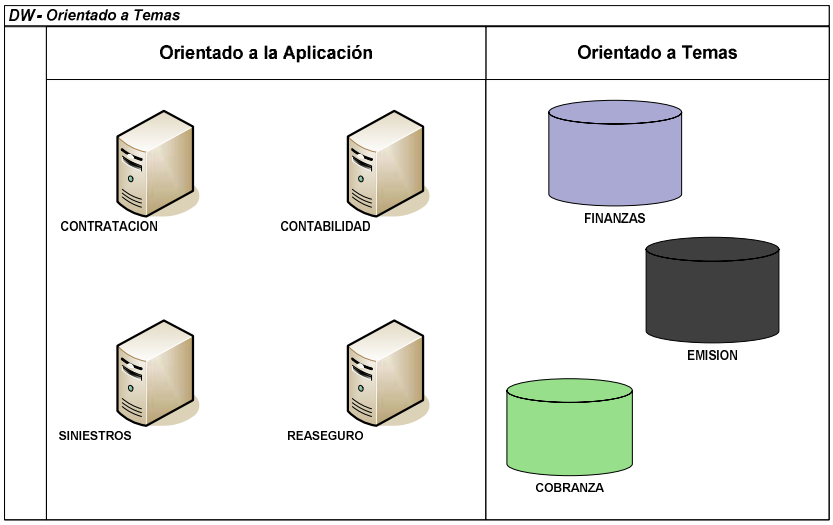


Figura 2.1. DW orientado a temas

Los sistemas operacionales están directamente relacionados con el diseño de la base de datos y el proceso; en cambio el DW está enfocado al modelado de los datos y el diseño de la base de datos.

Las principales diferencias entre la orientación a procesos y la orientación a temas, radican en el nivel de detalle que ambas presentan. Por ejemplo, la orientación a temas tiene un nivel de resumen, además contiene información que proviene de varios sistemas operacionales.

2.2.1.2. Integridad de datos

Otra característica importante que debe tener un DW, es la integridad de los datos que contiene. El DW reúne datos que provienen de diversas fuentes, que están distribuidas en diferentes bases de datos; las aplicaciones, plataformas y sistemas operacionales pueden ser distintos. La distribución de las tablas, archivos, nombres de campos, longitudes y tipos de datos, pueden ser dispares.

Por esta razón el DW se encarga de generar esquemas estandarizados, con definiciones comunes de nombres, tipos, longitudes y formatos de datos.

2.2.1.3. Datos variables en el tiempo

Un DW está diseñado para soportar el análisis y la toma de decisiones, por esta razón es necesario que almacene datos actuales e históricos, debe proveer al usuario con la información necesaria para realizar análisis comparativos, por ejemplo, si un analista de negocio revisa los patrones de compra de un cliente en específico, le interesará saber no sólo las compras que ha realizado actualmente, sino las compras que ha realizado en el pasado, o quizá dentro de un periodo específico de tiempo.

2.2.1.4. Datos no volátiles

Los datos que residen en el DW, no tienen como propósito soportar las operaciones diarias de la empresa, por esta razón el DW sólo tiene "fotografías" de los datos en periodos específicos de tiempo; no se hace necesario actualizar los datos en el DW cada vez que se genera una nueva transacción en el sistema operacional.

Estos datos del sistema operacional, se mueven hacia el DW en periodos determinados de tiempo, dependiendo de los requerimientos del negocio, estas cargas pueden ser diarias, cada dos días, cada semana o cada mes. De manera típica, los datos en el DW se refrescan a distintos intervalos, por ejemplo una base con los códigos de productos se mueve una vez por semana, en cambio una base con los datos de clientes se actualiza todos los días; esto obviamente obedece a la velocidad con que una base crece.

2.2.1.5. Granularidad

En los sistemas operacionales los datos son almacenados al nivel más bajo de detalle, por ejemplo, en un sistema de ventas para un centro comercial, los datos se almacenarán al nivel de transacción y producto vendido; si en algún momento dado requerimos saber cuál es el volumen de ventas en el periodo de un mes, para un producto determinado, realizamos una consulta filtramos los datos por día

y mes de la transacción, y finalmente sumamos las unidades vendidas; pero en un sistema operacional usualmente, no se almacenan datos sumarizados.

En cambio, dentro de un DW se almacenan los datos a diferentes niveles de sumarización; que van de lo general a lo particular. La granularidad se refiere al nivel de detalle que puede tener un DW, y cuando estamos hablando del nivel más fino de granularidad, nos estamos refiriendo al nivel más bajo de detalle, y viceversa.

2.2.2. Componentes de un data warehouse

Como hemos visto hasta ahora, el DW es en realidad un sistema de entrega de información. En este sistema se integran y transforman los datos que genera la empresa para convertirlos en información estratégica que apoye a la toma de decisiones.

El DW toma datos históricos desde uno o varios sistemas operacionales, así como datos externos que sean relevantes, posteriormente se llevan a cabo procesos de validación, limpieza, integración y derivación de estos datos. A continuación los datos, ya transformados en información, se cargan al repositorio DW corporativo, a partir del cuál se generan repositorios departamentales de información denominados Data Marts, en donde los usuarios podrán explotar esta información. En el siguiente diagrama (Figura 2.2.) se muestra la arquitectura de un DW:

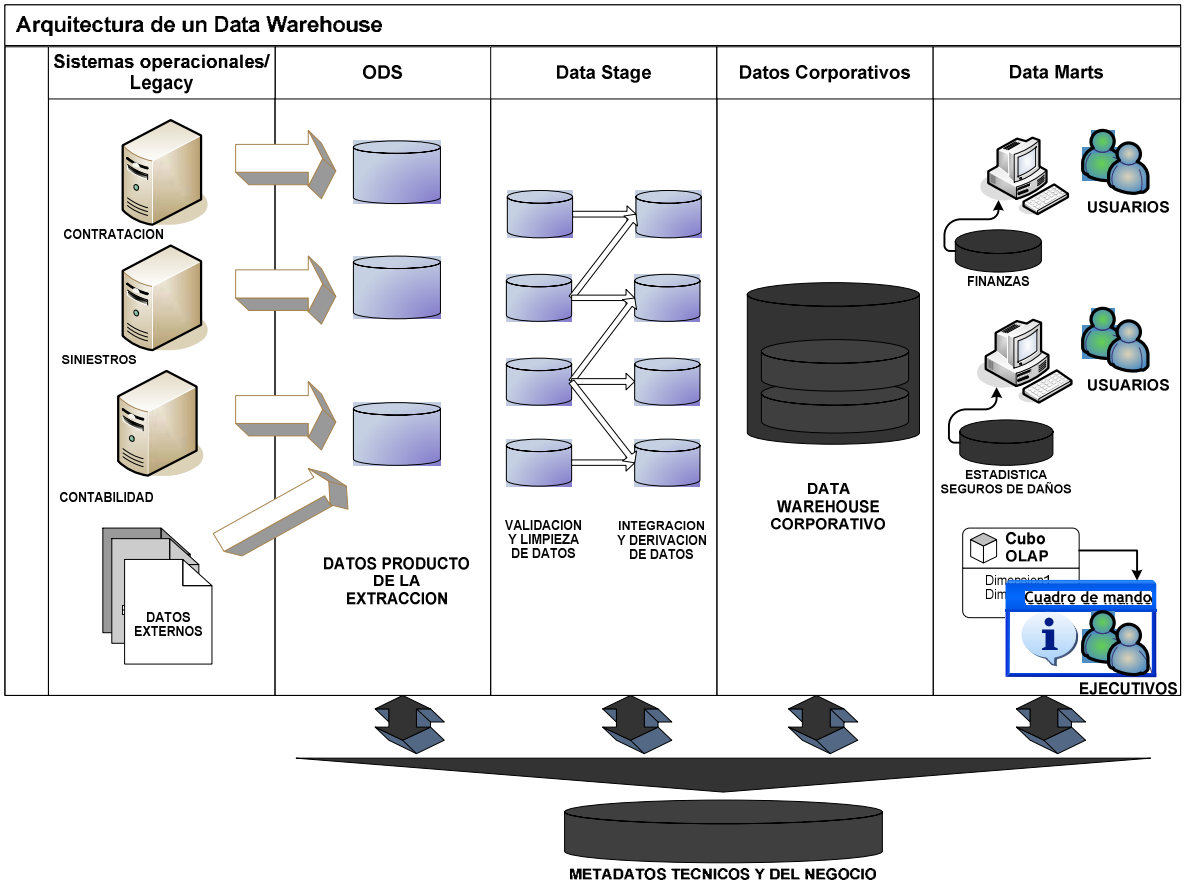


Figura 2.2. Arquitectura de un DW

2.2.2.1. Fuente de datos - sistemas operacionales/legacy

Prácticamente todo DW deberá ser alimentado con datos provenientes de los sistemas operacionales (OLTP), también se incluyen como fuentes posibles, archivos externos con información relevante. Pero ¿qué son los sistemas operacionales?

Son los sistemas que soportan la operación diaria de la empresa, se encargan de procesar transacciones, como pedidos, ingresos, pagos, etc. Por lo general, cada empresa utiliza una gran variedad de tecnologías y arquitecturas para soportar sus sistemas operacionales y pueden mezclar sistemas comerciales y sistemas desarrollados a la medida. Algunos de estos sistemas dentro de la empresa pueden entrar en la categoría de Legacy, esto significa que son sistemas heredados, de esquemas anteriores de la organización, y no están del todo preparados para soportar los nuevos requerimientos, y por tanto están iniciando un proceso de obsolescencia.

Los sistemas operacionales son estáticos por naturaleza, es decir, se modifican únicamente como respuesta a cambios en los procesos del negocio, o por razones técnicas como mejora en el desempeño o actualizaciones tecnológicas. Estos sistemas operacionales son la fuente de la mayoría de los datos administrados de forma electrónica dentro de la organización; y por esta razón son sistemas de soporte en tiempo real, optimizados en su desempeño.

Los datos que residen en los sistemas operacionales, pueden estar duplicados en varios de los sistemas, y de manera típica no están sincronizados. Estos sistemas representan el primer paso en la aplicación de las reglas de negocio a los datos de la empresa, y la calidad de estos datos tendrá un impacto directo en la calidad de la información utilizada dentro de la organización.

2.2.2.2. Repositorio operacional de datos – ODS

Recientemente el DW ha crecido en complejidad, y las necesidades actuales exigen que se encuentren disponibles datos con un nivel de detalle más bajo, para permitir análisis más minuciosos, tanto en tendencias del mercado, como en cambios operacionales que impactan a la organización. Es en este punto se hace necesario introducir un nuevo componente arquitectónico del DW: El repositorio operacional de datos (Operacional Data Store, ODS).

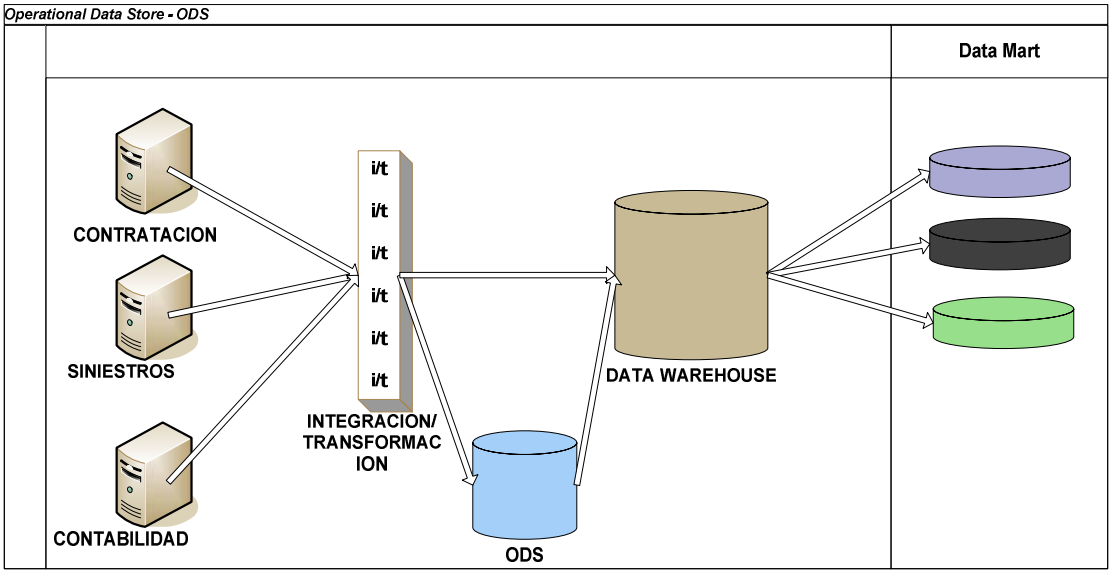
Es un almacén de datos orientado a temas, integrado, volátil, con datos actuales, y que contiene únicamente datos corporativos operacionales a nivel detalle[11].

Debido a la granularidad de los datos residentes en este repositorio, se requiere de una gran cantidad de espacio de almacenamiento, consecuentemente estos datos son volátiles, es decir, no almacenan historia, son actualizados continuamente. El tiempo que estos datos residen en este repositorio, depende de las necesidades de la organización, puede ser diario, semanal o mensual.

En el siguiente diagrama (Figura 2.3.) podemos ver que el ODS se alimenta de programas de integración y transformación (i/t), los cuales pueden ser o no los mismos programas que alimentan directamente al DW, y el ODS a su vez, puede alimentar al DW.

[11] Bill Inmon & Claudia Imhoff, Building the Operational Data Store. John Wiley & Sons, U.S.A. 1996.

Este componente cumple doble función, por una parte, el ODS es completamente operacional; tiene una gran disponibilidad, así como una excelente velocidad de respuesta en consultas. Por otro lado el ODS tiene características que muy claramente se ajustan a la definición de un DSS.



En el siguiente diagrama (Figura 2.4.) podemos observar la diferencia entre un datamart independiente y uno dependiente.

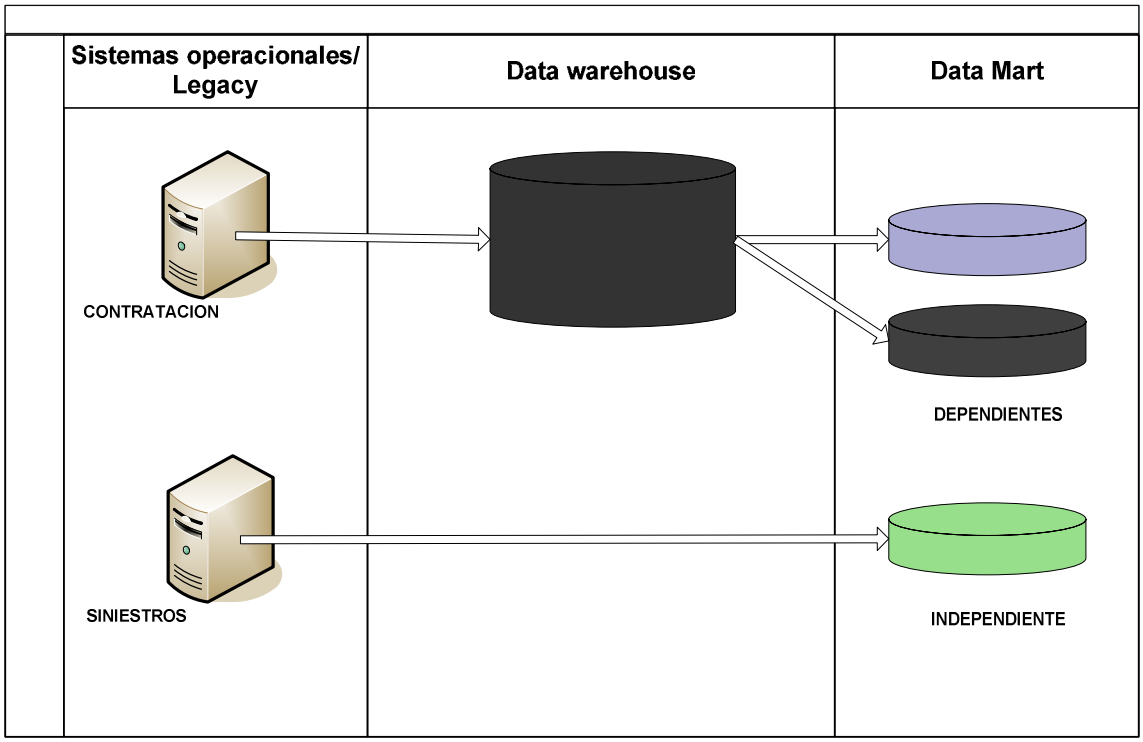


Figura 2.4. Tipos de Data Mart

Tabla comparativa DW vs. Data Mart

Data warehouse	Data mart
Corporativo	Departamental
Unión de varios data marts	Centrado en un solo proceso de negocio
Datos recibidos desde el área stage	Proveniente de tablas de hechos y dimensiones
Estructura para soportar la visión corporativa de los datos	Estructura para soportar la visión departamental de los datos
Organizado bajo el modelo E-R	

2.2.2.5. Metadatos

Este elemento de arquitectura, es de vital importancia debido a que coordina los servicios y actividades dentro del DW, así mismo controla las transformaciones y transferencia de los datos hacia el repositorio; por si fuera poco, también este componente se encarga de moderar la forma en cómo los datos son presentados a los usuarios finales.

Los metadatos dentro del DW cumplen una función similar a la de un diccionario, contiene información acerca de las estructuras lógicas de los datos, direcciones físicas, índices, etc. En resumen, contiene datos sobre los datos que residen en el DW. Incluso los componentes de control y mantenimiento dentro del DW, deben interactuar con este

Los metadatos dentro de un DW, pueden clasificarse en tres diferentes categorías:

- Metadatos operacionales
- Metadatos de extracción y transformación
- Metadatos para usuario final

Metadatos operacionales. Como ya sabemos, los datos en el DW provienen de diversos sistemas operacionales dentro de la organización. Estos sistemas fuente tienen diferentes estructuras de datos, es decir que los campos pueden tener diferentes nombres, tipos de dato e incluso pueden diferir en longitud. En el proceso llevado a cabo para entregar la información al usuario final, se transforman, dividen, concatenan los datos. Al momento de la entrega, deberemos ser capaces de responder de dónde es que provienen los datos. Los metadatos operacionales son los que llevan este registro.

Metadatos de extracción y transformación. Metadatos que contienen información respecto a las fuentes, métodos y frecuencia de extracción, así como las reglas de negocio aplicadas a los mismos. También en esta categoría de metadatos se lleva el registro de todas las transformaciones realizadas en el área stage.

Metadatos para usuario final. Este tipo de metadato constituye el mapa de navegación dentro del DW, permite al usuario localizar la información que busca, así como asignar su propia terminología; lo que facilita el proceso análisis.

2.2.2.6. Herramientas de acceso a datos

Este componente es la interfase que el usuario utilizará, para explotar los datos que residen en el DW, sin embargo, es muy importante que al inicio, se establezca el alcance planeado para el DW; así como tener una clasificación de los usuarios recurrentes que tendremos realizando consultas.

Por ejemplo tendremos usuarios novatos, que sin entrenamiento necesitarán de reportes fijos y consultas pre-armadas. El usuario casual es aquel que sólo necesita información de manera poco frecuente, de cualquier forma este usuario también requiere información preempaquetada.

Por otro lado tenemos al analista de negocio, que requiere una herramienta que le permita realizar consultas complejas y análisis de la información residente en el DW.

Finalmente tenemos al Súper usuario (Power User), quien necesitará navegar a través de los datos en el DW, seleccionar aquellos que le sean relevantes, generar consultas personalizadas y reportes a la medida.

Para lograr proveer la información adecuada, a la comunidad de usuarios del DW debemos integrar varios métodos de entrega de información. En el siguiente diagrama (Figura 2.5.) podemos observar cada uno de los métodos de entrega de información.

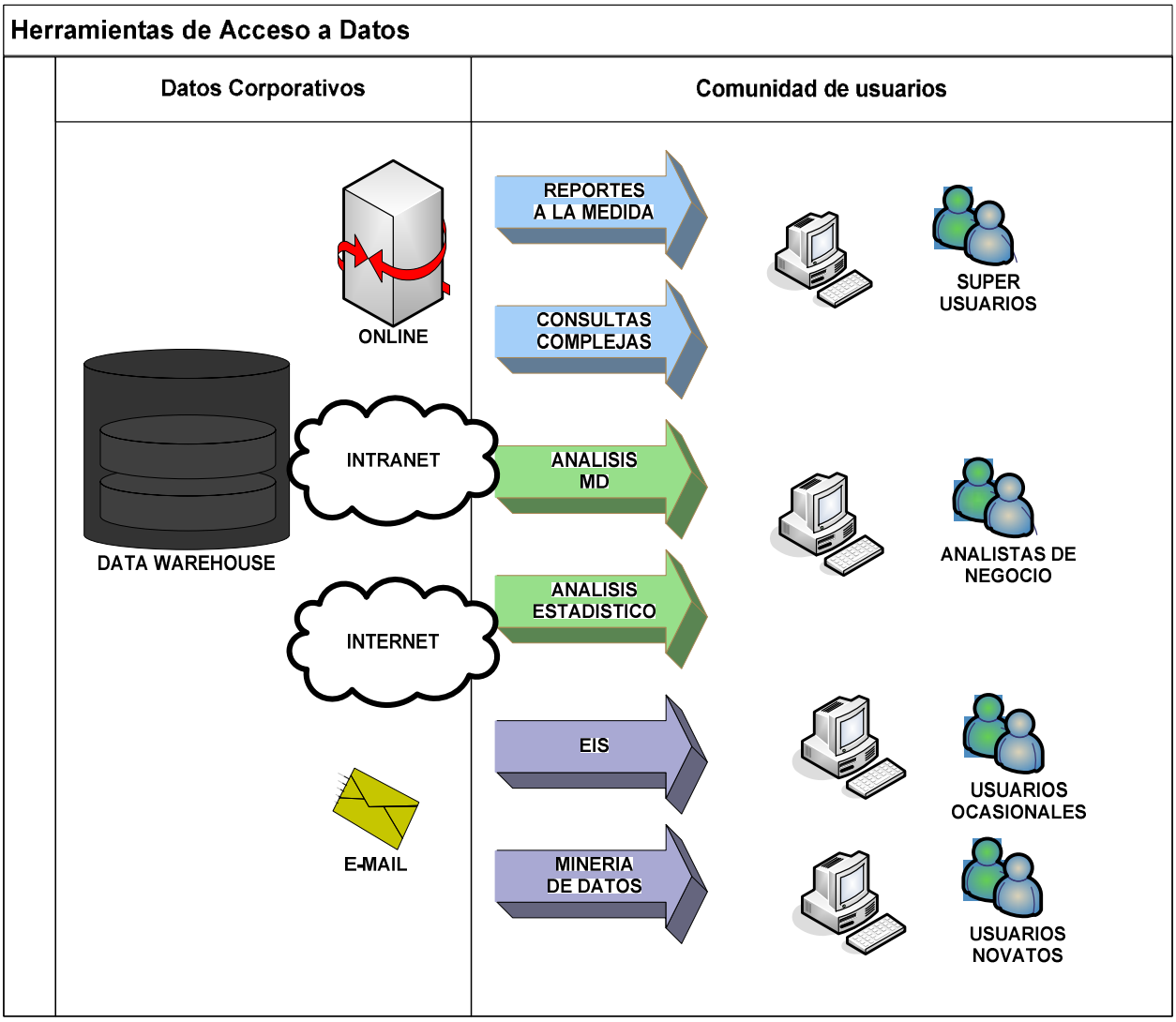


Figura 2.5. Herramientas de acceso a datos

Como podemos observar en el diagrama, tendremos reportes a la medida, que básicamente están planeados para usuarios novatos u ocasionales. Por otra parte las consultas complejas, el análisis multidimensional (MD) y análisis estadístico están disponibles para los analistas de negocio y super usuarios. Finalmente los sistemas de información ejecutivos (Executive Information System, EIS) están disponibles para ejecutivos de alto nivel y gerentes.

Algunos DW tienen disponibles aplicaciones de minería de datos (Data Mining, DM), estas aplicaciones se clasifican como sistemas para descubrimiento de conocimiento; debido a que los algoritmos de minería ayudan al usuario a realizar análisis estadísticos, descubrir patrones y tendencias a través de los datos residentes en el DW.

Entre estos algoritmos podemos encontrar el de Bayes, que nos muestra cómo los modelos de probabilidad pueden ser aplicados conjuntamente con los métodos de cadenas de Markov y Monte Carlo.

2.3. Desarrollando un modelo de datos

A este punto ya hemos revisado los objetivos del DW, así como las características que lo definen; ahora podemos pasar al desarrollo de un modelo de datos.

2.3.1. Metodología

El modelo de datos para un DW se desarrolla en ocho pasos, que son:

- Seleccionar los datos de interés
- Agregar el factor tiempo a la llave
- Agregar derivaciones de datos
- Determinar el nivel de granularidad
- Sumarizar los datos
- Fusionar entidades
- Crear arreglos
- Segregar datos

Los ocho pasos pueden ser agrupados en dos categorías, los primeros cuatro tienen que ver con las necesidades del negocio; primeramente seleccionamos los datos basándonos en las necesidades del negocio, se agrega la variable tiempo para dar soporte a los datos históricos, para mayor consistencia se generan datos derivados, y el nivel de granularidad se determina para asegurar que los datos cumplen con los requerimientos.

Una vez que estos pasos se han completado, el DW deberá ser capaz de satisfacer los objetivos del negocio.

Los siguientes cuatro pasos tienen que ver con el desempeño, ya que al sumarizar los datos ayuda a reducir los tiempo de entrega de resultados, el fusionar entidades nos evitan el tener que hacer uniones (joins) de datos que sean usados juntos, los arreglos nos facilitan la creación de data marts, y finalmente el segregar datos nos ayuda a mantener la estabilidad (disminuir la cantidad de registros que son agregados al DW), minimizando el espacio de almacenamiento requerido.

Ya aplicados estos pasos, se pueden tener pasos adicionales para afinar el desempeño, estas actividades incluyen la denormalización y particionamiento.

2.3.1.1. Seleccionando los datos de interés

El primer paso, en el desarrollo del modelo de datos es seleccionar los datos que sean relevantes, y existen dos principales motivos por las que este paso es el primero en ejecutarse. Primeramente colocan los objetivos y propósitos del negocio en primer plano, todas las decisiones que conciernen al modelo del DW tienen una íntima relación con los objetivos del negocio. Posteriormente, esta paso nos ayuda a acotar los alcances del proyecto del DW, funciona como un embudo que sólo dejará pasar aquellos datos que en realidad son necesarios.

Los insumos necesarios para llevar a acabo este primer paso de la metodología, se definen a continuación:

El modelo de datos es sólo uno de los insumos necesarios para el primer paso, otras entradas son el documento de alcance para el proyecto, requerimientos de información, prototipos, consultas y reportes existentes y modelos físicos de los sistemas que serán las fuentes de información para el DW.

Una de las reglas es que, no todos los datos requeridos pueden preverse con antelación, por ello utilizar el modelo relacional dentro del DW nos trae una ventaja, nos permite agregar datos sin que esto implique grandes modificaciones en los procesos que ya están desarrollados.

Modelo de Datos del Negocio

Un modelo de datos del negocio completo, constituye un inventario en el cual nosotros podremos consultar los elementos de datos disponibles. Cuando no existe un modelo, el equipo de desarrollo debe generarlo, asegurándonos de que éste cubre sólo los requerimientos que están en el alcance de esta etapa del proyecto. Todos los elementos de datos incluidos en este modelo, estarán presentes en el DW, una vez que éste último esté terminado.

En el siguiente diagrama (Figura 2.6.) podemos observar el modelo de datos entidad-relación a nivel lógico, para una compañía de seguros, este modelo se muestran las entidades y relaciones requeridas para la emisión de pólizas:

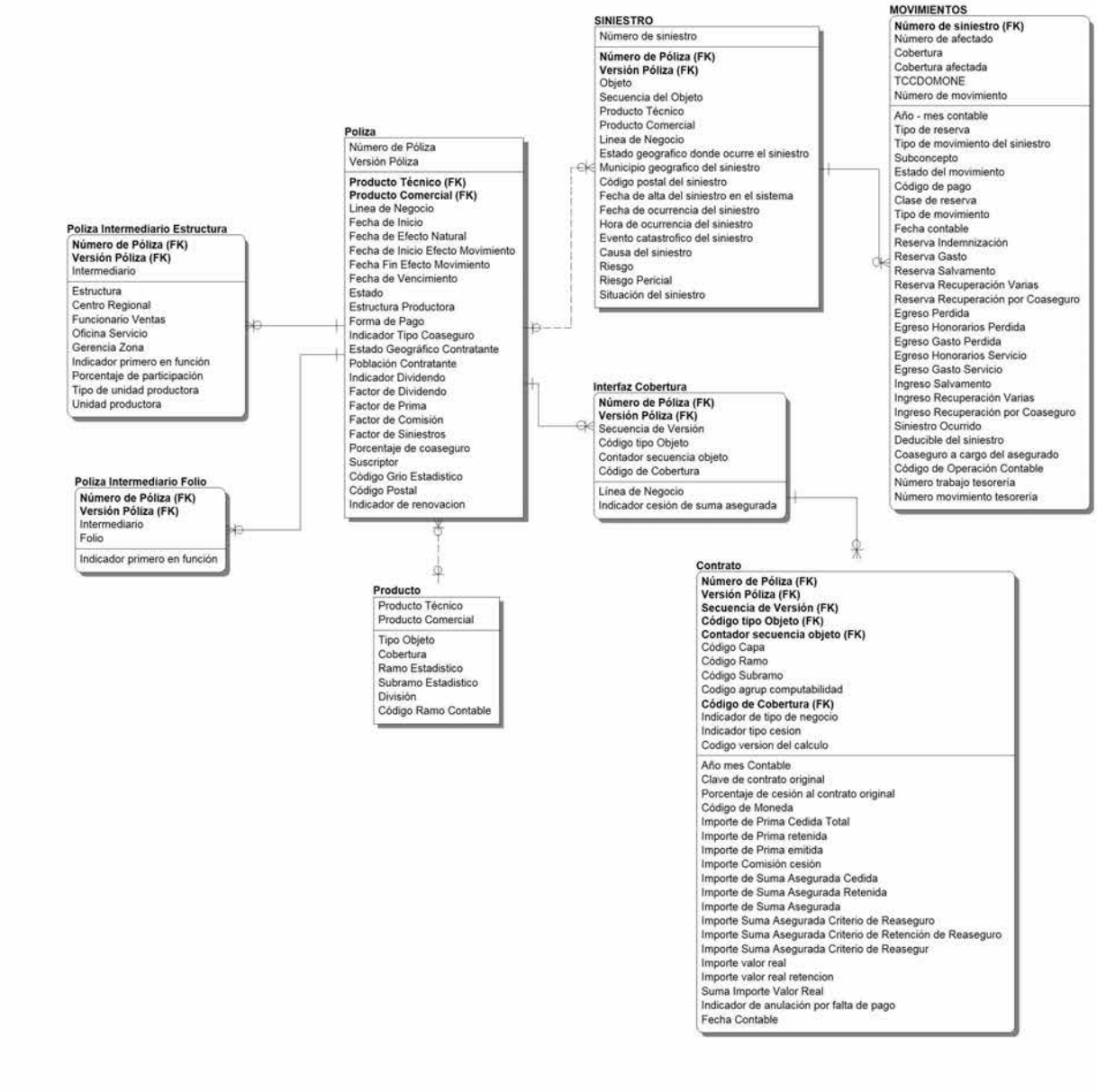


Figura 2.6. Modelo Entidad Relación

Documento de Alcance

El documento de alcance para el proyecto, establece las expectativas para esta primera fase (iteración) en el desarrollo del DW. Además incluye información de qué datos serán incluidos, así como una sección que detalla aquellos datos que han sido segregados en esta iteración.

Requerimientos de Información

Estos requerimientos constituyen una más de las entradas, para el primer paso en la metodología de desarrollo, y existen varias fuentes para estos requerimientos. Dado que el DW debe alinearse con las metas del negocio, resulta muy útil hacer una revisión de documentos como el plan corporativo y el reporte anual. También es necesario llevar a cabo reuniones con ejecutivos, analistas del negocio, y usuarios finales.

En estas reuniones podremos identificar las principales preguntas sobre el negocio, los elementos de datos necesarios para dar respuesta a dichas preguntas, así como la forma en la que serán usados los datos.

El trabajo conjunto con analistas de negocio, que requieren acceso al análisis dimensional de los datos, es una buena oportunidad para no sólo determinar los datos requeridos, sino los data marts que pueden construirse a partir de estos datos. También es muy útil determinar los niveles de agregación que estos datos deberán presentar.

Consultas y Reportes Existentes

Las consultas y reportes constituyen otra fuente de información, de los requerimientos de datos para negocio, sin embargo deben utilizarse cuidadosamente, ya que el tener una consulta o un reporte, no es garantía de que en realidad alguien lo requiera. Esto sucede porque en ocasiones, los reportes son hechos para satisfacer necesidades específicas que con el tiempo dejan de existir, y el problema es que nadie se toma la molestia de quitar estos reportes de la planificación de las ejecuciones batch.

De forma similar, incluso los reportes en uso, pueden contener datos que ya no son necesarios, esto ocurre porque dichos datos fueron agregados, sólo por si acaso, o porque en algún momento si era utilizado, pero las condiciones del negocio cambiaron, descontinuando el uso de dichos campos.

Prototipos

Probablemente la forma más efectiva que tenemos, para identificar los elementos requeridos es crear un prototipo, que deberá alimentarse con datos actuales; esto ayudará al usuario a visualizar el resultado final del desarrollo, así como ayudarle a articular mejor los requerimientos de datos.

Es muy importante tener una retroalimentación adecuada con el usuario en esta fase, sin embargo no debemos perder de vista que este proceso no puede prolongarse por mucho, dado que el objetivo de esta fase es afinar detalles sobre los requerimientos, no proveer al usuario de manera temprana, con un entregable productivo.

Fuentes de datos

La estructura de datos de los sistemas operacionales, que serán la fuente de datos para el DW, nos proveen con información de cómo y dónde son almacenados físicamente estos datos, también nos dan información acerca de elementos que están relacionados con los datos que han sido solicitados, lo que nos da una pauta para preguntar al usuario si estos datos adicionales con requeridos.

2.3.1.2. Agregando el factor tiempo a la llave

Dentro de los sistemas operacionales, el modelo de datos sólo retrata el negocio en forma presente. En cambio el DW retrata el negocio, desde una perspectiva histórica, ya que el DW es variante en el tiempo (tiene una serie de "fotografías" del negocio a través del tiempo). Por este motivo el modelo de datos Entidad-Relación (Entity Relationship, E-R) es utilizado, ya que en este modelo, es posible lograr una perspectiva histórica en los datos, con sólo agregar el elemento fecha a la llave.

Este modelo responde bien al cambio, haciéndolo transparente para el usuario final, no tenemos que preocuparnos por consultas de alta complejidad, que pudieran generarse. Otra ventaja más, es el poder generar una dimensión con la vista más reciente de la jerarquía de datos, o bien, una dimensión de fechas para lograr la perspectiva histórica; las necesidades del usuario determinarán si generamos uno u otra.

En el modelo dimensional, varias entidades pueden unirse para crear una jerarquía, además si agregamos la fecha a esta dimensión, tendremos como resultado una dimensión que cambiará cuando cualquiera de los datos dentro de la jerarquía cambie.

También es recomendable utilizar llaves sustitutas para las dimensiones que sean generadas; el uso de este tipo de llave es conveniente por dos razones: en primer lugar logramos tener un identificador único para tablas que requieren llaves compuestas de varios campos, y en segundo lugar nos ayuda a generar llaves más compactas al momento de trabajar con tablas de hechos; consecuentemente facilita la generación de consultas para el usuario final.

2.3.1.3. Agregando los datos derivados

Los datos derivados son aquellos que resultan de aplicar operaciones aritméticas a dos o más elementos de datos; estos datos son incorporados al DW por dos principales razones: para asegurar la consistencia de los datos y para mejorar el desempeño.

Uno de los objetivos comunes del DW es entregar la información en una forma que permita que todo mundo tenga los mismos hechos (también conocidos como métricas), y además, que tengan el mismo significado para todos dentro de la organización.

Para el caso de seguros, el campo *Prima Total*, podría tener cualquier número de interpretaciones; sin embargo, es muy importante que todos en la organización conozcan el número de elementos utilizados para calcularlo, por ejemplo, la prima técnica, descuentos especiales, recargos por pago fraccionado, impuesto y comisiones sobre prima, pueden ser incluidos en dicho cálculo.

Básicamente existen dos métodos para generar los datos derivados: uno es realizar los cálculos al momento de la carga de datos, lo que nos ayuda a disminuir el costo en procesador, especialmente si estos datos derivados son utilizados en varios data marts; es decir, el cálculo se realiza sólo una vez y es aprovechado por todos los departamentos que lo requieran.

Otro método es generarlo en tiempo de entrega, es decir, al momento que el usuario está accediendo a una consulta. Esto puede darle mayor flexibilidad al usuario, cuando estamos desarrollando data marts distribuidos, porque todos los usuarios retienen las mismas definiciones y algoritmos para la derivación.

2.3.1.4. Determinando la granularidad

La granularidad o nivel de detalle, es de suma importancia desde el punto vista técnico, de negocio y del proyecto. Cuando nos referimos al aspecto de negocio, dicta la potencialidad, capacidad y flexibilidad que tendrá el DW para responder a las preguntas de negocio que surjan. Desde el punto de vista técnico determina el tamaño, alcance, costo de operación y desempeño del DW.

Finalmente, cuando hablamos desde la perspectiva del proyecto, determinará el esfuerzo necesario para desarrollar el DW, entre mayor sea el nivel de detalle, se requerirá más tiempo y esfuerzo para su construcción, ya que los desarrolladores tendrán que lidiar con más entidades, y relaciones más complejas entre éstas. Sin embargo, debemos tener en cuenta que cuando el nivel de detalle sea mayor, el DW también será muy grande, y se necesitarán consideraciones técnicas adicionales para su construcción.

Existen diversos factores que influyen en el nivel de detalle del DW:

Necesidades actuales del negocio. Este es el principal de los factores, dado que el nivel de granularidad debe ser suficiente para responder a todas y cada una de las preguntas del negocio, las cuales deberán estar de acuerdo a la fase de desarrollo en la que se encuentra el DW.

El añadir un mayor nivel de granularidad, aumenta los costos en el desarrollo del DW, y si las necesidades del negocio no incluyen tener el detalle de información, no se justifica invertir más tiempo y dinero si no agrega ningún valor.

Anticipando las necesidades del negocio. Las necesidades futuras del negocio también deberán ser consideradas, la palabra clave es considerar antes de incluir, es muy importante sostener entrevistas con los usuarios para asegurarnos de la percepción a futuro que tienen del negocio.

Se recomienda alternar entre construir el DW a partir de los datos que sabemos serán requeridos, y diseñar, así como extraer datos previendo los requerimientos futuros.

Datos derivados. Dado que los datos derivados, dependen de otros datos, es importante considerar todos los elementos necesarios para realizar los cálculos, aunque esto represente un costo de almacenamiento.

Granularidad del sistema operacional. Otro factor que impacta al nivel de detalle de los datos almacenados en el DW, es sin duda el nivel de granularidad con el que cuenta la fuente de datos; es tan simple como esto: si la fuente de datos no lo tiene, el DW tampoco.

De primera vista esto parece muy obvio, sin embargo existen casos en los cuáles existen varios sistemas operacionales desde los que se extrae información; es posible que el nivel de agregación entre éstos difiera, es aquí cuando el equipo de desarrollo del DW, debe determinar si se extrae del sistema que tenga el nivel de detalle más bajo, o extraer desde cada uno de los sistemas con el nivel de detalle que cada uno tenga disponible.

2.3.1.5. Sumarizando los datos

El siguiente paso al desarrollar el modelo Data Warehouse (DW), es crear datos sumarizados o datos resumidos. La generación de datos sumarizados, no necesariamente significará un ahorro de espacio

en disco, ya que los datos de detalle usados para generar estos datos resumidos, generalmente se mantienen almacenados. Sin embargo, el uso de datos resumidos, sí mejora el desempeño del proceso de entrega de la información, ya que reduce los requerimientos de almacenamiento online (los datos a nivel detalle pueden estar almacenados en dispositivos secundarios).

El criterio de sumarización más común es el tiempo, ya que el DW, representa los datos ya sea en un punto específico del tiempo, o bien, a lo largo de un periodo de tiempo. Por ejemplo el número de pólizas emitidas en un día específico, o el número de pólizas emitidas a lo largo de un mes.

Los principales tipos de sumarización se detallan a continuación:

Sumarización por un periodo de tiempo

Un acumulado simple, resume los datos en base a uno solo de sus atributos, como puede ser el tiempo. Por ejemplo, puede ser el volumen de venta de pólizas en un día, asociado los datos que son relevantes para los analistas de negocio, es decir, por vendedor o por tipo de cobertura.

En el siguiente ejemplo, podemos observar como se resumen o sumarizan los datos en base a la fecha de emisión de la póliza:

Fecha de Emisión de Póliza	Número de Póliza	Cobertura	Vendedor	Importe de Prima
01-Ene-06	126890	Incendio Edif.	Rafael Olvera	2,468.00
01-Ene-06	129456	Incendio Edif.	Rafael Olvera	3,670.00
02-Ene-06	129980	Terremoto Edif.	Claudia Ochoa	5,460.00
03-Ene-06	131000	Terremoto Edif.	Andrés Marquez	4,200.00
03-Ene-06	131200	Terremoto Edif.	Andrés Marquez	5,150.00

Fecha de Emisión de Póliza	Cobertura	Vendedor	Importe de Prima
01-Ene-06	Incendio Edif.	Rafael Olvera	6,138.00
02-Ene-06	Terremoto Edif.	Claudia Ochoa	5,460.00
03-Ene-06	Terremoto Edif.	Andrés Marquez	9,350.00

Otro tipo de acumulación puede llevarse a cabo utilizando un rango de fechas, de tal forma que deberemos contar con una fecha inicial y una final, para llevar a cabo la sumarización.

Sumarización por episodios

En este tipo de sumarización, también podemos emplear el acumulado simple y el acumulado por rangos, sin embargo la diferencia es que se operan sobre datos que son episódicos; es decir, son datos que presentan la situación en un punto determinado del tiempo. Como ejemplo, podemos mencionar información del volumen de ventas, que se reporta a intervalos regulares de tiempo; ya sea de forma semanal, mensual, bimestral, semestral, etc.

Sumarización Vertical

El último tipo de sumarización es aplicable tanto a la sumarización por periodo de tiempo, como a la sumarización por episodios. Sin embargo la sumarización no siempre es útil, y deberá ponerse especial atención en los resultados que éste proceso produzca, ya que podríamos obtener información que desoriente a los usuarios.

2.3.1.6. Fusionando entidades

El fusionar entidades es útil, ya que mejora el rendimiento del proceso de entrega de información, al reducir el número de uniones requeridas, para obtener el resultado deseado en una consulta; aún cuando las entidades originales se mantengan almacenadas. Una ventaja más es que mejora la consistencia de la información.

2.3.1.7. Creación de arreglos

La creación de arreglos no es comúnmente usada, sin embargo puede ser útil cuando trabajamos con datos que son agrupados.

2.3.1.8. Segregando datos

A diferencia del DW, los sistemas operacionales y los modelos de datos para negocios, generalmente no mantienen datos históricos almacenados. Esto significa que cada vez que un atributo en alguna entidad, cambia en valor, se genera un nuevo registro.

2.4. Requerimientos de información y análisis dimensional

La definición de requerimientos, es de suma importancia para el diseño de un modelo de datos apropiado para el Data Warehouse. El diseño de datos debe ser dividido en lógico y físico; el modelo lógico sirve para determinar los elementos de datos que serán utilizados, así como las combinaciones de éstos para formar estructuras de datos. El modelo físico, nos es de utilidad para el momento de llevar a cabo la construcción del DW.

La construcción de un Data Warehouse es diferente, en varias formas, a construir un sistema operacional. Esto se hace mucho más evidente en la fase de requerimientos. Debido a estas diferencias, los métodos tradicionales de realizar levantamientos de información, que trabajan muy bien para diseño de sistemas operacionales; no pueden ser aplicados al Data Warehousing.

Sin embargo no estamos solos en la oscuridad, a pesar del hecho de que los usuarios no pueden describir por completo qué es lo que quieren integrar al Data Warehouse, pueden darnos importantes pistas de cómo ven ellos el negocio; pueden decirnos qué métricas son relevantes para el trabajo que desarrollan diariamente. Adicionalmente, podemos guiarnos en las preguntas más frecuentes que efectúan los ejecutivos dentro de la organización; a éstas preguntas las llamamos “preguntas del negocio”.

Por ejemplo un director de área, frecuentemente pregunta ¿cuánto ha generado un nuevo producto? Además para darle más sentido a la información, ésta debería estar dividida por mes, zona geográfica, por oficina de ventas, y además con un comparativo respecto a lo planeado.

Para un gerente de marketing, la pregunta sería ¿cuáles son las estadísticas de ventas? sumarizadas por producto, categoría, además de forma diaria, semanal, mensual y por canal de distribución.

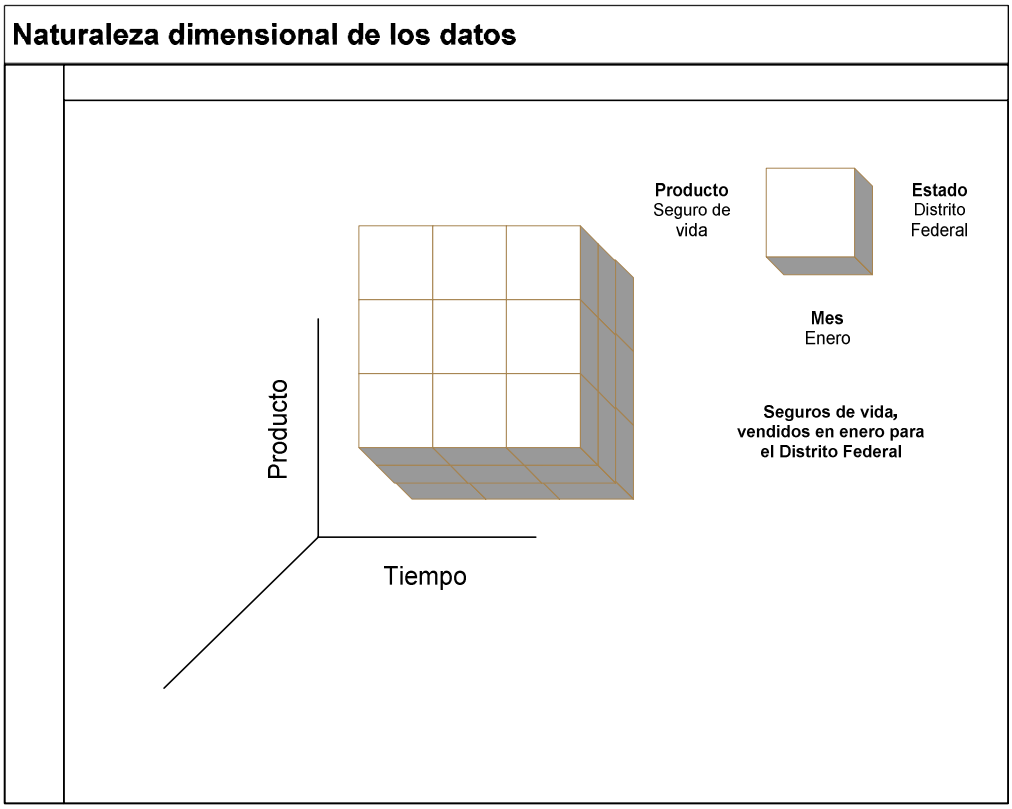
Finalmente para un supervisor financiero, la pregunta giraría en torno a las ventas actuales vs. el presupuesto, divididas en mes, trimestre y anual; divididas en línea de negocio, división y a total de compañía.

Estas son el tipo de preguntas que los ejecutivos y gerentes de una organización desean responder antes de poder tomar una decisión; pero existe una razón para ello. Por ejemplo el director de área hace esas preguntas, porque está interesado en saber cuál es la ganancia generada por el lanzamiento del nuevo producto; pero no está interesado en un importe solamente; le interesa saber cómo se ha vendido en cada uno de los meses, además esto dividido en las diferentes zonas geográficas, y en cada una de las oficinas de ventas. Finalmente le interesa saber cómo van las ventas con respecto a lo planeado, en caso de una desviación respecto al plan, podría tomar una decisión correctiva a tiempo y disminuir el impacto negativo de dicha tendencia.

De forma similar ocurre con el gerente de marketing y con el supervisor financiero; desean ver la información dividida en diferentes cortes; esto les ayuda a tener una visión más completa y organizada de lo que está sucediendo en el negocio; es por ello que a estos cortes les llamaremos dimensiones.

Si los usuarios de la información, piensan en términos de dimensiones para la toma de decisiones, nosotros también deberemos hacerlo, a la hora de colectar los diferentes requerimientos. A pesar del hecho que el uso propuesto para el DW, es todavía poco claro, las dimensiones utilizadas por los usuarios no son del todo nebulosas.

Veamos un ejemplo, que nos ayudará a entender mejor la naturaleza dimensional de los datos:



En el diagrama (Figura 2.7.) se muestra el análisis de ventas de seguros de vida, dividido en tres dimensiones de negocio: producto, geografía y tiempo. Al generar un diagrama de éstas dimensiones, en un sistema cartesiano de tres dimensiones, obtendremos cubos de información; en los cuáles podremos encontrar las unidades vendidas para un periodo de tiempo determinado (enero), un producto (seguro de vida) y una determinada división geográfica (estado).

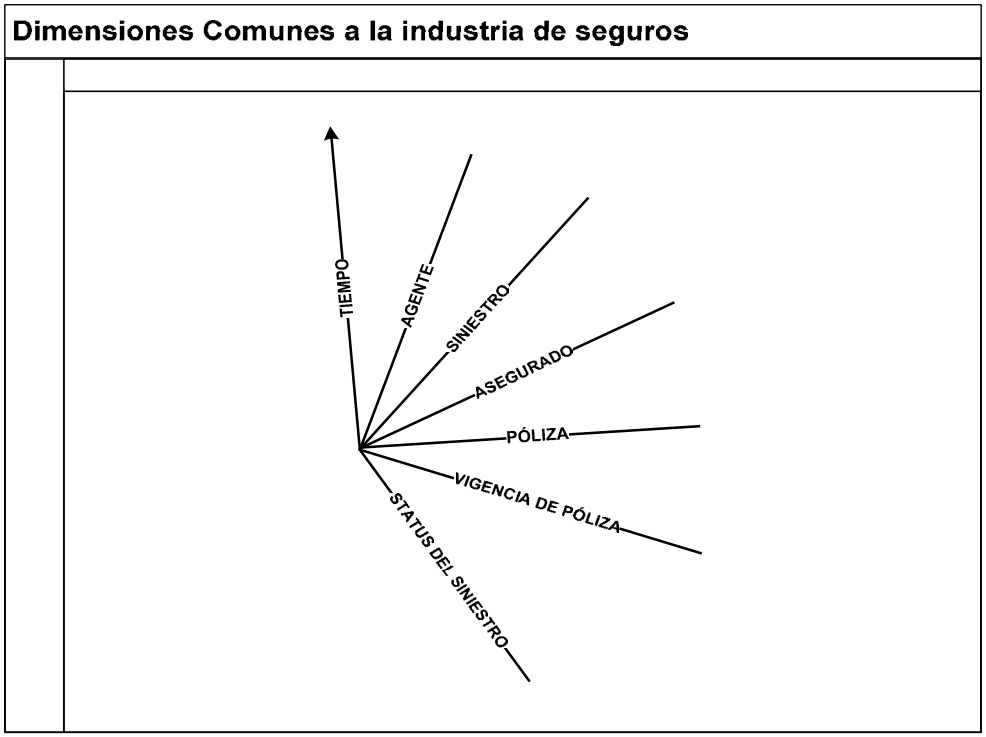
Este mismo concepto se puede extender a un análisis con múltiples dimensiones, en tal caso estaremos hablando de cubos de información multidimensionales, también conocidos como hipercubos.

Como hemos visto hasta ahora, las dimensiones de negocio constituyen la base, en la cuál descansa la metodología para la definición de requerimientos. Para el éxito del proyecto, deberemos asegurarnos que todos los datos necesarios quedarán almacenados, para así poder alimentar a las dimensiones.

Por esta razón, se debe poner especial atención a las dimensiones de negocio, así como a sus diversos niveles jerárquicos. Debemos ser capaces de seleccionar un conjunto óptimo de dimensiones y jerarquías para que se relacionen con las métricas que deseamos obtener.

Esta relación entre dimensiones y jerarquías, le darán mucho más sentido a la información. Es así como transformamos los datos, en información útil para el usuario.

Para profundizar en el caso de estudio de la industria del seguro; en el siguiente diagrama (Figura 2.8.) mostramos las dimensiones de negocio más relevantes:



2.4.1. Requerimientos de información

Como hemos comentado, al inicio de ésta sección; los usuarios no son capaces de describir por completo, qué desean obtener del Data Warehouse. Por esta razón nosotros, tenemos una mayor dificultad para determinar qué información mantendremos y cuál dejaremos fuera del alcance. A este punto no podemos determinar cómo, cada tipo de usuario estará utilizando el Data Warehouse.

Sabemos que el requerimiento, no puede definirse por completo usando las técnicas de levantamiento de información convencionales; por tanto deberemos aplicar un nuevo concepto para obtener información útil del usuario, que nos ayude a delimitar y definir adecuadamente los nuevos requerimientos.

Es así como la metodología se basa en obtener de los usuarios, las métricas básicas que utilizan, así como las dimensiones de negocio, a través de las cuales desean analizar la información. A este grupo de métricas y dimensiones, se le llama paquete de información; y está enfocado a responder las preguntas de negocio sobre un tema específico. Por ejemplo, un tema podría ser *ventas*.

En el diagrama que sigue (Figura 2.9.), podemos observar un ejemplo de paquete de información, para el tema de ventas.

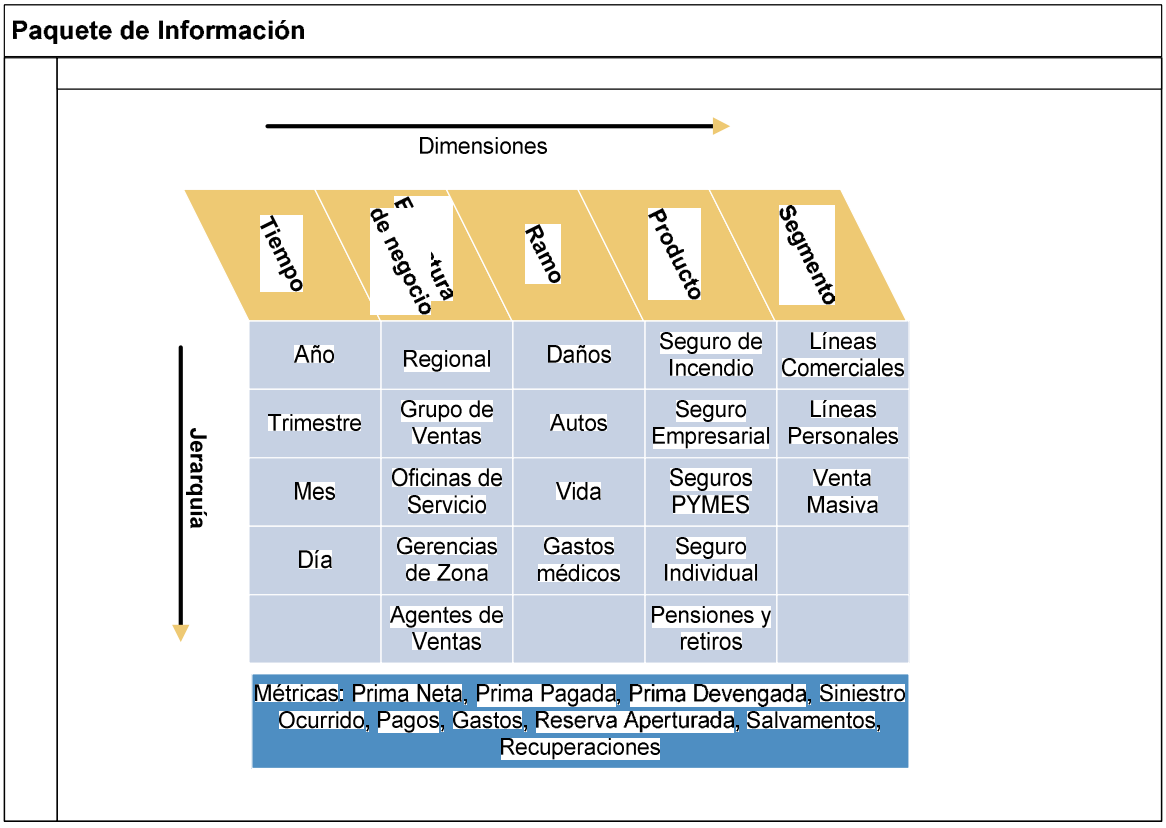


Figura 2.9. Paquete de Información

Un paquete de información consta de una matriz, donde en la parte inferior colocamos las métricas a ser utilizadas. Posteriormente, en forma de columnas, establecemos las dimensiones de negocio, que son relevantes para los usuarios. Finalmente cada una de estas dimensiones, tienen varios niveles o jerarquías. Por ejemplo la dimensión tiempo, tiene una jerarquía que va desde año hasta el día.

Podemos resumir, diciendo que el uso de paquetes de información nos permite:

- Definir temas comunes entre áreas de la organización
- Diseñar la métricas clave para la organización
- Decidir cuántos datos deberán ser presentados
- Determinar cómo los usuarios agregarán y desagregarán los datos
- Decidir el volumen de datos para analizar o consultar
- Decidir cómo serán accedidos los datos
- Establecer la granularidad de los datos
- Estimar el tamaño del Data Warehouse
- Determinar la frecuencia de refresco de datos
- Verificar cómo deberán ser empaquetados los datos

2.4.1.1. Métodos de recopilación de requerimientos

Debemos recordar que un Data Warehouse, es un sistema de entrega de información, que provee información para el soporte a la toma de decisiones. No es un sistema para administrar el día a día de un negocio. Así viene la pregunta obligada, ¿Quiénes hacen uso de la información contenida en el Data Warehouse?

De manera muy general podemos clasificar a los usuarios como sigue:

- Altos Ejecutivos (incluyendo a los patrocinadores del proyecto)
- Gerentes de área
- Analistas de negocio
- Administradores de Base de Datos para sistemas operacionales
- Alguno otro designado por cualquiera de los anteriores

Los altos ejecutivos nos darán la dirección y el alcance del proyecto, ya que tienen una visión más clara de los indicadores que desean obtener, así como el cambio de estrategia que está buscando la organización. Los más involucrados con el área o departamento objetivo son los gerentes de área, que a su vez reportan a los altos ejecutivos.

Los analistas de negocio generan reportes y análisis para los gerentes y altos ejecutivos; los administradores de base de datos, nos darán la información necesaria para identificar las fuentes de información.

Los elementos que debemos tomar en cuenta, a la hora de definir un requerimiento, se listan a continuación:

- Elementos de datos: hechos y dimensiones
- La forma como se graban los registros respecto al tiempo
- Extracciones de datos desde los sistemas fuente
- Reglas de negocio: atributos, rangos, dominios, registros operacionales.

Será necesario contactar a varias personas, a lo largo de varias áreas o departamentos, para lograr obtener la información necesaria, y definir el requerimiento en su totalidad. Para esto existen dos técnicas universalmente empleadas:

- Reuniones uno a uno, o en pequeños grupos
- Sesiones para desarrollo de aplicaciones

A continuación listamos las características de cada una de éstas técnicas:

Reuniones:

- Dos o tres personas simultáneamente
- Se agenda fácilmente
- Es una buena práctica, cuando los detalles son intrincados
- Algunos usuarios, sólo se sienten cómodos con reuniones uno a uno
- Requiere de una buena, preparación para ser efectiva
- Es bueno hacer una breve investigación, previa a la entrevista
- También es buena idea, hacer que los usuarios se preparen

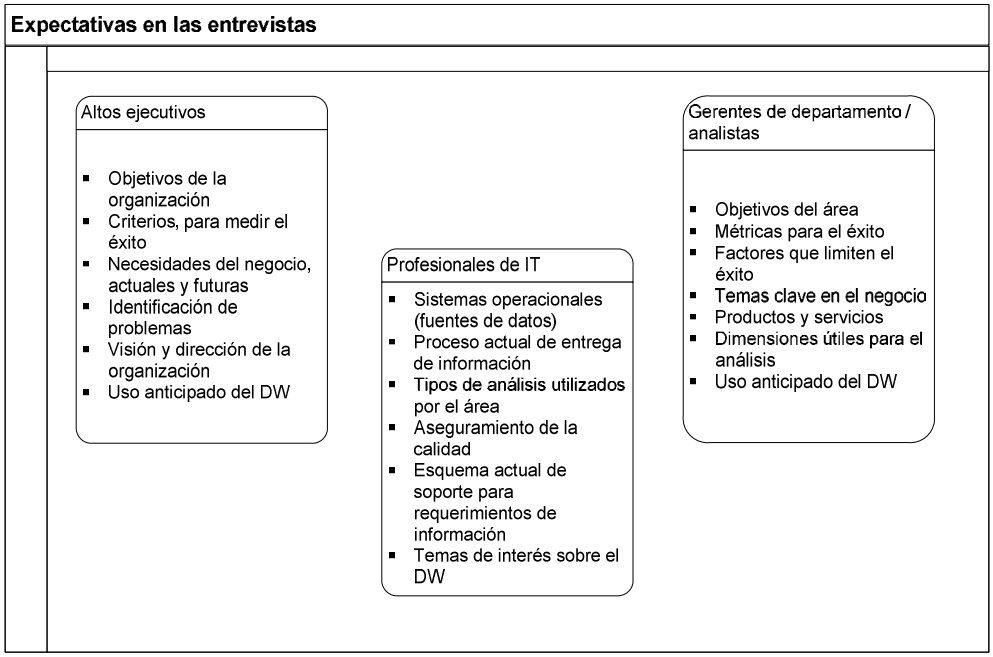
Sesiones:

- Grupos de veinte o menos personas, simultáneamente
- Emplear esta técnica, sólo después de tener una buena base de conocimiento sobre el tema
- No es adecuada para un levantamiento inicial de información
- Es muy útil para confirmar los requerimientos
- Es necesario contar con muy buena organización

Ya que las entrevistas, pueden requerir de un buen porcentaje del tiempo para un proyecto, es imprescindible hacer un buen manejo y organización de las mismas. Antes de enviar una convocatoria para alguna reunión, es bueno verificar que el equipo cumple con los siguientes puntos:

- Tener seleccionados y capacitados, a los miembros del equipo que estarán conduciendo las reuniones
- Asignar roles específicos para cada miembro del equipo (líder, entrevistador, escriba)
- Preparar una lista de los usuarios a ser entrevistados, un plan de las fecha y horarios
- Lista de las expectativas, para cada uno de los grupos de entrevistas
- Completar una breve investigación, previa a la entrevista
- Preparar un cuestionario para la entrevista
- Preparar a los usuarios para la entrevista
- Conducir una reunión de inducción para todos los usuarios a ser entrevistados

Finalmente, es necesario tener en mente una serie de expectativas a cubrir, para cada uno de los grupos de usuarios que estaremos entrevistando a lo largo de la vida del proyecto. En el diagrama siguiente (Figura 2.10.) se listan dichas expectativas, segmentadas por grupo de interés.


Expectativas en las entrevistas
Altos ejecutivos
Objetivos de la organización
Criterios, para medir el éxito
Necesidades del negocio, actuales y futuras
Identificación de problemas
Visión y dirección de la organización
Uso anticipado del DW
Profesionales de IT
Sistemas operacionales (fuentes de datos)
Proceso actual de entrega de información
Tipos de análisis utilizados por el área
Aseguramiento de la calidad
Esquema actual de soporte para requerimientos de información
Temas de interés sobre el DW
Gerentes de departamento / analistas
Objetivos del área
Métricas para el éxito
Factores que limiten el éxito
Temas clave en el negocio
Productos y servicios
Dimensiones útiles para el análisis
Uso anticipado del DW

Áreas o departamentos objetivo

¿Cuáles son las áreas más importantes para los análisis?
¿Cuáles son las dimensiones? ¿éstas tienen una jerarquía natural?
¿Qué divisiones son usadas en la toma de decisiones?
¿Las diferentes áreas, requieren información global o sólo información local, para la toma de decisiones? ¿Si requieren información mixta, cuál es?
¿Existen productos o servicios ofrecidos sólo en ciertas áreas?

Métricas

¿Cuáles son las unidades en las que se mide el desempeño?
¿Cuáles son los factores clave para el éxito, y cómo se monitorean?
¿Cómo acumulan estas métricas, a lo largo del tiempo?
¿En el mercado se hacen las mediciones de la misma forma?

Frecuencia de información

¿Con qué frecuencia deben actualizarse los datos, para la toma de decisiones?
¿Cuál es la ventana de tiempo para ésta actualización?
¿Cómo compara cada tipo de análisis, las métricas respecto al tiempo?
¿Cuál es el plazo de entrega de información?
Adicional a esto, es necesario ir generando cierta documentación. Entre los documentos básicos se encuentran:

- Perfil del usuario
- Antecedentes y objetivos
- Requerimientos de información
- Requerimientos de análisis
- Inventario de herramientas, utilizadas actualmente
- Criterios para medir el éxito
- Métricas útiles para el negocio
- Dimensiones relevantes para el negocio

2.4.2. Jerarquías en los negocios

Las jerarquías constituyen una parte integral para todo negocio, además de ser un concepto fundamental en el Data Warehouse. Una colección y publicación adecuada de información, organizada de forma jerárquica, así como su integración con datos transaccionales; dan las pautas para una implantación exitosa.

Sin las jerarquías, no sería posible definir una línea de acción, tampoco podríamos organizar los materiales, productos o clientes. No sería posible llevar a cabo planeaciones, presupuestos, proyecciones ni análisis. Los reportes gerenciales, estarían sobrecargados de información a nivel detalle, sin la posibilidad de identificar dónde están los posibles problemas, ni dónde están las áreas de oportunidad.

Las jerarquías forman parte del proceso natural de evolución de todo negocio; ya que a medida que un negocio se expande, es natural subdividirlo y extender también el control, para permitir que los recursos sean administrados de la mejor manera posible.

Adicionalmente, cada elemento de información puede tener más de una jerarquía. Por ejemplo, un producto, puede tener una jerarquía basada en sus características físicas, y otra muy diferente, si se basa en sus características de uso.

También es muy común, encontrar jerarquías combinadas, para llevar a cabo análisis y reportes. Por ejemplo la fuerza de ventas de una aseguradora, típicamente genera información relativa a sus ventas, basándose en una estructura o jerarquía local; sin embargo, cuando se reportan resultados a los niveles superiores, es necesario alinear éstos con la jerarquía o estructura competencial de la organización.

Es por ello, que las entidades dentro de un Data Warehouse, frecuentemente muestran una relación "padre-hijo", en este tipo de relaciones un padre puede tener varios hijos, pero un hijo sólo puede pertenecer a un padre. Por ejemplo, un departamento dentro de la empresa, puede tener varias personas asignadas, pero una persona sólo puede pertenecer a un departamento. Por esta razón las jerarquías de este tipo, son frecuentemente llamadas árboles.

Dentro de una jerarquía, un hijo representa el nivel más bajo de detalle o granularidad de un padre; esto desde luego, genera un vínculo de pertenencia o control, de un nivel superior (padre) hacia los niveles inferiores (hijos).

Vale la pena mencionar que la terminología de las jerarquías, puede variar; ya que algunos autores prefieren dar el nombre de "nodo" a cada uno de los miembros de la jerarquía. De tal forma que el nivel superior a todos, será llamado "nodo raíz", y el nivel más bajo de todos será conocido como "nodo hoja". Un nodo padre, es aquel que tiene hijos, y un nodo hijo es aquel que desciende de un padre. Un nodo padre (excepto el nodo raíz) puede a su vez ser un nodo hijo, y un nodo hijo (excepto el nodo hoja) puede ser padre.

A medida que una jerarquía se extiende, debemos incluir el concepto de profundidad. La profundidad es el número de niveles o generaciones que tiene una jerarquía; esto es el conjunto de padres, y sucesivamente de los hijos de los hijos.

En el siguiente diagrama (Figura 2.11.) podemos observar un ejemplo de los niveles jerárquicos, o estructura competencial dentro de una organización:

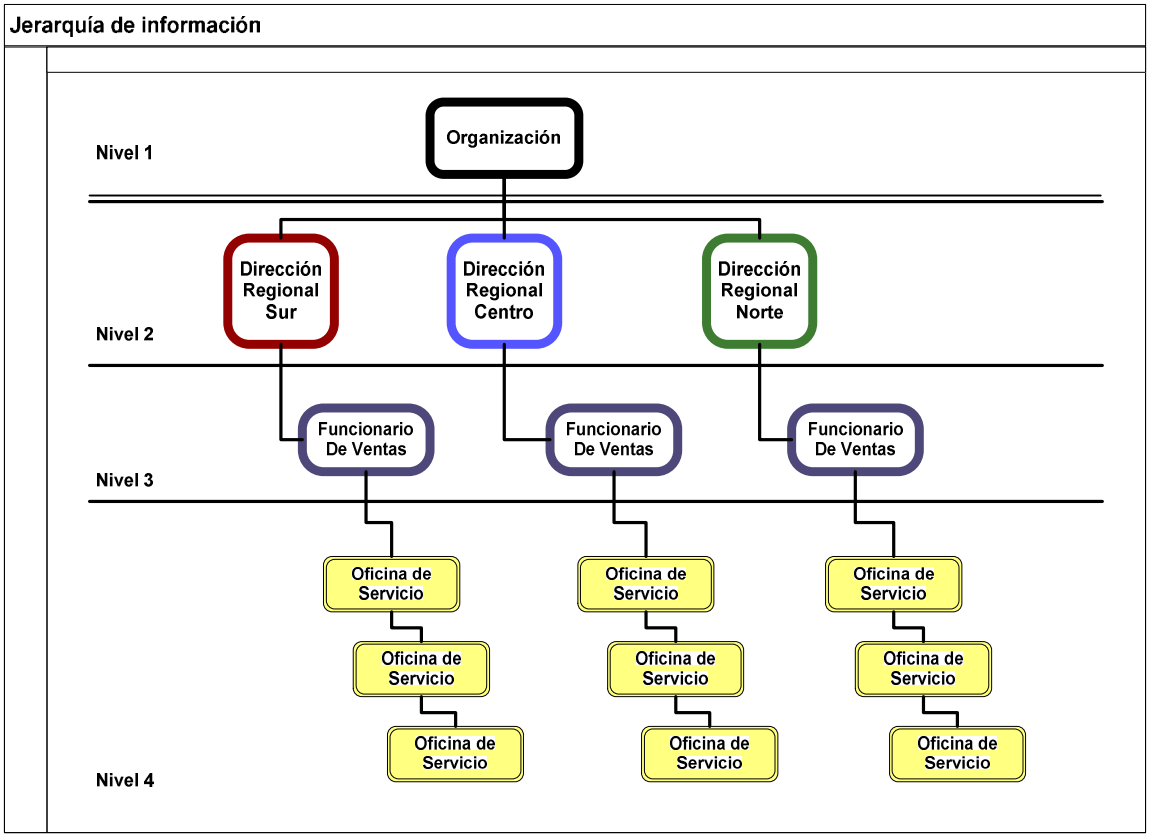


Figura 2.11. Jerarquía de Información

En la figura, podemos ver un ejemplo de estructura para la fuerza de ventas; que está dividido en oficinas de servicio, donde cada oficina de servicio le reportará al nivel superior, funcionario de ventas. El funcionario de ventas, a su vez le reportará director regional. Finalmente los directores regionales, darán su reporte al consejo de administración de la organización.

2.4.3. Modelando las dimensiones

El modelado dimensional toma su nombre de las dimensiones de negocio, que necesitamos incorporar al modelo lógico de datos. Es una técnica que se utiliza para estructurar las dimensiones y las métricas que serán analizadas a través de dichas dimensiones.

El paquete de información mencionado en la sección *2.4.1. Requerimientos de información*, es la base para el modelado multidimensional; que se compone de las estructuras de datos específicas que son requeridas por los analistas del negocio.

En el diagrama siguiente (Figura 2.12.) se muestra el flujo que sigue el proceso para la definición de requerimientos:

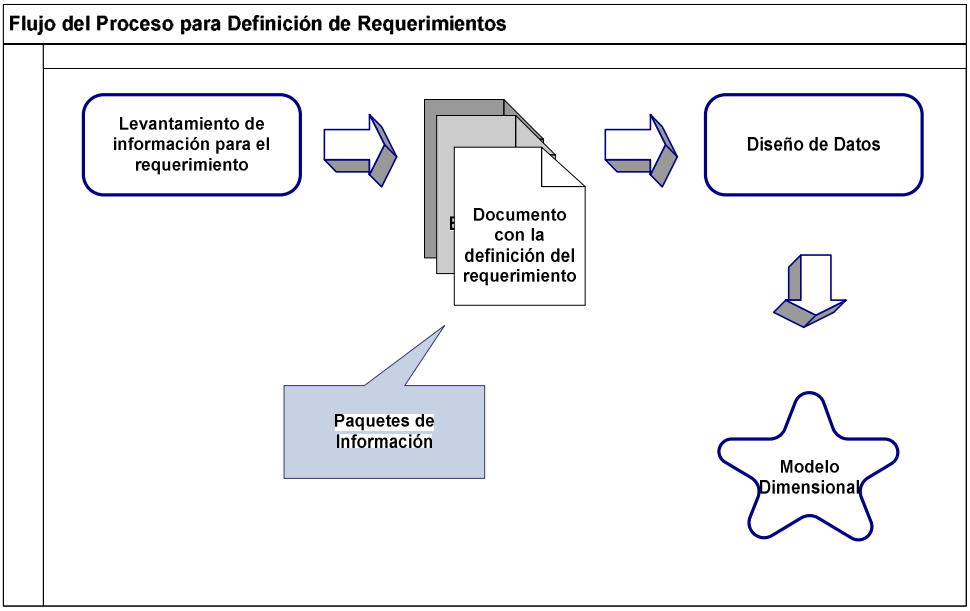


Figura 2.12. Proceso de Definición de Requerimientos

Como se muestra, iniciamos con el levantamiento de información, que es la etapa previa; posteriormente generamos la documentación con la definición de requerimiento, donde incluimos los paquetes de información. Posteriormente procedemos a diseñar el modelo de datos y finalmente generamos el modelo dimensional.

Ahora, antes de continuar, es necesario listar algunas de las decisiones de diseño que deben ser tomadas en cuenta:

- **Elección del proceso**: Consiste en seleccionar los temas, para el primer conjunto de estructuras lógicas a ser diseñadas, basándonos en los paquetes de información.
- **Elegir la granuralidad**: Determinar el nivel de detalle de los datos en las estructuras.
- **Determinar y conformar las dimensiones**: Elegir las dimensiones (producto, tiempo, geografía, etc.), y asegurarnos de que cada elemento dentro de las dimensiones está alineado.
- **Elegir los Hechos**: Los hechos o métricas a ser incluidas en el primer conjunto de estructuras (unidades vendidas, ventas en dólares, ventas en pesos, etc.).
- **Determinar la duración:** Definir qué tanto iremos hacia atrás, cuando se trate de datos históricos.

Volviendo al tema del paquete de información, y analizando la información contenida en él. Podemos distinguir tres tipos de entidades de datos: (1) métricas, (2) dimensiones, (3) atributos para cada una de las dimensiones. En este punto necesitamos contar con un modelo de datos que represente estos tres tipos de entidades. ¿Cómo hacemos esto?

En primer lugar trabajaremos con las métricas, recordemos que están listadas en la parte inferior, del paquete de información. Retomando el ejemplo de la sección *2.4.1. Requerimientos de información:*

- Prima Neta
- Prima Pagada

- Prima Devengada
- Siniestro Ocurrido
- Pagos
- Gastos
- Reserva Apertura
- Salvamentos
- Recuperaciones

Cada uno de estos elementos es un hecho o métrica. La prima neta, es el importe que un asegurado deberá pagar por la póliza de seguro. La prima pagada, es el importe que el asegurado ha pagado a la compañía de seguros, en contraprestación del servicio. La prima devengada, es el monto que la compañía de seguros ha devengado en virtud del tiempo de cobertura del riesgo asegurado. El siniestro ocurrido es el importe total que la compañía de seguros ha tenido que pagar al asegurado y/o beneficiarios, en virtud de un siniestro. Pagos y gastos son los importes pagados a terceros, derivados de la atención de un siniestro. La reserva de apertura es el importe planeado de los gastos derivados de un siniestro. Salvamentos y recuperaciones, son ganancias que la compañía aseguradora puede obtener a partir de un siniestro atendido. Todos estos elementos pueden estar agrupados en una sola estructura de datos; una tabla relacional. De este modo, las métricas forman lo que llamaremos *tabla de hechos*.

Esta es una de las estructuras de datos que estarán incluidas, en el modelo dimensional; que a su vez fue derivada del paquete de información.

Ahora continuemos con las dimensiones, que como hemos visto, son usadas cuando deseamos analizar las métricas. Para nuestro modelo de seguros, retomamos el paquete de información, donde encontramos las dimensiones de tiempo, estructura del negocio, ramo, producto y segmento.

La dimensión de producto puede ser usada, cuando deseamos analizar las métricas divididas por productos. Sin embargo, en algunas ocasiones podremos hacer algunos análisis por productos individuales. La lista de elementos relacionados con la dimensión son los siguientes:

- Seguro empresarial
- Seguro pymes
- Seguro individual
- Pensiones y retiros

Al igual que hicimos con las métricas, todos estos elementos pueden ser agrupados dentro de una entidad de datos, que llamaremos tabla de dimensión. Para nuestro ejemplo, ésta será la dimensión de producto; donde cada uno de los elementos listados, será un atributo de dicha tabla.
Repetiremos este mismo procedimiento, para formar las dimensiones que están presentes en el paquete de información. Dichas dimensiones son: tiempo, estructura del negocio, ramo y segmento.

A este punto contamos con una tabla de hechos, y seis tablas de dimensión; ahora surgen varias preguntas: ¿cómo deberán distribuirse estas tablas en el modelo dimensional? ¿Cuáles son las relaciones entre estas tablas?

Básicamente el modelo dimensional debe facilitar las consultas. Ahora ¿cuáles serán los tipos de análisis y consultas? Estas consultas y análisis integran la información de las métricas contenidas en la tabla de hechos, distribuidas por uno o más de los atributos contenidos en las dimensiones.

Antes de poder decidir, cómo vamos a distribuir las dimensiones en nuestro nuevo modelo dimensional, y marcar las relaciones; conviene repasar algunos criterios para asegurar que el modelo cumpla su propósito:

- El modelo debe proveer el mejor acceso a los datos.
- El modelo debe estar enfocado a las consultas.
- Debe estar optimizado para consultas y análisis.
- El modelo debe mostrar la interacción de la tabla de hechos y las tablas de dimensión.
- Debe estar estructurado de tal modo, que cada dimensión interactúe de forma equitativa con la tabla de hechos.
- El modelo debe permitir visualizar el detalle hacia arriba y hacia abajo a lo largo de las jerarquías, para cada tabla de dimensión.

Con estos criterios en mente, encontramos un arreglo que satisface éstas condiciones, se trata de un esquema donde la tabla de hechos se encuentra en el centro, y las tablas de dimensiones se encuentran distribuidas alrededor de ésta última. Esto es porque cada tabla de dimensión con sus respectivos atributos, debe tener la misma oportunidad de interactuar en consultas y análisis con las métricas contenidas en la tabla de hechos. Tal arreglo es parecido a una estrella; por tal motivo se le llama *modelo estrella*.

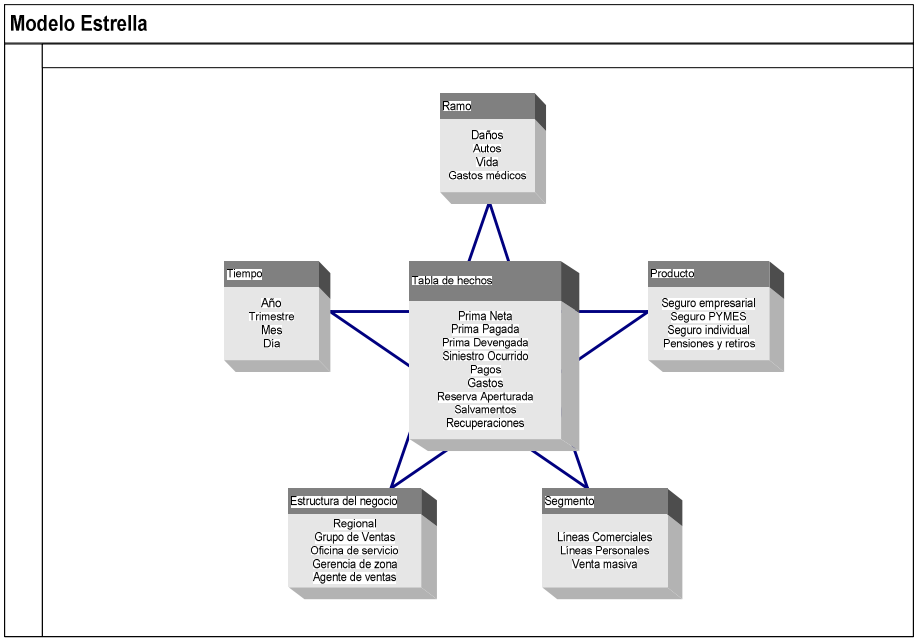


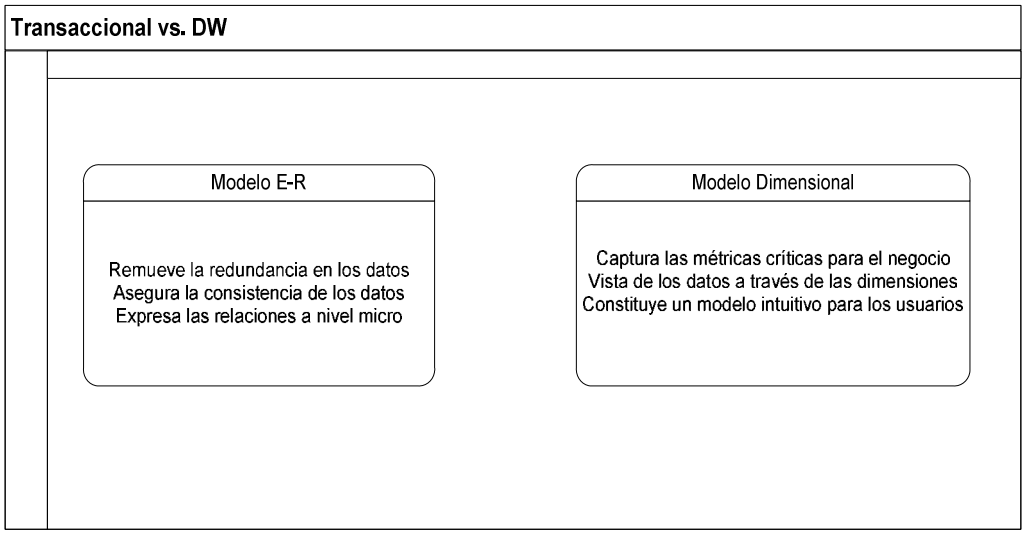
Figura 2.13. Modelo Estrella

En la figura anterior (Figura 2.13.) se muestra el modelo estrella, con los atributos que comúnmente son utilizados en una compañía de seguros. En el centro del modelo encontramos la tabla de hechos que contiene las métricas, alrededor de ésta las tablas de dimensión: ramo, producto, segmento, estructura y tiempo.

Cada tabla de dimensión tiene una relación “uno a muchos”, con la tabla de hechos; es decir, que cada renglón de las tablas de dimensión, tendrá una correspondencia con uno o más renglones en la tabla de hechos.

2.4.4. Diferencias de Modelado

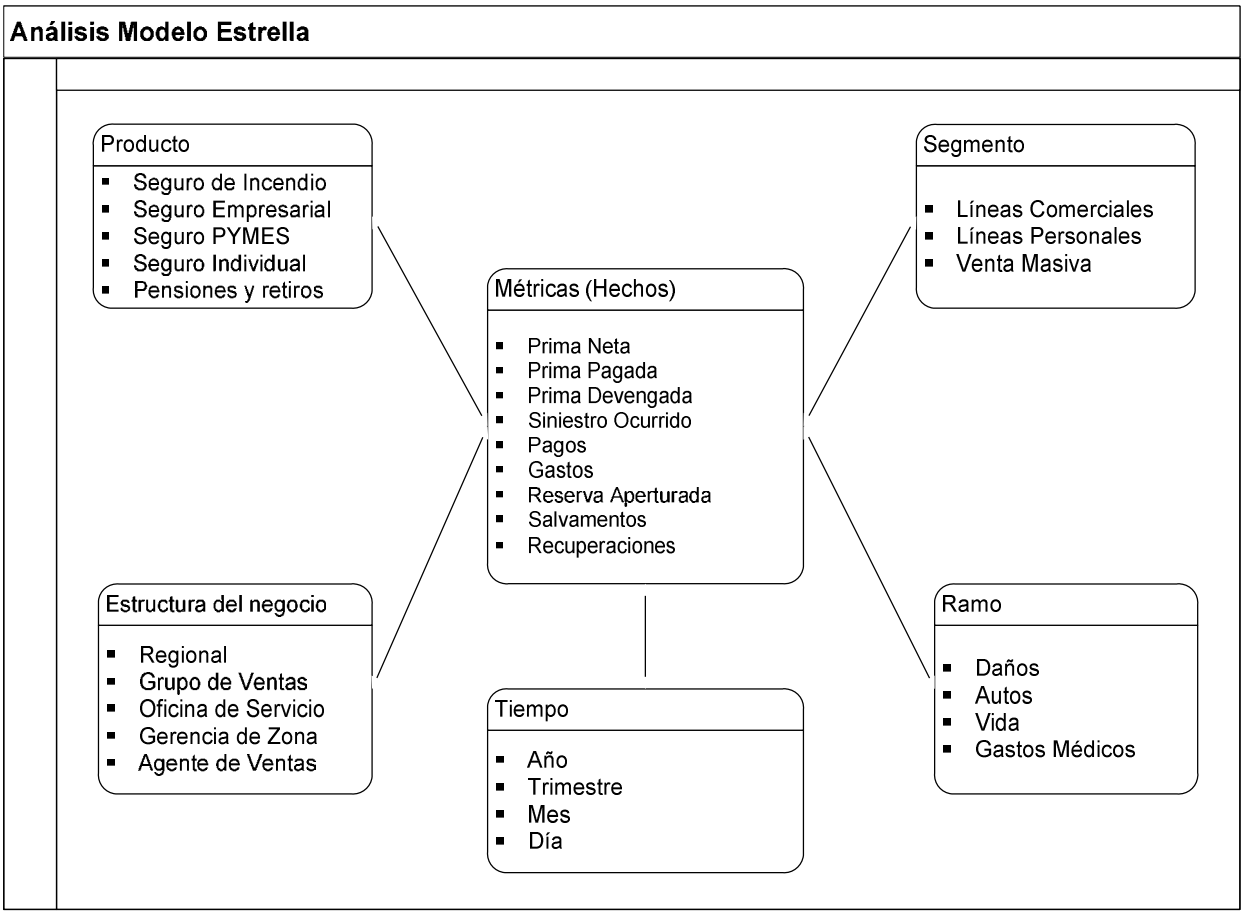
La técnica de modelado de Entidad-Relación, es la más adecuada para los sistemas transaccionales; por esta razón es algo con lo que la mayoría está más familiarizada. En cambio el modelo dimensional está pensado para responder preguntas de alto nivel, es decir preguntas sobre un proceso global dentro de la organización. En el siguiente diagrama (Figura 2.14.) podemos observar un comparativo entre el modelo E-R y el modelo Dimensional.



En modelo propuesto, será posible analizar los importes de Prima Neta, Prima Pagada, Prima Devengada, Siniestro Ocurrido, Pagos, Gastos, Reserva Aperturada, Salvamentos, Recuperaciones.

Adicionalmente la información puede tener cortes basados en tiempo, estructura, ramo, producto y segmento. En general, los cortes posibles, son las distintas combinaciones de las métricas, con los datos contenidos en cada una de las tablas de dimensión.

En el siguiente diagrama (Figura 2.15.) se muestra el modelo estrella propuesto en el ejemplo:



Hagamos una consulta simple, supongamos que el área de marketing, desea saber el importe total de prima emitida para el producto de seguro de incendio, contabilizado en el mes de enero, cuyo agente de ventas sea Paola Suzuki. En el siguiente diagrama (Figura 2.16.) se muestra el flujo de dicha consulta, en los globos puede verse los filtros aplicados.

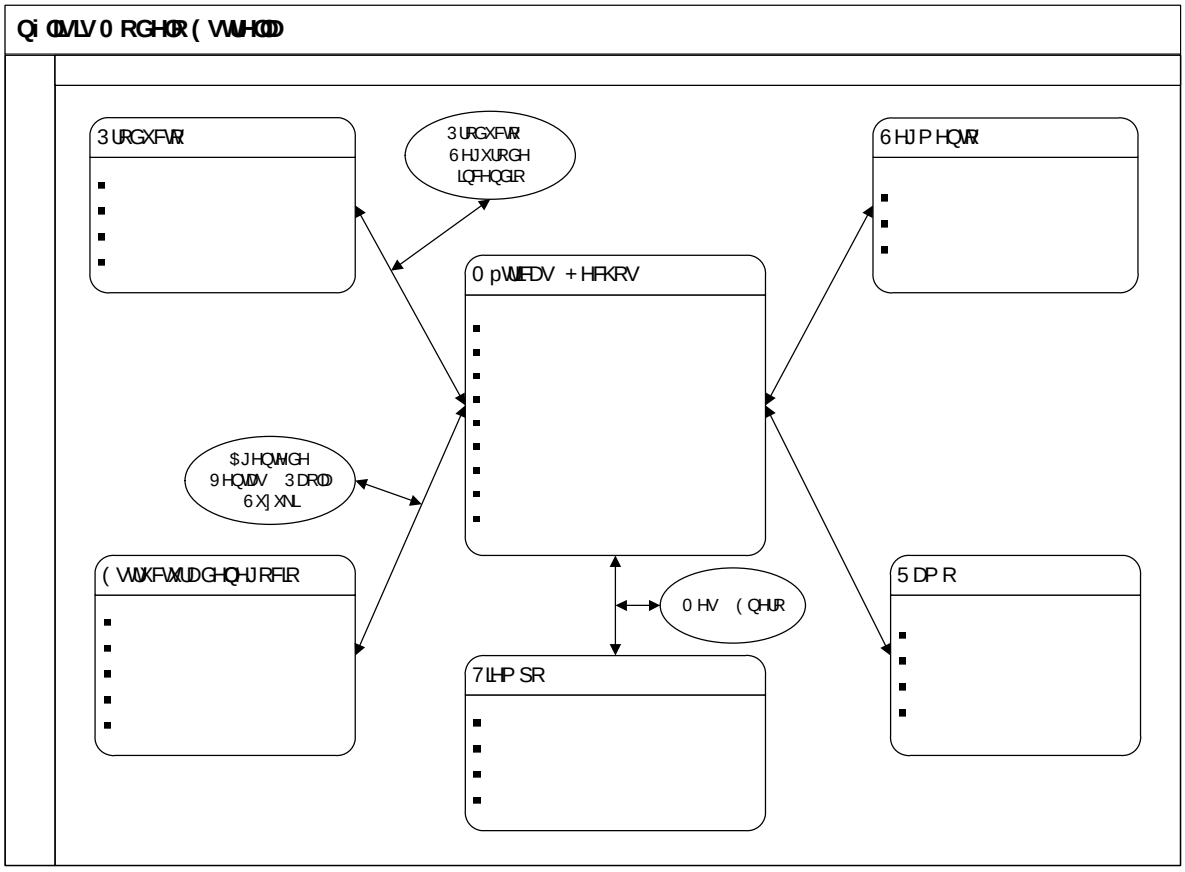


Figura 2.16. Análisis Modelo Estrella

Durante la consulta, se aplican los filtros que ya comentamos, obteniendo así la respuesta a la pregunta de negocio que fue formulada por los usuarios.

A continuación listamos las características más importantes, de las tablas de dimensión:

- **Cuenta con una llave:** Ésta llave identifica de manera única cada uno de los registros contenidos en la tabla de dimensión.
- **La tabla es grande:** De forma típica, una tabla de dimensiones cuenta con muchos atributos o columnas. No es raro encontrar tabla de dimensiones hasta con 50 atributos.
- **Atributos textuales:** En una tabla de dimensión, es poco frecuente encontrar valores numéricos para cálculos, la gran mayoría tiene atributos con formato texto. Ya que éstos representan las descripciones de los componentes de las dimensiones de negocio.
- **Atributos no relacionados directamente:** Es frecuente encontrarse con algunos atributos, dentro de la tabla de dimensiones, que no están directamente relacionados; por ejemplo, si

miramos a la dimensión producto, el seguro de incendio no está relacionado con las pensiones y retiros, sin embargo forman parte de la misma tabla.

- **No normalizado:** Los atributos en una tabla de dimensión, son utilizados una y otra vez en consultas. Un atributo se toma como filtro dentro de la consulta, y aplicado directamente a las métricas contenidas en la tabla de hechos. Por eficiencia en la consulta, es mejor utilizar este método, ya que si el modelo estuviera normalizado, estaríamos utilizando tablas intermediarias, que sólo aumentan el tiempo de respuesta y complican la consulta.
- **Navegación hacia arriba y hacia abajo:** Los atributos dentro de una tabla de dimensión, proveen la posibilidad de obtener los datos de detalle, viajando de lo general a lo particular; es decir podemos ir de los niveles más altos de granularidad hasta los niveles más bajos.
- **Jerarquías múltiples:** Las tablas de dimensiones, a menudo proveen jerarquías múltiples. Tomemos como ejemplo, el caso de la tabla de productos; el departamento de marketing, puede tener su propia clasificación jerárquica de productos. El departamento de contabilidad, puede agrupar los productos de una forma distinta. En este caso, la dimensión de producto, podrá contener atributos para la agrupación de contabilidad y la agrupación de marketing.
- **Número de registros reducido:** Una tabla de dimensión, típicamente contiene menos registros o renglones, que una tabla de hechos. Es así como la tabla de dimensión de productos puede tener sólo 500 registros, en contraste con la tabla de hechos, que con facilidad, puede contener millones de registros.

Ahora continuemos con las características de las tablas de hechos, recordemos que es aquí donde almacenamos las métricas:

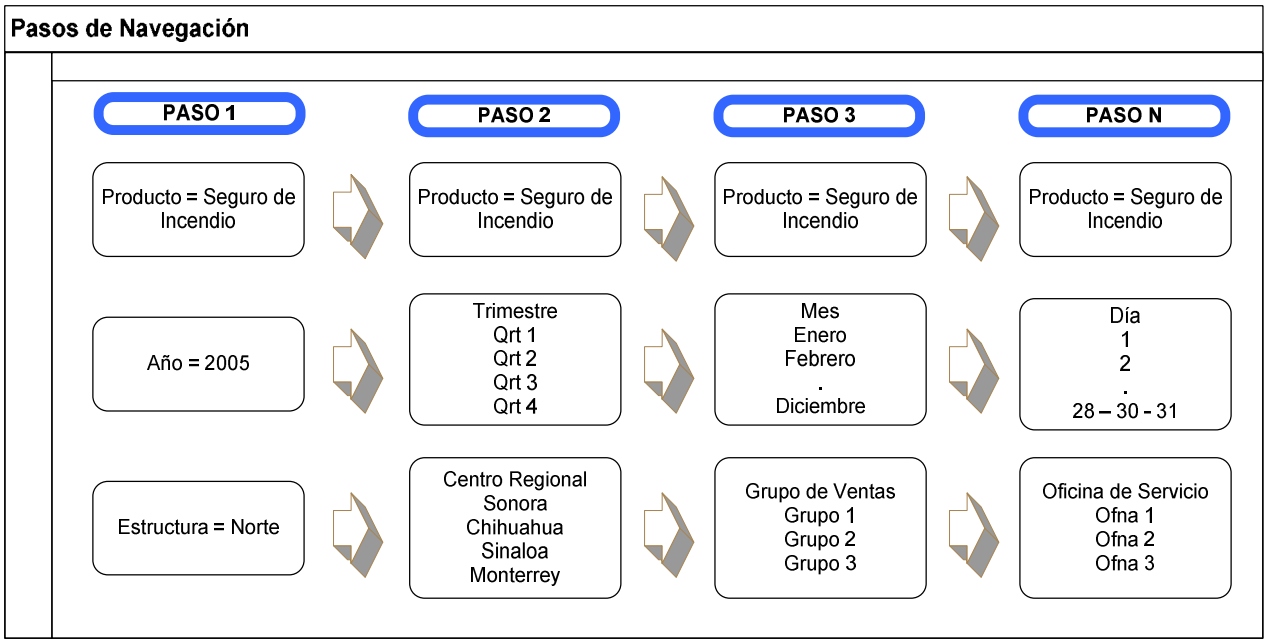
- **Llave concatenada:** Un reglón dentro de la tabla de hechos, está relacionado con una combinación de renglones, de todas las tablas de dimensión. Esto significa que un renglón en la tabla de hechos, debe estar identificado por la concatenación de las llaves primarias de todas las tablas de dimensión.
- **Granularidad:** Esta es una importante característica de la tabla de hechos. Como ya sabemos, la granularidad es el nivel de detalle que tendrán las métricas o hechos.
- **Métricas completamente aditivas:** Son aquellas que pueden calcularse, por suma directa. Debemos asegurarnos que las métricas almacenadas en la tabla de hechos, son aditivas entre si; de lo contrario no representaremos los totales correctos. Por ejemplo, pensemos en que la prima neta está almacenada en pesos mexicanos, y en algunos casos está almacenada en dólares americanos; para poder almacenar ésta métrica, dentro de la misma tabla de hechos; debemos dejar el importe de prima representado en una sola moneda (pesos mexicanos) aplicando algún tipo cambio definido por el usuario.
- **Métricas semiaditivas:** Son métricas que dependen, de otras para poder ser calculadas. Tomemos como ejemplo, el calculo del porcentaje de ventas dividido por producto; pero éste cálculo no puede completarse, sino hasta saber el total de productos contenidos en la tabla de dimensión de productos.
- **La tabla es profunda:** Generalmente, una tabla de hechos contiene menos atributos o columnas, que una tabla de dimensión. Pero en cambio, en número de registros es mucho mayor. Tomemos el caso simple, de tres productos, treinta días y diez agentes de ventas. El número de registros en la tabla de hechos será de 900. Si esto lo llevamos a 365 días, la tabla crecerá 10, 950 registros en un año.
- **Datos dispersos:** Hemos dicho que cada renglón en la tabla de hechos, está relacionado a un producto particular, fecha, estructura, agente de ventas, ramo y segmento. Pero qué sucede, cuando un agente de ventas está de vacaciones, en tal caso no tendrá producción; y entonces habrá datos nulos o inexistentes dentro de la tabla de hechos. Así podrán existir muchas combinaciones las cuales darán como resultado datos nulos.

- **Dimensiones degeneradas:** Al seleccionar atributos desde los sistemas operacionales, tanto para las tablas de dimensiones como para las tablas de hechos, en ocasiones terminamos con elementos que no son métricas, ni tampoco estrictamente elementos de tablas de dimensiones. Ejemplos de éste tipo de elementos son, los números de recibo, número de factura, número de siniestro, etc.

Para concluir esta sección, entraremos un poco más a detalle, con el asunto de la navegación hacia arriba y hacia abajo, también conocida como “Drill-Down”. Es muy importante dejar claro, que el dejar la tabla de hechos con la granularidad más baja, permite realizar la navegación hasta el mínimo nivel de información, usando el data warehouse, en lugar de tratar de obtener la información al mismo nivel, desde los sistemas operacionales.

En el diagrama (Figura 2.17) mostramos la navegación, con tres de las dimensiones utilizadas en el ejemplo:

En el paso uno tendremos la posibilidad de elegir un producto, un año y una estructura de negocio determinada. Posteriormente, en el paso dos, tendremos la posibilidad de ver un nivel más abajo del detalle establecido en la jerarquía de la tabla de dimensión; como lo son el trimestre y los centros regionales, y así sucesivamente. Si continuamos a los pasos tres, cuatro, n-ésimo, notamos que en realidad navegamos en las jerarquías de cada una de las dimensiones definidas:



2.4.6.1. Llave primaria

Cada renglón almacenado en una tabla de dimensión, está identificado por un valor único que llamamos llave primaria de la tabla de dimensión. En la dimensión de producto, la llave primaria identifica de forma única a cada uno de los productos, y así de manera similar ocurre con todas las demás tablas de dimensión.

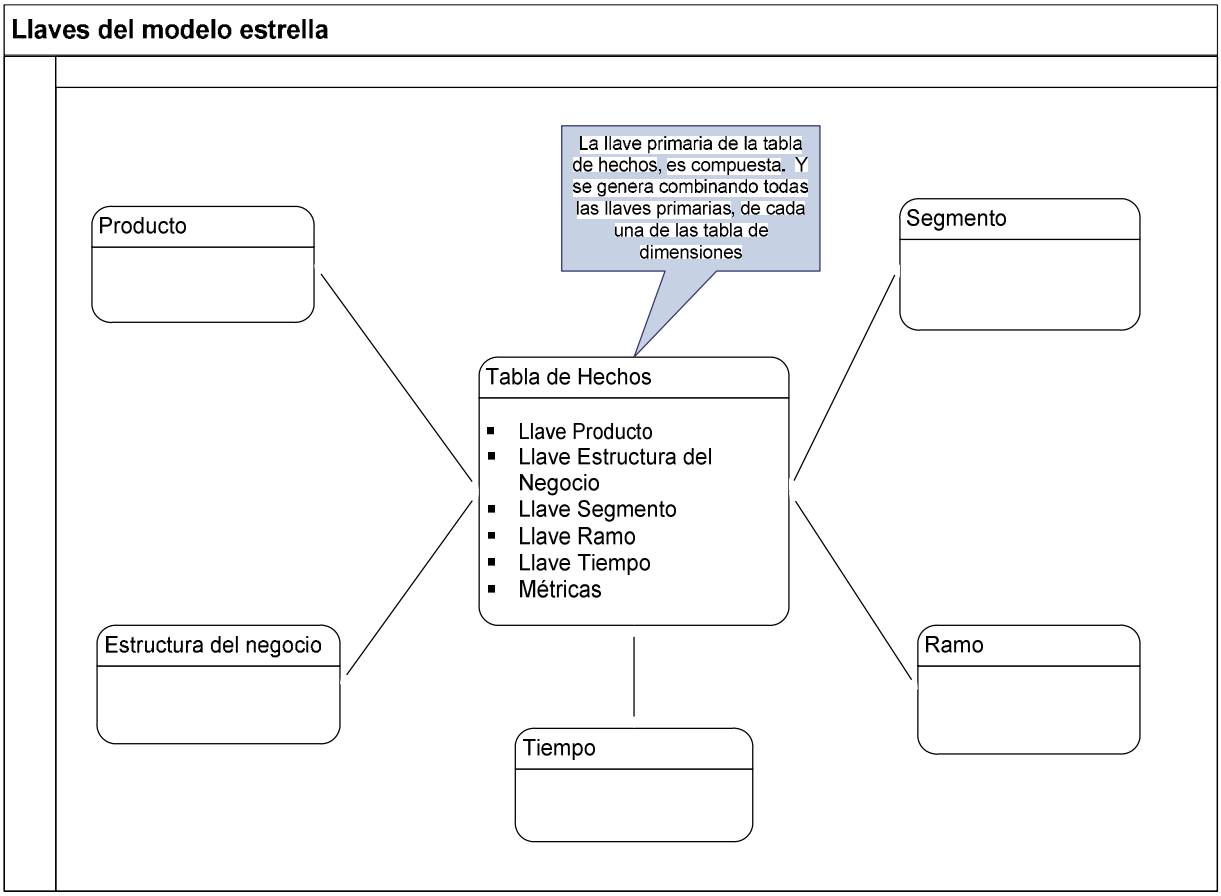


Figura 2.18. Llaves del modelo estrella

Debido a que la tabla de hechos, contiene la información al mínimo nivel de granularidad, y además almacena información histórica; la llave utilizada, es una combinación de las llaves de cada una de las dimensiones que componen al modelo estrella. En el diagrama anterior (Figura 2.18.) se muestra la forma como se relacionan las tablas de dimensión y la tabla de hechos, para generar la llave primaria de una tabla de hechos.

2.4.6.2. Llave sustituta

Supongamos que estamos evaluando diversas llaves candidatas, para finalmente seleccionar una para las tablas de dimensión, en el caso particular de la dimensión de producto, decidimos usar como llave el código de producto, utilizado en el sistema operacional. Un código de seis posiciones, donde los primeros dos caracteres representan el código del sistema origen de la información, las siguientes dos

posiciones indican una categoría de producto, y las posiciones restantes se refieren al código del producto en cuestión. Imaginemos que la organización decide modificar la nomenclatura del código de producto, debido a que el área de Marketing decide lanzar una nueva campaña de ventas. Esto obviamente generará problemas, a la hora de intentar agregar datos que fueron generados antes del cambio, junto con datos que fueron generados después del cambio.

Para evitar que algo similar nos suceda, contamos con dos principios generales, que deben aplicarse a la hora de asignar llaves a las tablas de dimensión. En el ejemplo utilizamos una llave, que en el sistema operacional tiene un significado intrínseco a los datos, porque diversos segmentos o posiciones de la llave, significan algo dentro del sistema. De tal forma que el principio número uno, es evitar lo más posible, el uso de llaves con significado intrínseco, también conocida como llave built-in.

El segundo principio es: No usar llaves productivas del sistema, como llave primaria para tablas de dimensión. Es decir, debemos evitar, el heredar llaves desde los sistemas operacionales. Para aclarar un poco más el concepto, propongamos el siguiente escenario:

Dentro de toda compañía, se tienen registro de los clientes en alguna base de datos. Sin embargo, algunos clientes seguramente han terminado su relación comercial con la organización, y existe la posibilidad de que los códigos de cliente pertenecientes a clientes dados de baja, se estén reasignando a clientes nuevos y activos. Ahora supongamos que decidimos usar el código de cliente, como llave para la dimensión de clientes; aquí tenemos un problema, ya que tendremos el mismo código de cliente, relacionado con datos que corresponden, tanto a nuevos clientes, como a clientes ya cesados.

Es así como, finalmente deberemos utilizar algún otro método, para asignar una llave primaria, a las tablas de dimensiones, deben utilizarse llaves sustitutas. Las llaves sustitutas son números secuenciales generados por el sistema, que no tienen ningún significado intrínseco. Por supuesto, las llaves primarias estarán mapeadas con las llaves productivas del sistema, a pesar de esto son diferentes.

La práctica general consiste en almacenar dentro de la tabla de dimensión, como atributos adicionales, las llaves productivas del sistema operacional. Así tendremos una tabla de dimensión, con una llave sustituta como llave primaria, y las llaves usadas en el sistema operacional almacenadas como referencia.

2.4.6.3. Llave foránea

Cada tabla de dimensión tiene una relación uno a muchos, con la tabla de hechos; lo que obliga a que cada una de las llaves primarias de las tablas de dimensiones, sea una llave foránea dentro de la tabla de hechos.

A la hora de asignar una llave primaria, para la tabla de hechos; la mejor opción resulta de formar una llave primaria concatenada, a partir de todas las llaves primarias de las tablas de dimensión. Esto nos permite relacionar de manera sencilla, cada uno de los renglones de la tabla de hechos, con los renglones de cada tabla de dimensión.

2.4.7. Ventajas del modelo estrella

A pesar que el modelo estrella es un modelo relacional, no está normalizado. Y cada una de las tablas de dimensión que lo componen, son desnormalizadas intencionalmente. Esta es la principal diferencia, entre un modelo dimensional y los modelos relacionales de los sistemas OLPT.

Sin embargo, antes de discutir las ventajas del modelo; debemos mencionar que la completa adherencia al modelo, no siempre es la mejor opción. Por ejemplo, si contamos con una dimensión de clientes, y la organización tiene demasiados; una tabla de dimensión desnormalizada no es lo más deseable. Ya que una tabla de dimensión grande, consecuentemente nos llevará a tener tablas de hechos muy grandes también. Al final, las ventajas del modelo estrella superan a las desventajas, y eso es lo que lo sitúa en un lugar predominante dentro del Data Warehouse.

2.4.7.1. Fácil comprensión para los usuarios

Los usuarios de sistemas OLPT, en general están habituados a interactuar con las aplicaciones, a través de una interfase gráfica (GUI), o bien con el uso de consultas predefinidas. Es así cómo los usuarios, prácticamente no tienen necesidad de conocer las estructuras de datos, que están detrás, para asegurar que todo funcione adecuadamente; este tipo de detalle, prefieren dejarlo a los profesionales de las tecnologías de información (IT).

Los usuarios de sistemas de soporte a las decisiones son diferentes, ellos formularán consultas por sí mismos, interactúan con el sistema usando herramientas de terceros, los usuarios deberán saber qué información pedir, y dónde obtenerlo.

El modelo estrella refleja de forma fiel, la manera como los usuarios ven la información, de cara a obtener datos relevantes a través de consultas. Debido a que ellos piensan en términos de métricas, pues se familiarizan rápidamente con la tabla de hechos, que contiene las métricas. Adicionalmente también piensan en términos de dimensiones de negocio, ya que es lo que le da sentido a las métricas con las que habitualmente trabajan. Además si añadimos que el modelo estrella, define las trayectorias de las uniones entre las tablas, de la misma en como los usuarios visualizan las relaciones entre éstas entidades de datos; el modelo se convierte en simple e intuitivo.

2.4.7.2. Optimiza la navegación

En una base de datos, el propósito de las relaciones o conexiones entre las entidades de datos, es usarlas para viajar de tabla a tabla, e ir obteniendo la información que se requiere. Si las uniones son numerosas e intrincadas, la navegación a través de la base de datos, se vuelve complicada y lenta. Por el contrario, si las relaciones son simples y directas, la navegación es optimizada y se vuelve más rápida.

Esta es una de las mayores ventajas al utilizar el modelo estrella, a pesar de obtener resultados de una consulta compleja, la navegación se mantiene simple y directa.

2.4.7.3. Ideal para el procesamiento de consultas

Ya hemos mencionado que el modelo estrella, es un esquema con una estructura centrada en las consultas. Veamos un ejemplo: Tratamos de obtener las pólizas vendidas para el producto de incendio, durante diciembre de 2006, ligadas al segmento de líneas personales; cuya prima neta sea igual o superior a $ 8, 500.

Esta es una consulta que involucra tres tablas de dimensiones y la tabla de hechos. La forma como se resuelve el ejercicio será: primero se obtienen sólo aquellos registros desde la dimensión de producto que pertenezcan a incendio, después tomamos de la tabla de hechos los registros que coincidan con los registros obtenidos de la dimensión producto; además filtraremos aquellos registros cuya prima sea mayor o igual a $ 8, 500; este es el primer resultado. Luego, se seleccionan los registros de la tabla de tiempo, que correspondan al mes de diciembre de 2006; éste será el segundo resultado. Ahora son seleccionados sólo aquellos registros del primer resultado, que coinciden con los registros obtenidos de la dimensión de tiempo. Finalmente viajaremos a la dimensión de segmento, para obtener los registros que pertenecen al segmento de líneas personales y éste es el resultado que estábamos esperando.

Independientemente del número de tablas de dimensión involucradas, y de la complejidad de la consulta, cada una es ejecutada, tomando primero los registros de las dimensiones, que coincidan con los filtros definidos como parámetros en la consulta; y luego buscando los registros correspondientes en la tabla de hechos.

2.5. Procesos ETL

Como profesionales de la información, deberemos estar conscientes de que los datos contenidos en los sistemas operacionales de toda organización, son inadecuados para generar información estratégica, que pueda dar un correcto soporte a la toma de decisiones.

En el pasado, hubo intentos por generar información estratégica de forma directa, a partir de los datos provenientes de sistemas operacionales; cabe mencionar que todos estos intentos fueron fallidos. Fue así como nace el Data Warehouse, cuyo objetivo es satisfacer la imperiosa necesidad de información estratégica.

La mayor parte de la información contenida en un Data Warehouse, proviene de los sistemas operacionales, que no pueden ser accedidos de forma directa. Entonces ¿qué hace la diferencia entre los datos operacionales y la información que reside en el Data Warehouse? La respuesta es sencilla, la diferencia la constituyen una serie de funciones y programas que a partir de ahora llamaremos ETL.

Los procesos ETL, obtienen su nombre de las siglas en inglés de Extract, Transform and Load, y constituyen la cimentación para el Data Warehouse. Un diseño apropiado de ETL obtiene datos desde los sistemas operacionales, refuerza los estándares de calidad y consistencia de datos, adapta los datos de tal forma que, fuentes distintas puedan ser utilizadas de manera integrada; y finalmente lleva a cabo el proceso de entrega de información, de una manera tal que los usuarios finales puedan tomar decisiones. [13]

[13] Ralph Kimball & Joe Caserta, The Data Warehouse ETL Toolkit. John Wiley & Sons, U.S.A. 2004. Pág.: 8

2.6. Factores clave

Cada proceso ETL cumple un objetivo específico, ya sea que obtenga los datos desde un sistema de información, convierta los datos aplicando las reglas de negocio que correspondan, o cargue la información al Data Warehouse; cada una de estas funciones es esencial.

El ETL es al mismo tiempo sencillo y complicado. Casi todos comprendemos el objetivo básico de un ETL: sacar datos de un sistema operacional y cargarlos en Data Warehouse. Sin embargo, es necesario limpiar y transformar los datos antes de cargarlos. Es aquí donde reside la parte complicada de la definición, ya que un ETL se subdivide en miles de pequeños casos, dependiendo de cada fuente de información, reglas de negocio, software existente en la organización, y aplicaciones para generar reportes que suelen poco usuales.

Así que el reto del ETL, es asumir los miles de pequeños subcasos, y mantener con claridad la sencilla misión del ETL.

2.7. Extracción de datos

Como profesionales de la información, debemos inevitablemente generar extracciones y conversiones de datos, al implementar sistemas operacionales. Sin embargo, existen dos grandes diferencias, entre el proceso de extracción llevado a cabo al construir sistemas operacionales, y el usado para los Data Warehouse. Primero, para un Data Warehouse generalmente extraemos información desde múltiples fuentes dispares; además deberemos programar cargas incrementales, así como una gran carga inicial de datos. En cambio para un sistema operacional, sólo tendremos que hacer una carga única, y una conversión de datos.

Estos dos puntos, son precisamente los que incrementan la complejidad del proceso de extracción para un Data Warehouse; además de asegurar el uso de herramientas de terceros para la extracción de datos, así como la creación de programas a la medida de las necesidades de la organización.

Las herramientas de terceros, son generalmente más caras que los programas a la medida; adicionalmente generan y administran sus propios metadatos. Por otra parte, los programas a la medida incrementan notablemente el costo de mantenimiento; sin mencionar que son más difíciles de mantener de cara a cualquier cambio que se realice en el sistema utilizado como fuente de datos.

Si en la industria donde trabajamos, los cambios en las condiciones del negocio son comunes, es una buena práctica, reducir el uso de los programas a la medida. Las herramientas de terceros, incorporan una flexibilidad que permite modificar los parámetros de entrada de datos.

Un proceso de extracción efectivo, es la clave del éxito del Data Warehouse; por tanto deberemos poner especial atención al formular la estrategia para nuestro caso particular.

A continuación listamos los elementos más importantes a tomar en cuenta:

Identificación de la fuente – Determinar las aplicaciones, y las estructuras de datos a partir de las cuales obtendremos los datos.

Método de extracción – Para cada fuente de datos, definir si el proceso será manual, o basado en una aplicación.

Frecuencia de la extracción – Para cada fuente de datos, deberemos determinar con qué frecuencia será realizada la extracción: diaria, semanal, mensual, trimestral o así sucesivamente.

La ventana de tiempo – A cada fuente de datos, le corresponde un espacio de tiempo, donde es permisible obtener los datos requeridos; sin afectar el funcionamiento del sistema operacional.
Secuencia de los procesos – Determinar todos aquellos casos, donde un proceso deberá esperar la finalización del anterior.

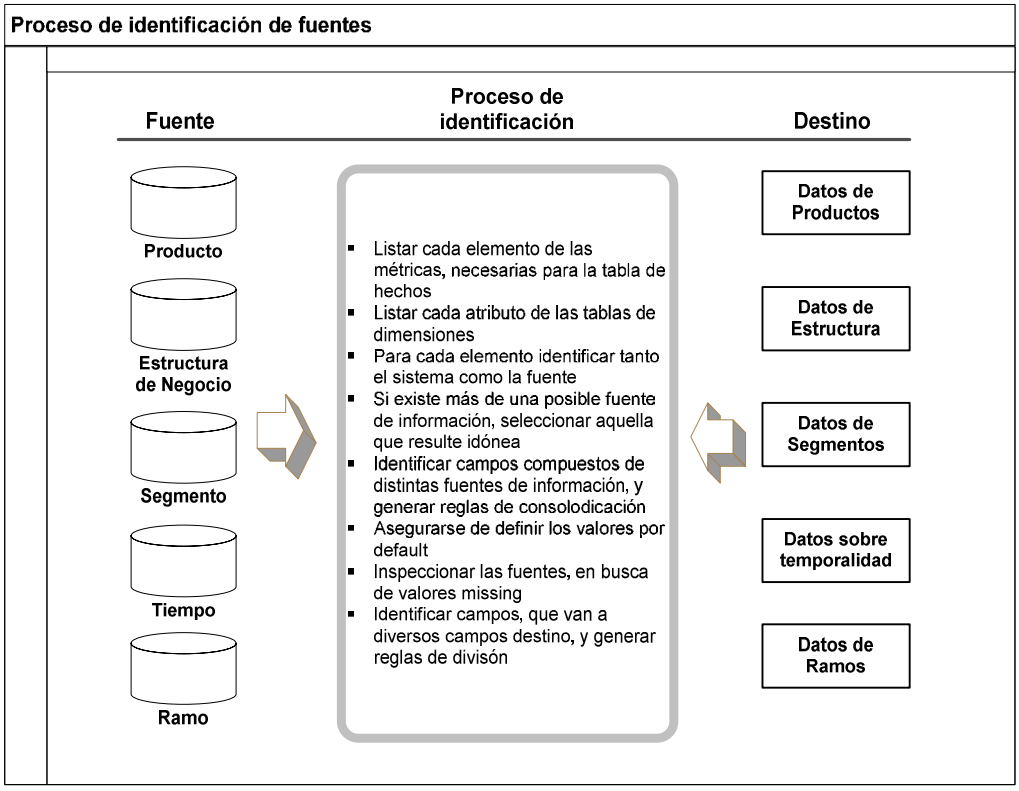
Control de errores – Identificar cuándo se hace necesario, que un proceso se detenga por algún error en los datos que impida su extracción.

2.7.1. Identificando las fuentes

La identificación de las fuentes, no sólo incluye determinar dónde encontrar los datos requeridos; sino que va más allá al verificar que las fuentes seleccionadas, realmente agregarán valor al Data Warehouse.

Veamos un ejemplo: digamos que contamos con un data mart, diseñado para proveer información estratégica para la satisfacción de las exigencias del área de estadística de siniestros de daños. Para tal propósito deberemos almacenar información histórica, referente a los siniestros pendientes, rechazados y pagados. Si tomamos en cuenta que los siniestros, llegan al sistema operacional desde diversos canales, entonces deberemos obtener la información capturada de dichos canales. Si además los usuarios están interesados en analizar los siniestros, a través de estatus; entonces igualmente deberemos incluir esa información en la lista de datos a extraer.

Dentro de la tabla de hechos, incluiremos atributos como el total pagado por siniestro, los descuentos aplicables, los gastos incurridos, reservas, tiempos de atención, así como las distintas fechas, en las que se realizaron los movimientos de pagos, gastos, etc. Debemos establecer las fuentes adecuadas, para cada uno de los rubros que ya mencionamos, y garantizar que dichas fuentes sean las correctas.


Proceso de identificación de fuentes
Fuente
Proceso de identificación
Destino
Producto
Estructura de Negocio
Segmento
Tiempo
Ramo
Listar cada elemento de las métricas, necesarias para la tabla de hechos
Listar cada atributo de las tablas de dimensiones
Para cada elemento identificar tanto el sistema como la fuente
Si existe más de una posible fuente de información, seleccionar aquella que resulte idónea
Identificar campos compuestos de distintas fuentes de información, y generar reglas de consolodicación
Asegurarse de definir los valores por default
Inspeccionar las fuentes, en busca de valores missing
Identificar campos, que van a diversos campos destino, y generar reglas de divisón
Datos de Productos
Datos de Estructura
Datos de Segmentos
Datos sobre temporalidad
Datos de Ramos

para analizar cómo estos cambios afectarán la información que residirá en el Data Warehouse, ya que los datos históricos por ningún motivo, pueden ser ignorados.

Hemos alcanzado el punto donde, deberemos idear un mecanismo que permita capturar la historia de los sistemas fuente; y esto dependerá directamente de la forma como el sistema operacional, almacene la información en las estructura de datos.

De forma general, los sistemas fuente almacenan la información en dos formas:

Valor actual – Casi todos los atributos en un sistema fuente, caen en ésta categoría. En este esquema, el dato almacenado corresponde al valor que tiene en este preciso momento. Los datos son transitorios, ya que irán cambiando en la medida que las transacciones propias del negocio, ocurren. No existe un método con el cuál seamos capaces de predecir, cuánto tiempo el valor presente estará almacenado en las estructuras de datos. El valor se mantendrá constante, hasta que una transacción lo modifique. Es aquí cuando se hace importante prever la extracción de dichos datos, para así asegurar que capturemos los datos históricos.

Estado periódico – Éste esquema no es tan común como el primero. Aquí el valor de un atributo se preserva como un estado, cada vez que ocurre un cambio. Con cada cambio el valor se almacena, con referencia al tiempo en el cuál el nuevo valor toma efecto. Un ejemplo clásico de ésta categoría son los eventos, en el caso de una compañía de seguros podemos referirnos a la vigencia de una póliza, así como a la como a la ocurrencia de un siniestro, la fecha en la que fue atendido, la fecha de los pagos relacionados, etc.

En el siguiente esquema (Figura 2.20), podemos ver un ejemplo de cómo se comportan los sistemas, que funcionan con los esquemas de almacenamiento que vimos antes.

Esquemas de almacenamiento

Atributo: Estado de residencia del cliente

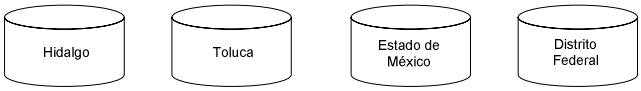
Almacenando el valor actual

06/03/2006 Valor: Hidalgo

15/06/2006 Valor: Toluca

10/10/2006 Valor: Estado de México

01/12/2006 Valor: Distrito Federal



Atributo: Estado del siniestro

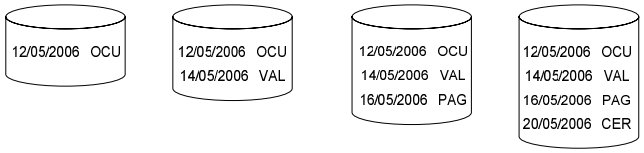
Almacenando por estados periódico

12/05/2006 Valor: Ocurrencia del siniestro

14/05/2006 Valor: Valuación del siniestro

16/05/2006 Valor: Pago del Siniestro

20/05/2006 Valor: Siniestro Cerrado



Captura a través de Log's de transacción – Esta opción hace uso de los archivos log's, de transacciones generados por el sistema administrador de la base de datos, DBMS por sus siglas en inglés. El propósito de dichos archivos, es tener una manera de recuperar los datos, en caso de que se presente alguna falla. Conforme una transacción agrega, elimina o actualiza registros en las bases de datos, el archivo de Log, registra cada uno de estos cambios. La técnica se basa en leer los archivos de log generados, y seleccionar aquellas transacciones que han sido completadas.

La ventaja, es que no hay una sobrecarga de los recursos, ya que la generación de éstos archivos log's, son parte del DBMS. Sin embargo, si es necesario que estemos atentos para asegurar que los cambios han sido debidamente capturados, antes de que el archivo de log sea traspasado a otro dispositivo de almacenamiento; ya que típicamente los archivos de log crecen bastante, en ese momento son traspasados a cintas que son reutilizadas con una periodicidad determinada; si no realizamos la captura de las transacciones en el tiempo adecuado, corremos el riesgo de perder para siempre esos datos históricos.

El uso de esta técnica es adecuado, cuando todos los sistemas fuente son DBMS; en aquellos casos donde obtengamos información de archivos indexados, archivos planos o en otro formato, esta opción no podrá ser aplicada para nuestros propósitos.

Captura a través de triggers de la base de datos – los triggers son programas que están almacenados en la base de datos, y son "disparados" cuando ciertos eventos predefinidos ocurren. La idea es que nosotros generemos programas, para cada uno de los eventos que deseamos captar, escritura, actualización o borrado de registros. La salida de los triggers, será almacenada en un archivo a parte, el cual utilizaremos para realizar el traspaso de los datos al Data Warehouse.

Dado que los programas residen y son ejecutados desde la fuente, son procedimientos relativamente confiables. Las implicaciones de generar programas triggers, es que se agrega una carga adicional a los procesos del DBMS, además de incrementar las horas de mantenimiento de estos procesos, dentro del esfuerzo de desarrollo del Data Warehouse. Finalmente debemos aclarar que ésta opción, es sólo aplicable a sistemas DBMS.

Captura en aplicaciones fuente – Esta técnica se refiere a procesos de captura asistida, es decir, la aplicación para captura de datos, está diseñada para auxiliar al Data Warehouse, al identificar los cambios que ocurren en las bases de datos del sistema fuente. A diferencia de las técnicas mencionadas anteriormente, ésta técnica puede aplicarse a todos los casos, independientemente de la base de datos, archivos indexados, o incluso archivos planos.

El uso de esta técnica puede llegar a ser muy complicada, si el número de sistemas fuente en grande; además también implica un procesamiento en segundo plano, que agregará una mayor carga, afectando el desempeño.

Captura diferida – Todos las técnicas que hemos visto hasta el momento, suceden mientras la transacción ocurre en el sistema operacional. En estos casos la captura es inmediata, o en tiempo real. Las técnicas que discutiremos a continuación, son distintas ya que no capturan los cambios en tiempo real, ésta ocurre después. Dentro de la captura diferida, existen dos técnicas:

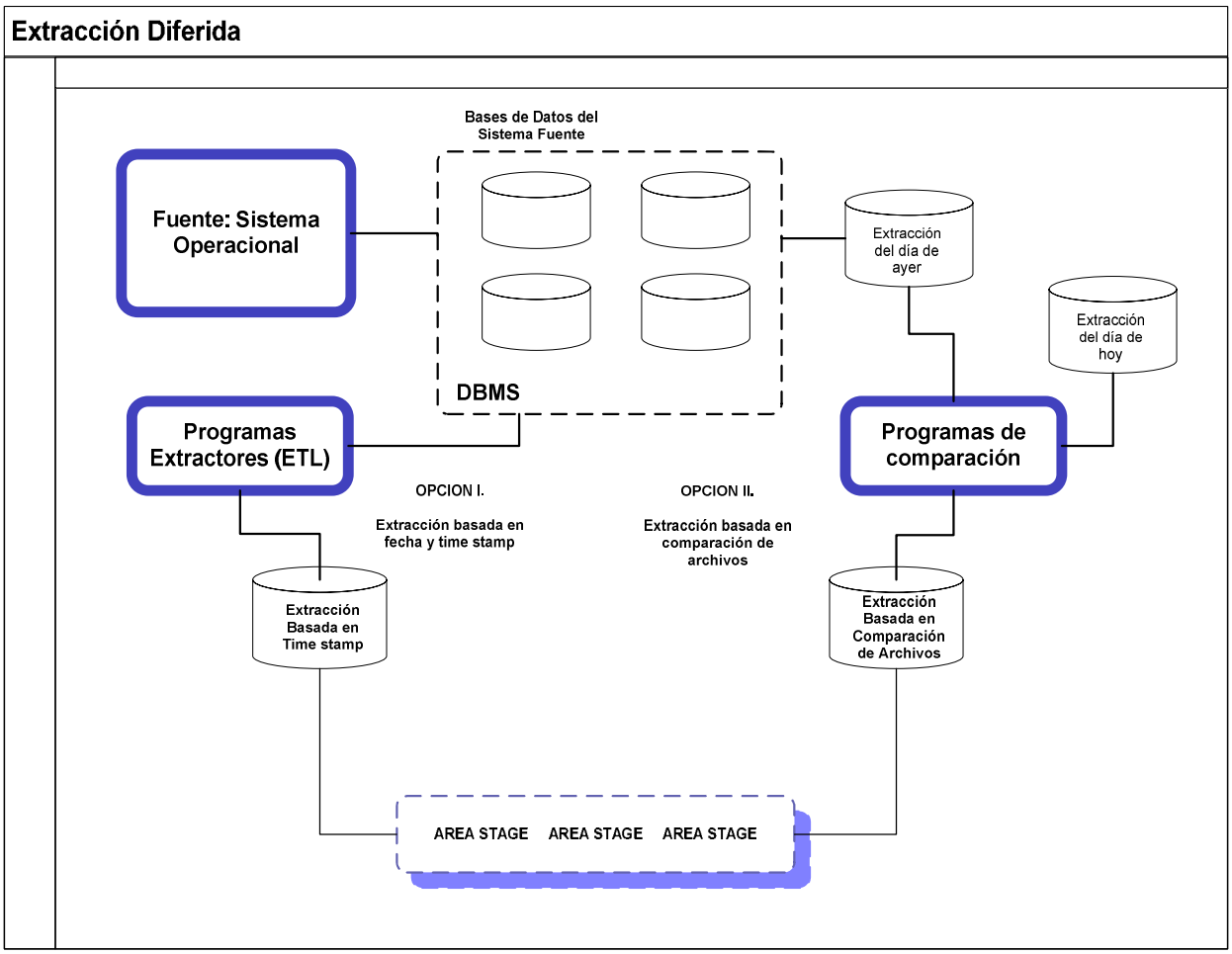
Captura basada en fecha y time stamp – Cada vez que un registro es creado en el sistema fuente, puede ser marcado con una "etiqueta de tiempo" que nos dirá la fecha y hora de creación de dicho registro. De este modo el time stamp, puede convertirse en un filtro básico que nos ayudará a seleccionar un grupo determinado de registros, para las extracciones.

Si nosotros ejecutamos los programas extractores, todos los días a la media noche, estaremos obteniendo sólo aquellos registros, creados después de la media noche del día anterior. La técnica funciona muy bien, cuando el número de registros es reducido; y desde luego que presupone que los registros tendrán un time stamp. Sin embargo es importante tener en cuenta, que sólo tomaremos la última situación de cada registro, cualquier cambio intermedio entre dos extracciones, se perderá y no pasará al Data Warehouse.

El borrado de registros representa un problema especial, ya que si un registro es eliminado entre dos extracciones, la información del registro borrado no será detectada. En estos casos lo que podemos hacer es generar una marca de borrado, es decir agregar un indicador extra en las tablas de las bases de datos, proceder con la carga, para finalmente eliminar físicamente aquellos registros marcados.

Estos último significa, que deberemos agregar la lógica necesaria, a los programas extractores.

En la figura siguiente (Figura 2.21), podemos observar la forma como trabajan ambas opciones de la extracción diferida:



Captura por comparación de archivos – Esta técnica se usa, cuando ninguna de las anteriores es factible; por ello ésta es considerada el último recurso disponible. Consiste en tomar dos "fotografías" de los datos en el sistema fuente. Digamos que deseamos aplicar ésta técnica, para capturar los cambios realizados a los datos sobre productos. Entonces mientras realizamos la extracción del día de hoy, realizaremos una comparación completa entre la copia de la extracción del día de ayer y la del día de hoy, comparando las llaves de los registros, y así identificar si se han agregado, modificado o borrado datos; es entonces cuando capturamos los cambios y lo pasamos al Data Warehouse.

La desventaja es el espacio, ya que requerimos almacenar copias previas de los procesos de carga; adicionalmente una comparación directa entre dos archivos, parece no ser del todo eficiente, pero ésta opción nos será de utilidad al acceder a fuentes de sistemas legacy, que no cuentan con log's de transacciones, o time stamps.

2.8. Transformación de datos

En este punto, los datos que hemos recolectado desde los sistemas fuente, no pueden ser aplicados de forma directa al Data Warehouse. Primero deberemos hacerlos utilizables, como hemos visto en secciones anteriores, los datos provienen de sistemas operacionales diversos, y por ello la calidad de los datos, puede no ser la adecuada para tener información de soporte a la toma de decisiones, que como también hemos discutido, constituye el principio básico del Data Warehouse.

Para enriquecer e incrementar la calidad de los datos, deberemos hacer varias transformaciones, basándonos en estándares, ya que los datos provienen de sistemas operacionales distintos debemos asegurar que, una vez que unamos los datos, no estemos violando ninguna de las reglas de negocio definidas.

Debemos ser conscientes del hecho que la transformación de datos, es una tarea que involucra análisis al encargarse de la conversión y reformateo de los datos; así que no cometamos el error de subestimar el tiempo necesario para esta importante función.

Los estudiosos del Data Warehouse, han intentado crear una clasificación para las funciones de transformación de datos; sin embargo existe un confusión en cuanto a si se debe incluir el proceso de integración de datos dentro de las tareas de transformación, o si debemos considerarlo como un proceso previo a la transformación. Al margen de ésta confusión, en general se han clasificado las funciones de transformación en simples y complejas.

2.8.1. Tareas básicas

Independientemente de la variedad y complejidad de los sistemas operacionales, utilizados como fuente de datos, encontramos que las funciones de transformación se dividen en un conjunto de tareas básicas:

Selección – La primera tarea a llevar a cabo, es donde tomamos la decisión de seleccionar todos o sólo algunos de los campos desde el sistema fuente. Ésta tarea forma parte del proceso de extracción, sin embargo es recomendable, realizar la extracción de todos los campos primero, y posteriormente hacer la selección dentro del proceso de transformación.

Unión / División – Incluye la manipulación que realizaremos sobre los datos que fueron seleccionados a través de proceso anterior. De forma poco común, será necesario a este punto realizar una división de los datos; incluso en ésta etapa; pero por lo general, el proceso más comúnmente utilizado es la unión de datos, que provienen de diversos sistemas operacionales.

Conversión – Los dos objetivos principales de ésta tarea, son estandarizar los datos, y hacerlos utilizables y entendibles para los usuarios de la información. Aquí incluiremos una gran variedad de conversiones rudimentarias de datos, así como homologaciones.

Sumarización – En ciertos casos, almacenar la información con el nivel mínimo de detalle, no resulta la mejor práctica, ya que los usuarios pueden no requerir ese nivel de información para sus consultas. Por ejemplo, si los usuarios deben saber el volumen de ventas de pólizas por producto, no necesitan conocer el detalle de cada transacción en cada póliza emitida. Es aquí cuando la función de transformación de datos, incluye sumarizar la información.

Enriquecimiento – Consiste en reorganizar y simplificar campos individuales, para hacerlos más útiles para el ambiente de Data Warehouse. Podríamos utilizar, por ejemplo, dos o más campos del mismo registro de entrada, para crear un solo campo en el Data Warehouse, y así dar una mejor vista de la información.

2.8.2. Tipos de transformación

Hasta el momento hemos discutido, cuáles son las tareas básicas del proceso de transformación. Al considerar un conjunto específico de datos, para ser integrados al Data Warehouse; deberemos hacerlo mediante una combinación de las tareas básicas. A continuación veremos a detalle los diferentes tipos de funciones de transformación más comúnmente utilizadas:

Revisión de formatos – Son revisiones que incluyen cambios en los tipos de datos, así como en las longitudes de campos individuales. Un ejemplo clásico, es cuando deseamos unir la información que proviene de varios sistemas operacionales; por ejemplo el código de producto, puede tener distintos tipos y longitudes de datos, aquí es importante estandarizar y cambiar, según sea necesario, el tipo de dato para utilizar el formato más adecuado.

Decodificación de campos – Otro tipo de transformación bastante común, donde deberemos decodificar el significado de campos específicos, y transformarlos en valores que sean más descriptivos para los usuarios. Como ejemplo el género de una persona, almacenada en forma de código dentro del sistema operacional; digamos que tenemos ‘01’ para hombre y ‘02’ para mujer. De tal forma que el Data Warehouse, aplicaremos una transformación, cambiando estos códigos por una descripción ‘Hombre’ / ‘Mujer’ según corresponda.

Valores calculados y derivados – Datos económicos que sean integrados al Data Warehouse, pueden participar en cálculos, y ser almacenados. Ejemplos de este tipo de campos, son los promedios, y balances diarios.

División de campos únicos – Estos se refieren a datos de sistemas legacy, que son almacenados en campos de texto muy grandes. Por ejemplo, el nombre y apellidos de los clientes, almacenados en un único campo tipo texto. En algunos casos es mejor separar los campos, para mejorar el desempeño durante las consultas, además de facilitar los análisis que requieren los usuarios.

Fusión de campos – A pesar de lo que pudiera pensarse, ésta función no es la opuesta a la división de campos únicos. Se trata de una combinación de códigos y descripciones que serán fusionados dentro de un campo.

Conversión de conjunto de caracteres – Este conversión es útil, cuando trabajamos con sistemas fuente que residen en Main Frame. De tal modo, que si el Data Warehouse está basado en una arquitectura PC, deberemos convertir del conjunto de caracteres EBCDIC al ASCII.

Conversión de unidades de medida – En la actualidad muchas compañías cuentan con representaciones en otros países, si es el caso de la organización para la que trabajamos, deberemos entonces convertir las unidades de medida, para asegurar que las unidades están todas en un mismo sistema.

Conversión de fechas/horas – Este tipo de conversión se refiere, a representar las fechas y las horas en un formato estándar, un formato internacional estándar puede ser: 12 OCT 2006.

Sumarización – Este tipo de transformación, consiste en crear datos resumidos para ser cargados al Data Warehouse, en lugar de almacenar los datos con el mínimo nivel de granularidad.

Reestructuración de llave – Mientras extraemos datos desde los sistemas fuente, demos un vistazo a las llaves de los registros extraídos. Deberemos basarnos en éstas llaves, para a su vez generar las llaves de para la tabla de hechos y las tablas de dimensión. Recordando lo visto en secciones anteriores, debemos reestructurar las llaves, para evitar el uso de significados inherentes.

Depuración de duplicados – En muchas organizaciones, existen referencias duplicadas de un mismo registro. Por ejemplo en la tabla de clientes, podemos llegar a tener varios registros para un mismo cliente; la mayor parte de estos registros son generados por error. Dentro del Data Warehouse debemos asegurarnos que exista sólo un registro por cliente, evitando así los duplicados.

2.8.3. Consolidación e Integración de datos

El verdadero reto en el ETL es el lograr unir los datos que provienen de diversos sistemas fuente, que por supuesto tienen estructuras de datos, códigos y reglas distintas. Los Data Warehouse más actuales, toman información desde sistemas legacy en Main Frame, y desde aplicaciones más modernas que tienen una arquitectura cliente-servidor.

La mayoría de estos sistemas, no tienen definiciones uniformes de reglas de negocio, además que típicamente, tienen estándares de nomenclatura y representación de datos muy distintos entre sistemas fuente. En este punto, podemos clasificar al proceso de integración y consolidación de datos, como un preproceso para las rutinas de transformación. Es decir, que en este punto, primero que nada deberemos estandarizar los nombres y representaciones, así como resolver las discrepancias que surgen de las distintas formas de representación de los datos en los distintos sistemas fuente.

Ahora discutiremos los problemas, a los que generalmente nos enfrentamos al realizar las tareas de consolidación e integración de datos. El primero es la identificación de las entidades, luego tendremos que lidiar con las múltiples fuentes.

Identificación de Entidades. Esto sucede cuando dentro de la organización contamos con dos o más sistemas legacy, donde cada uno contará, por ejemplo, un archivo de clientes. Tal vez uno de los sistemas fue creado para dar soporte a los pedidos, otro más podría ser par servicio al cliente, y el último puede estar diseñado para campañas de marketing. La mayoría de los clientes, estarán almacenados en cada uno de los sistemas, cada una de las bases tendrá un número único de identificación para el cliente; el problema es que este número único no es común a los tres distintos sistemas. Ocasionando que registros con números de identificación distintos se refieran al mismo cliente.

Al margen de esta situación, el Data Warehouse deberá almacenar un único registro por cliente. Para ello deberemos diseñar programas que identifiquen y construyan un único registro, a partir de la información relacionada a un mismo cliente. Este problema es muy común cuando trabajamos con entidades de datos, que son comunes a diversos sistemas fuente. Ejemplos de entidades propensas a generar este tipo de problema son: Vendedores, Proveedores, Empleados, Clientes e incluso Productos.

Debemos encontrar un equilibrio al diseñar el algoritmo de identificación, ya que si los criterios usados son demasiado específicos, corremos el riesgo de perder información relacionada a un mismo cliente, por el contrario, si los criterios son muy generales aquí el riesgo, es incluir información de más de un cliente, asignando dichos datos a un único cliente, lo cuál constituye un error severo.

En algunos casos, y dada la complejidad del proceso de identificación; algunos recomiendan llevar a cabo el proceso en dos fases: la primera todos los registros, independientemente de si están o no duplicados, le son asignados identificadores únicos. La segunda fase consiste en realizar una conciliación de forma periódica, a través de algoritmos automáticos, y también con verificaciones manuales.

Múltiples Fuentes. Este es otro tipo de problema que afecta la integración de datos, dentro del DW, menos común y menos complejo que el anterior. Esto sucede cuando contamos con un único elemento de datos, se encuentra en más de una fuente. Por ejemplo, vamos a suponer que el costo de los productos que la organización ofrece a sus clientes, están almacenados en dos distintos sistemas. En la aplicación estándar de costos, los precios de las unidades se calculan y actualizan, a intervalos regulares. Y el sistema que da soporte al proceso de pedidos, también contiene la información del costo por unidad. Sin embargo, pueden existir pequeñas diferencias entre estos dos sistemas. ¿Desde qué sistema debería obtener la información el DW?

Una solución muy directa, sería el otorgar una prioridad mayor, a uno de éstos sistemas, y obtener de éste el costo de las unidades por producto. En algunas ocasiones, ésta solución no satisface las necesidades del cliente, entonces deberemos considerar otros opciones, como por ejemplo basarnos en aquellos que tengan la fecha de actualización más reciente. Finalmente, puede definirse n criterios para llevar a cabo la actualización de la información; y éstos dependerán la situación particular en la que nos encontremos.

2.8.4. Implementando la transformación

La complejidad y extensión de las transformaciones de datos, sugiere que el uso de métodos manuales, no es suficiente. Debemos ir un paso adelante, y generar programas de conversión. Los métodos que adoptemos dependerán de factores clave: si estamos considerando automatizar la mayor parte de las tareas de transformación, entonces debemos tener en cuenta si contamos con el tiempo

necesario para evaluar, seleccionar, instalar y probar herramientas de transformación comerciales, que además pueden ser costosas. Si el proyecto de DW en el que estamos, tiene un alcance modesto, así como un presupuesto limitado, el uso de éstas herramientas será prohibitivo.

Por otra parte el uso de técnicas manuales, puede no ser la mejor opción. El encontrar un equilibrio entre ambos métodos, ha probado ser más efectivo. Permitiéndonos generar una solución en un tiempo razonable, y a un costo competitivo y razonablemente apegado al presupuesto.

Uso de herramientas comerciales. En años recientes, las herramientas de transformación comerciales, han incrementado su flexibilidad y funcionalidad. No obstante, sin importar cuán sofisticada sea una herramienta comercial, nunca podrá eliminar por completo la necesidad del uso de técnicas manuales. Es cierto que el uso de estas herramientas, incrementan la eficiencia y precisión, ya que sólo debemos establecer los parámetros, definiciones de datos y reglas de transformación, que la herramienta estará utilizando; y con ello la herramienta hará el resto del trabajo de forma muy eficiente.

Es aquí donde obtenemos una ventaja adicional, ya que las herramientas de este tipo, generan y administran sus propios metadatos. Dichos metadatos pueden ser reutilizados por todo el DW. Cuando ocurre un cambio en las funciones de transformación, debido a cambios en las reglas de negocio o bien cambios en la definición de datos del negocio; sólo deberemos actualizar los parámetros en la herramienta, y los metadatos se ajustarán de forma automática reflejando dichos cambios.

Uso de técnicas manuales. Este ha sido el método predominante, hasta la aparición de las herramientas comerciales; sigue siendo un método idóneo para Data Warehouses pequeños, en donde programas y scripts codificados a mano, llevan a cabo todas las tareas de transformación. Desde luego que el uso de éste método, implica un esfuerzo de desarrollo y periodo de pruebas; así mismo las tareas de mantenimiento de dicho programas, probablemente incrementen el costo del proyecto.

La desventaja de éste método, está relacionada con los metadatos, ya que si ocurre algún cambio en las reglas del negocio, esto nos obligará a cambiar los códigos de los programas, agregando así una carga adicional al esfuerzo del desarrollo.

2.9. Carga de datos

El siguiente grupo de funciones, está constituido por aquellas que toman los datos que han sido preparados, por las funciones de transformación y los transportan y almacenan en el Data Warehouse. Es en este punto donde se crean las imágenes de carga, que son archivos que corresponden a los layout objetivo que residen en las bases de datos dentro del DW.

El proceso de cargar datos en el repositorio del DW, es conocido como aplicación de datos, carga de datos, o incluso refresco de datos. Para mantener la claridad, nosotros estaremos usando los siguientes conceptos:

Carga inicial – Consisten poblar las bases de datos del DW, por primera vez.

Carga incremental – Aplicar los cambios que ocurran, de forma periódica.

Refresco completo – consiste en borrar todos los datos contenidos en las bases del DW, y cargarlas de nuevo con datos frescos.

Debido a que las cargas de datos al DW, pueden llevar una cantidad de tiempo bastante considerable, son de gran importancia, ya que durante este proceso el DW deberá estar fuera de línea, por lo que debemos encontrar una ventana de tiempo adecuada, para no afectar a los usuarios del DW.

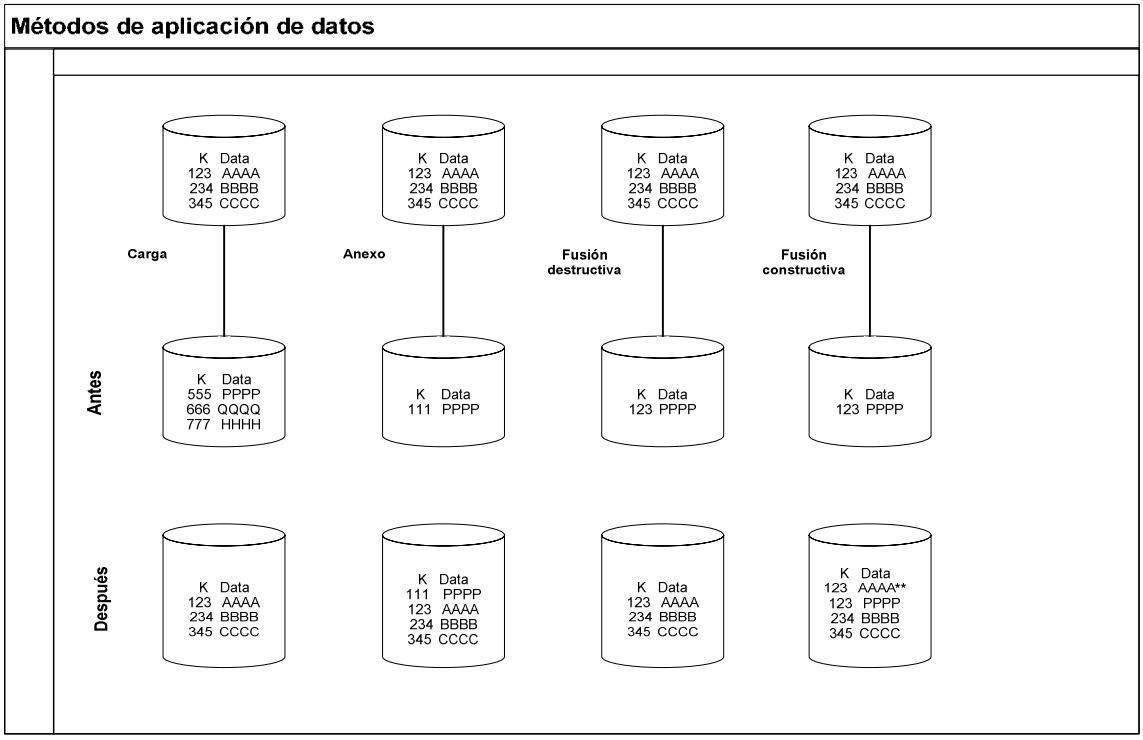
Por esta razón, es razonable considerar, dividir el proceso de carga en varias secciones, cargando pocos archivos a la vez, obteniendo así el beneficio de tener, varios procesos de carga en paralelo acortando el tiempo que el DW estará fuera de línea. Otra beneficio de dividir el proceso de carga, es mantener en operación algunos módulos del DW, mientras otros son cargados con datos frescos.

Se recomienda hacer ciclos de prueba, para asegurarnos de que todo funciona como debe, además de que estas pruebas nos darán una buena idea de cuánto podrían durar las cargas. Otro importante tema a tener en mente, es contar con un plan de aseguramiento de la calidad de los datos cargados.

2.9.1. Técnicas y procesos

Previamente comentamos los tres tipos de aplicación de datos: carga inicial, carga incremental y refresco completo. En cada uno de estos casos, creamos archivos de carga que serán aplicados en las diversas tablas que residen en el DW. ¿Pero cómo podemos realizar el proceso de carga?

Existen cuatro diferentes formas de aplicar los datos: carga, anexo, fusión destructiva, fusión constructiva. En la siguiente figura (Figura 2.22) se muestra el funcionamiento de cada uno de éstos métodos:



Carga – Si la tabla a ser cargada ya existe, y contiene datos; el proceso de carga sobre escribe los datos y aplicará los datos de la carga actual. Si la tabla se encuentra vacía, el proceso simplemente aplicará los datos de la carga.

Anexo – Para entender mejor cómo funciona el anexo, podemos definirlo como una extensión de la carga, si un registro ya existe en la tabla, el proceso de anexo de forma incondicional agregará los datos preservando los datos ya existentes en la tabla destino. Si existieran duplicados, podremos definir reglas para tratar el registro duplicado; es decir, podemos elegir entre añadir el registro como duplicado, o rechazarlo.

Fusión destructiva – En este método se aplican los datos entrantes, si la llave primaria del registro entrante coincide con la llave primaria del registro existente, el proceso realizará la actualización de dicho registro. Si el registro tiene una llave primaria sin coincidencia con alguno de los registros existentes, simplemente se añadirá como un nuevo registro.

Fusión constructiva – Este método es ligeramente distinto a la fusión destructiva, y se diferencia en que, si la llave primaria de un registro entrante coincide con la llave primaria de un registro existente en la tabla, agrega el registro entrante y al mismo tiempo conserva el registro preexistente y finalmente marca el nuevo registro.

2.9.2. Tipos de Actualización (Refresh vs. Update)

Una vez realizada la carga inicial del DW, los datos pueden mantenerse actualizados, utilizando cualquiera de los dos métodos disponibles:

Actualización – Se aplican de forma incremental, los datos que han sufrido cambios en los sistemas fuente.

Refresco – Recarga completamente la información a intervalos específicos de tiempo.

Técnicamente, el refresco de información es una opción mucho más fácil, que la actualización. Si elegimos la actualización, primero deberemos visualizar y determinar la mejor estrategia para aplicar los cambios al DW, identificando los cambios que sucedan en cada una de las fuentes. En caso de utilizar el refresco de información, debemos tener en mente que el proceso, llevará mucho más tiempo que la actualización, además de que cuando trabajamos con tablas grandes, el tiempo para la actualización se vuelve inaceptable.

Existe una especie de guía, que nos puede indicar cuándo es mejor realizar una actualización, y cuán es mejor realizar un refresco. En la siguiente figura (Figura 2.23), podemos observar de forma gráfica esta guía:

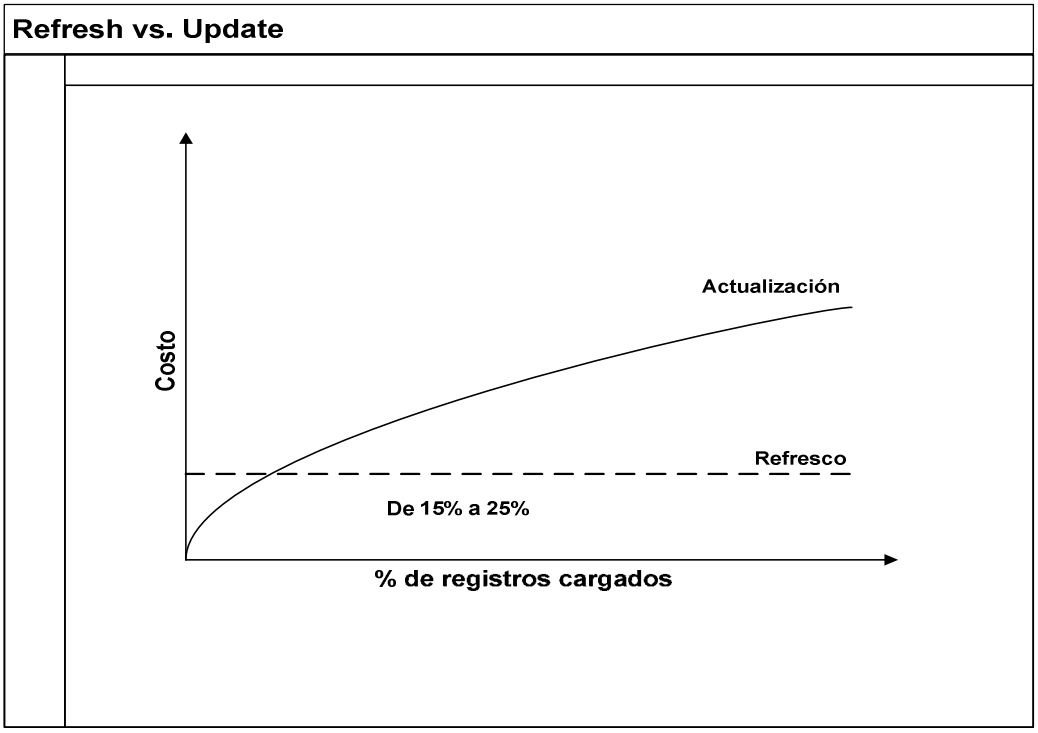


Figura 2.23. Refresh vs. Update

2.9.3. Procesos para tablas de dimensiones

En un Data Warehouse, las tablas de dimensiones contienen atributos que son utilizados para analizar, métricas básicas, como por ejemplo los costos por producto. El proceso de dar mantenimiento a tablas de dimensión, está dividido en dos partes: primero la carga inicial de datos, y posteriormente la aplicación de los cambios, para lo cual debemos poner atención en dos temas importantes:

El primero tiene que ver con las llaves de los registros, en los sistemas fuente y las llaves dentro del DW. Por razones que ya comentamos anteriormente, dentro del DW, no haremos uso de las llaves productivas de los sistemas operacionales. En su lugar, generaremos llaves de sistema.

Por lo cual es necesario hacer la conversión, de las llaves del sistema operacional a llaves generadas, sin importar si se trata de una carga inicial o incremental; antes de cargar los datos a las tablas de dimensión.

Se posible llevar a cabo la conversión de las llaves, ya sea durante el proceso de transformación o bien, en un proceso separado, pero es preferible que se lleve a cabo previo a la carga de datos.

2.9.4. Procesos para tablas de hechos

Como vimos en secciones anteriores, una tabla de hechos es dónde almacenaremos todas las métricas de interés, para los analistas de negocio. Por ello es recomendable almacenar datos a un nivel de detalle lo más bajo posible. Por ejemplo, en el área de cobranza de una compañía aseguradora, deberíamos almacenar el detalle a nivel de recibo y cobertura.

En algunos casos, existen tablas de hechos que sólo almacenan información resumida, en cuyo caso son llamadas tablas de hechos agregadas.

A continuación revisaremos las características de una tabla de hechos:

Llave concatenada. Un renglón en una tabla de hechos, está relacionado con una combinación de renglones de todas las tablas de dimensión que compongan el modelo multidimensional. Por ejemplo, un modelo con las dimensiones tiempo, póliza, cliente y agente. Ahora si asumimos que el mínimo nivel de la jerarquía de las dimensiones son un número de póliza, una fecha del calendario, un cliente específico, así como un agente determinado. Entonces un único renglón en la tabla de hechos debe relacionarse a un número de póliza en particular, a una fecha específica, a un cliente y a un agente.

En otras palabras, un único renglón en la tabla de hechos, estará identificado por una llave primaria compuesta, generada a partir de la concatenación de las llaves primarias de cada una de las tablas de dimensión relacionadas.

Granularidad. Esta es una de las características más importantes, como sabemos la granularidad no es otra cosa más que el nivel de detalle que tendrán las métricas almacenadas en la tabla de hechos. Por ejemplo, el importe de prima emitida, estará relacionado a una póliza particular, a una fecha específica, a un cliente y agente, respectivamente. Si en cambio, decidimos almacenar el importe de prima emitida por bimestre, entonces tendremos una granularidad a un nivel más alto, es decir, más resumido.

Métricas aditivas. Si tomamos algunos de los atributos que componen a la tabla de hechos, como la prima emitida, la prima devengada, y el importe de siniestro ocurrido. Cada uno de estos atributos está relacionado con una póliza, con una fecha, con un cliente y un agente. Si nosotros deseamos obtener el total de éstas métricas, pero no por un cliente en específico, sino para todos aquellos clientes que vivan en un estado de la república, entonces necesitamos obtener con una consulta todos los clientes que vivan en el estado especificado, agrupar y sumarizar las métricas de prima emitida, prima devengada y siniestro ocurrido. El resultado puede obtenerse por una simple adición, a esto le llamamos métricas aditivas.

Cuando hagamos una consulta, debemos asegurarnos que las métricas pueden sumarse por simple adición; de lo contrario corremos el riesgo de que el total obtenido no sea el correcto.

Métricas semiaditivas. Una póliza puede ser vendida por uno o varios agentes, en cuyo caso cada agente tendrá un porcentaje de participación determinado. Si requerimos obtener el total de prima emitida por cada uno de los agentes, no podremos obtenerlo por simple adición, sino debemos realizar una conversión, multiplicando la prima emitida total por el porcentaje de participación de cada uno de los agentes. Este tipo de métricas que no pueden obtenerse por simple adición, se conocen como semiaditivas.

Tabla profunda, no amplia. Por lo general una tabla de hechos contiene menos atributos que una tabla de dimensión. Sin embargo el número de registros en una tabla de hechos, es mucho mayor que en una dimensión. Por ello decimos que es una tabla profunda, pero no amplia. Muchos registros y pocos atributos, generalmente 10 ó menos. De una manera muy simple, si consideramos 10 pólizas cada una con 5 coberturas, y los pagos a lo largo de 30 días, podemos hablar de una tabla de hechos que crece a un ritmo de 1,500 registros al mes, aproximadamente.

Datos esparcidos. Hemos comentado cómo se relacionan los registros de una tabla de hechos con cada una de las dimensiones que componen el modelo multidimensional. Ahora bien, pueden darse casos en los que una tabla de hechos contenga valores nulos, para ciertos registros o renglones. Por ejemplo, si realizamos una consulta sobre los pagos realizados, cuando la fecha coincida con un día no hábil, entonces tendremos valores nulos.

Dimensión degenerada. Cuando tomamos atributos que vienen desde los sistemas operaciones, y los hacemos viajar hasta las tablas de dimensión y/o tablas de hechos, el resultado será que tendremos elementos de datos, que no son estrictamente métricas, ni tampoco son estrictamente atributos propios de una dimensión. Ejemplos de este tipo particular de dato son: número de póliza, número de recibo, número de referencia, número de siniestro, etc. Este tipo de datos resulta muy útil en cierto tipo de análisis. Por ejemplo, si deseamos calcular el promedio de coberturas por póliza, debemos ligar el número de póliza y los códigos de cada una de sus coberturas, para entonces obtener el promedio.

Como ya hemos comentado la llave primaria de una tabla de hechos, está compuesta por la concatenación de las llaves primarias de las distintas tablas de dimensión. Por consecuencia, los registros de las tablas de dimensiones deberán ser cargados primero, una vez hecho esto, procederemos a generar la llave concatenada de cada uno de los registros, para finalmente cargarlos en las tablas de hechos.

A continuación listamos algunos consejos prácticos, a la hora de cargar tablas de hechos:

- Identificar los datos históricos, que son de interés para el DW
- Definir y refinar las reglas de negocio
- Generar estadísticas de carga, para asegurar que todos los registros leídos han sido cargados
- Llevar a cabo una búsqueda con llaves sustitutas, para verificación
- Afinar el contenido de la tabla de hechos

Capitulo 3

3. Tecnologías y tendencias

Para la mayoría de las organizaciones, el DW es algo más que un simple repositorio, que provee información, para satisfacer las necesidades de información en cada área o departamento, dentro de la organización. Un Data Warehouse es el cimiente de la arquitectura de lo que ahora se conoce con el nombre de inteligencia de negocios o BI por sus siglas en inglés (Business Intelligence).

La inteligencia de negocios es un concepto que incluye al conjunto de estrategias, y herramientas enfocadas a la administración y creación de conocimiento; mediante el análisis de datos existentes en una organización.[14]

Este conjunto de herramientas y metodologías, comparten características comunes, que listamos a continuación:

Accesibilidad – Debe garantizarse que los datos sean accesibles, independientemente de la procedencia de éstos.

Apoyo en la toma de decisiones – Se busca ir más allá de la presentación de la información, de manera que los usuarios, cuenten con herramientas que les permitan seleccionar y manipular sólo aquellos datos que les sean relevantes.

Orientación al usuario final – Las herramientas deben ser accesibles a todos los usuarios, sin importar su nivel de experiencia.

De acuerdo a su nivel de complejidad, las soluciones para BI pueden clasificarse en:

- Consultas y reportes
- Cubos OLAP
- Minería de datos
- Sistemas de previsión empresarial

El business intelligence, puede mejorar el desempeño a nivel corporativo, de cualquier organización que utilice de forma intensiva la información. Las organizaciones pueden potenciar las relaciones con sus clientes y proveedores; así como incrementar el margen de ganancia de los productos y servicios que ofrecen, al generar nuevas y atractivas ofertas. Adicionalmente pueden llevar a cabo, una mejor administración del riesgo, y una dramática disminución en los costos.

Contar con una inteligencia adecuada, significa tener respuestas definitivas, a preguntas como:

¿Cuál de nuestros clientes es más rentable, y cómo podríamos extender nuestra relación con él?
¿Cuáles de nuestros clientes, nos representan ganancias, o gastos?
¿Qué tan cerca viven nuestros mejores clientes, de nuestros almacenes/oficinas?
¿Cuáles de nuestros productos, pueden venderse de forma más efectiva y a quiénes?

[14] Ralph Kimball, The Data Warehouse Toolkit. John Wiley & Sons, U.S.A 2002. Pág. 393.

Breve Historia del BI

Antes de que se iniciara la llamada “era de la información”, las organizaciones debían obtener datos, de fuentes no automatizadas. Las organizaciones carecían de recursos y de equipo de cómputo, para analizar los datos. Como consecuencia, las organizaciones realizaban la toma de decisiones, basándose en la intuición.

A medida que las organizaciones, iniciaron el proceso de automatización, mediante los sistemas de cómputo, más datos estuvieron disponibles para el análisis. Sin embargo en este punto, el reto mayor era el intercambio de datos entre distintos sistemas; debido a la carencia de infraestructura. De tal modo que la adquisición de datos, y la generación de reportes, podía tomar varios meses para completarse; por lo cuál la información sólo era útil para la toma de decisiones a largo plazo; dejando nuevamente la toma de decisiones en el corto plazo, en el reino de la intuición.

Sin embargo, las organizaciones actuales, que cada vez más aplican estándares, automatización de procesos, y nuevas tecnologías; generan una gran cantidad de información disponible para el análisis. Adicionalmente, tecnologías como el OLAP, permiten la generación de reportes en un tiempo considerablemente más corto.

Por todo esto, el BI se ha convertido en el arte de “cribar” grandes volúmenes de datos, para extraer sólo la información relevante, para entonces convertirla en conocimiento, sobre el cuál basaremos nuestras acciones.

El software para BI incorpora capacidades de minería de datos, análisis y reportes. Incluso algunas herramientas más sofisticadas, permiten realizar análisis cruzados, y búsquedas mucho más rápidas, para medir el desempeño de una o varias áreas, de forma agrupada o individual.

El futuro del BI

La forma como cambian las condiciones del mercado, demanda una respuesta cada vez más rápida y eficiente, por parte de las organizaciones. Para poder mantenerse a la cabeza, las compañías deben satisfacer, e incluso sobrepasar las expectativas de sus clientes. Las organizaciones son más dependientes de sus sistemas de inteligencia, para poder anticiparse a las tendencias más recientes, así como eventos que potencialmente sucederán en el futuro cercano.

Los usuarios de BI, están demandando inteligencia de negocios en tiempo real; se espera tener información disponible, en la misma forma como se monitorean en línea, las transacciones bursátiles; la información deberá estar siempre disponible y actualizada.

El término más reciente, que forma parte del proceso continuo de desarrollo de la industria de la Inteligencia de negocios, es BI 2.0. Se usa para describir la adquisición, provisión y análisis de datos, en tiempo real. Aunque esta tendencia tiene sus bemoles, porque en realidad no es nada sencillo llegar al nivel de actualización en tiempo real. Para llegar al verdadero tiempo real, nuevas tecnologías y paradigmas tendrán que venir.

3.1. Arquitectura BI

Los potentes sistemas operacionales, también conocidos como sistemas orientados a transacciones; son bastante comunes en las organizaciones de la mayoría de las industrias. Sin embargo, para mantenerse competitivas, las organizaciones hoy por hoy requieren sistemas orientados al análisis; que puedan incrementar la capacidad de la compañía para descubrir y utilizar, la información que ya posee.

Muchas compañías toman ventaja del uso de sólo, una pequeña fracción de la información que generan; el resto de la información que aún está sin descubrir, al combinarse con fuentes externas; puede convertirse en una verdadera mina de oro, esperando a ser explorada. Al explotar estas valiosas fuentes de información, estaremos generando conocimiento; tomando conciencia de cosas que no sabíamos, pero siempre estuvieron allí. Esto es lo que se llama descubrimiento de conocimiento.

Sin embargo, al realizar el esfuerzo de crear una organización BI, deberemos afrontar dos grandes barreras. Primeramente tenemos la organización en sí misma, ya que deberemos modificar su cultura corporativa; así como lidiar con ejecutivos y líderes muy renuentes al cambio. La segunda barrera, es superar la carencia de una tecnología integrada.

La arquitectura que soporte a la organización BI, debe ser construida con tecnología seleccionada de forma muy meticulosa, para garantizar una integración completa, o cuando menos, con tecnología que tenga adherencia a los estándares definidos.

Adicionalmente, la administración de la empresa debe asegurar que, la implementación de BI se apegue a la estrategia de la organización; impidiendo desarrollos que sólo satisfagan necesidades u objetivos individuales. Lo que no significa que el ambiente BI, no sea capaz de responder a las necesidades individuales de ejecutivos, y grupos o comunidades de usuarios; más bien esto se refiere a que el esfuerzo del desarrollo deberá estar enfocado al beneficio de toda la organización.

3.2. Componentes de la arquitectura BI

Actualmente, las personas encargadas de tomar decisiones dentro de una organización, requieren información precisa para contar con una “fotografía” actualizada de la situación presente de la organización. Sin embargo esta información, generalmente está dispersa en diversas plataformas, aplicaciones y bases de datos.

El BI se enfoca en las necesidades de información, y su principal objetivo es potenciar la capacidad de innovación de la organización.

Desde un punto de vista tradicional, la arquitectura BI incluye varios componentes, que se relacionan muy estrechamente con el DW.

En el siguiente diagrama (Figura 3.1.), se representan dichos componentes:

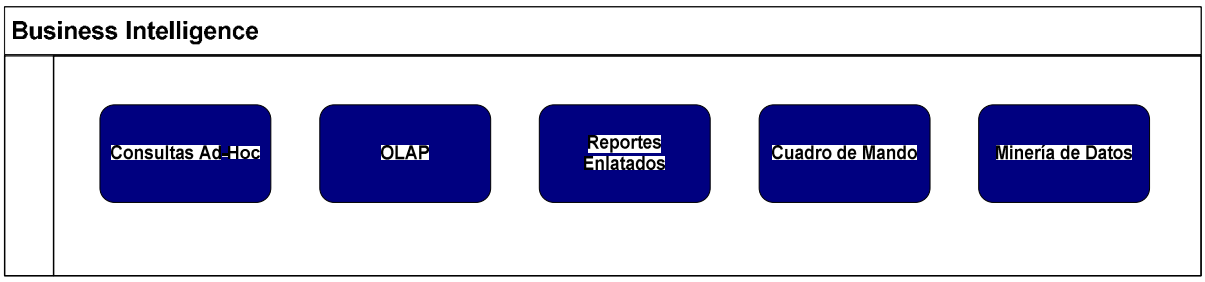


Figura 3.1. Componentes de BI

3.2.1. Consultas Ad-Hoc

Las herramientas para generar consultas ad-hoc, básicamente se encargan de generar y ejecutar las consultas, a través de una interfase amigable; adicionalmente cuentan con funcionalidades que permiten la presentación de resultados.

El proceso de generar una consulta, puede dividirse en dos etapas:

- Consulta
- Reporte

Consulta

Probablemente la parte más importante de una herramienta de este tipo, es la capacidad de extraer datos. Ni el formato, ni el análisis pueden llevarse a cabo sin que los datos hayan sido obtenidos.

Algo también muy importante, es el contar con una capa de metadatos entre el usuario y los datos provenientes de los sistemas operacionales, dicha capa nos permite presentar los datos de una forma entendible para el usuario, para ayudarlo a obtener los datos que le sean más relevantes.

Reporte

Permite al usuario dar formato adecuado a los resultados de la consulta, para producir reportes de buena calidad. El mercado de BI actualmente demanda que los resultados de una consulta tengan una presentación profesional.

3.2.2. OLAP

Procesamiento Analítico en Línea, OLAP por sus siglas en inglés (On-Line Analytical Processing), es un método para encontrar respuestas a consultas complejas. El OLAP toma los datos y los transforma en datos dimensionales, contra los cuales pueden ejecutarse las consultas.

Una estructura OLAP creada a partir de datos operacionales es conocida como Cubo OLAP. Dicho cubo es generado a partir de tablas organizadas bajo un esquema estrella, como hemos visto en el centro del modelo se encontrará la tabla de hechos, así como un conjunto de tablas de dimensión alrededor de ésta última. Para un mayor detalle respecto al OLAP,

3.2.3. Reportes Enlatados

Son reportes predefinidos, que forman parte de un paquete de software. Son prácticos ya que permiten a los usuarios ahorrar tiempo en la generación de reportes que deben generarse con una periodicidad. Por lo general se pueden tomar como plantillas, modificarlos y entonces contar con reportes más específicos.

3.2.3. Cuadro de Mando

Un cuadro de mando es una interfase de usuario, que de algún modo se asemeja al tablero de un auto, organiza y presenta información de una forma fácil de leer. Sin embargo, un cuadro de mando es mucho más interactivo que un tablero de auto. En BI, un cuadro de mando puede ser usado para tener acceder, organizar y analizar datos y reportes.

3.2.6. Minería de datos y agentes

El uso de la minería de datos, permite a una organización, crear perfiles de clientes, predecir en cierta medida las tendencias en las ventas, y entre otras iniciativas de BI, habilita el desarrollo del CRM. La minería deberá integrarse a las estructuras de datos del DW, también deberá estar soportado por los procesos del DW, para asegurar tanto efectividad como eficiencia en el uso de las distintas tecnologías y técnicas relacionadas. Dentro de la arquitectura BI, el ODS y los datamarts son excelentes fuentes de datos para la minería, además los resultados arrojados por la minería, deben ser aplicados en estructuras de datos que residan en el mismo DW, para asegurar la distribución de la información a todos los usuarios, dentro de la organización.

Las herramientas principales de la minería de datos, son los agentes. Los agentes son avanzados programas, basados en redes neuronales, que son entrenados para identificar puntos de contacto con los clientes, tendencias en la demanda de productos; basándose para ello en reglas predefinidas, de acuerdo a las circunstancias particulares de la organización. Existen también grupos de agentes, menos sofisticados, que simplemente reportan excepciones encontradas, a los altos ejecutivos de la organización.

3.3. CRM

La competencia cada vez más feroz ha forzado a muchas compañías a poner especial atención, en retener a sus clientes, y ganar nuevos. Es así como se lanzan grandes campañas de marketing, intentando obtener la fidelidad de sus clientes; modificando el enfoque, cambiando del marketing masivo, a un marketing más enfocado y personalizado.

Concentrarnos en la experiencia que el cliente ha tenido con nuestra compañía, se ha convertido en la clave para brindar un mejor servicio. Por ello más y más organizaciones, están adoptando los sistemas de administración de relación con el cliente, o CRM por sus siglas en inglés (Custumer Relationship Maganement).

El CRM promete incrementos significativos en las ganancias de las organizaciones, además de permitir una operación mucho más eficiente. El cambiar del enfoque tradicional de una organización (enfocado a procesos), a uno centralizado en el cliente, puede ayudarnos a incrementar la efectividad en la venta, por consiguiente darnos un margen mayor en las ganancias, mejorar la productividad a un costo menor, al tener disponible información y estadísticas relevantes sobre clientes. Brindar mayor satisfacción a los clientes de la organización, por consiguiente éstos estarán más comprometidos y dispuestos a continuar contratando los servicios que ofrece la organización.

Dada la situación actual, partiendo del hecho que realizar campañas para atraer nuevos clientes, generalmente conlleva un esfuerzo y un gasto bastante fuerte; no estamos en posibilidad de perder a ninguno de nuestros clientes. Por ello el objetivo final del CRM, es convertir a clientes no rentables, en rentables.

En muchas organizaciones, la visión respecto a los clientes varía dependiendo de los productos que se ofrecen, de la misión del negocio y de la localización geográfica. Cada área en la compañía puede estar haciendo uso de datos, relativos a los clientes, usando métodos muy diversos para obtenerlos, y con resultados igualmente diferentes. El cambio en este tipo de prácticas, requiere un compromiso por parte de todos los colaboradores de la organización, para entonces llegar a una fuente de información integrada.

Es aquí donde encontraremos retos importantes, ya que una iniciativa de esta índole implica nuevas formas de interactuar con los clientes, así como cambios radicales en los canales de venta. El CRM requiere nuevos flujos de información basados en la adquisición de datos en cada punto de contacto con los clientes. Este tipo de cambios generan una resistencia muy fuerte por parte de los empleados que laboran en la organización, ya que la estructura organizacional, así como los sistemas de incentivos, se ven alterados de forma dramática.

Desafortunadamente no es posible adquirir, o pretender tener una aplicación CRM, que solucione todos los problemas de la organización, sin afectar los procesos y cultura de la misma. Desde luego que el mejor lugar para empezar con el CRM, es generar un plan de acción y estrategia.

Como consejo, no debemos darnos a la tarea de primero adquirir la tecnología, de lo contrario nuestro intento podría fracasar; la tecnología debe soportar, no dirigir la implementación de una solución para CRM.

Toda organización que esté por iniciar un esfuerzo para brindar un mejor servicio a sus clientes, debe enfrentarse a preguntas como ¿qué debemos hacer si ya existe un DW en la organización, cómo lo reajustamos?

Si la organización planea iniciar el desarrollo de un nuevo data warehouse, ¿qué aspectos debemos tener en cuenta?

3.3.1. DW especialmente diseñado para CRM

El DW debe almacenar el detalle de cada transacción en cada punto de contacto, con cada uno de los clientes. Esto significa, que cada producto vendido a cada cliente, deberá estar registrado y almacenado en el repositorio del DW.

No sólo requerimos los detalles de ventas, sino el detalle de cualquiera de nuestras actividades, que se relacione de alguna forma con los clientes. Entre mayor sea el detalle, que dejemos disponible en el repositorio, el DW ofrecerá una mayor flexibilidad a los procesos de CRM. Sin embargo, almacenar este nivel de detalle, ocasionará que el DW contenga volúmenes de información verdaderamente grandes; por fortuna están disponibles nuevas tecnologías, que permiten el acceso a datos en dispositivos de almacenamiento masivo (Racks).

También deberemos asegurar que los datos sobre los clientes, estén completos, limpios y libres de duplicados. Además dichos datos, podrían enriquecerse al añadir datos demográficos e incluso referentes al perfil psicológico de cada uno de los clientes.

El CRM, se enfoca en dos tipos de requerimientos: operacionales y analíticos. La parte operacional del CRM lanza solicitudes para sincronizar los datos de los clientes, y los procesos centrales de la organización. Debido a esto, es muy frecuente que los sistemas operacionales de la empresa, sufran actualizaciones o modificaciones mayores, para satisfacer éstas necesidades de información.

La parte analítica del CRM, permite medir la efectividad de las decisiones hechas en el pasado de la organización, con el fin de optimizar las decisiones futuras. Todo esto se realiza en base a los datos almacenados en el DW, que como mencionamos deben ser precisos, integrados y accesibles.

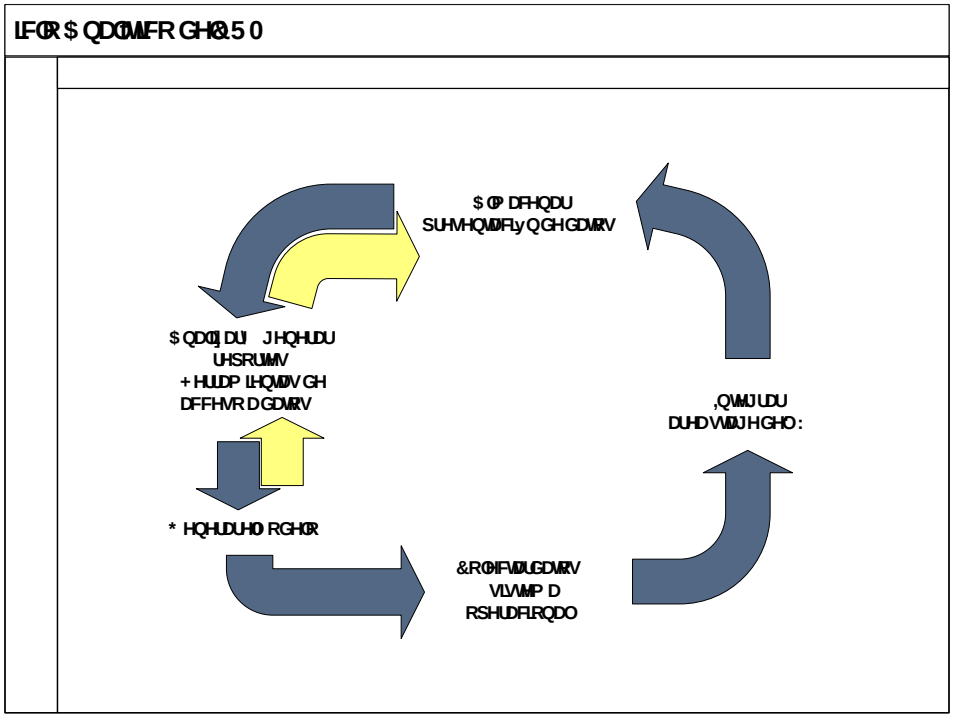


Figura 3.2. Ciclo Analítico del CRM

En la medida en que la organización se centra más en el cliente, así deberá hacerlo el DW, por ello resulta inevitablemente que el CRM derive en cambios para el DW. Adicionalmente generará un incremento en el volumen de los datos, ya que almacenemos más datos referentes a los clientes.

En la figura 3.2, mostramos el flujo de información del llamado ciclo analítico del CRM. A partir del modelo de CRM que adoptemos, tendremos que generar y analizar reportes, para ello deberemos obtener la información desde el repositorio del DW. Y para obtener información de ésta fuente, sabemos que el DW la extrae desde los sistemas operacionales, luego la transforma, limpia e integra, para finalmente, añadirla al repositorio.

3.3.2. Tendencias en Data Warehousing

Hasta ahora, hemos discutido una serie de pasos que podemos aplicar a la hora de planear el desarrollo de un DW, específicamente enfocado al CRM. Pero es hora de repasar algunas de las tendencias más relevantes respecto al DW.

3.3.2.1. Data Warehousing activo

Esto significa que son DW de uso extenso, podemos hablar del orden de los 30,000 usuarios en una corporación a nivel mundial; contando a empleados, clientes, y colaboradores de negocios. Si además lo hacemos disponible las 24 horas, los 7 días de la semana, podemos hablar de un ambiente con un nivel de servicio del 99.9%. Es así como el DW, se convierte en una herramienta crítica para la misión de la organización, en lugar de sólo ser herramienta estratégica.

3.3.2.2. Servicio uno en uno

Esto es lo que una empresa global, consigue con un DW activo. La compañía opera en más de 50 países, con fábricas en 30 países, además conduce investigaciones en 25 países, y vende más de 50,000 productos, en 80 países. Por todo esto, existen enormes ventajas, al abrir el DW a grupos de colaboradores externos. Los proveedores, trabajan de forma conjunta con la compañía bajo un plan de administración de la demanda. Los distribuidores de la organización colaboran bajo el plan de las diversas estrategias de ventas; los clientes toman importantes decisiones de compra. El DW activo, provee servicio uno en uno, a clientes y a colaboradores de negocio.

3.3.3. Estándares

El Data Warehouse es producto de la combinación de varias tecnologías; el rango es bastante amplio: modelado de datos, extracción de datos, transformación de datos, sistemas DBMS, módulos de control, sistemas de alerta basados en agentes, herramientas para consultas, herramientas de análisis, herramientas generadoras de reportes, etc.

Con tal multitud de tecnologías dando soporte al DW, diversidad de distribuidores y productos disponibles; nos enfrentamos al problema de cómo crear una solución efectiva, con lo mejor que ofrece el mercado. Sin embargo, la desventaja de utilizar productos de múltiples distribuidores, es la carencia de estándares para el intercambio de información.

Cuando por ejemplo, usamos el DBMS de un distribuidor, la herramienta para reportes de otro, y la herramienta de OLAP de un tercer distribuidor, estos productos no tendrán un estándar definido para funcionar conjuntamente, lo cual puede crear un verdadero caos.

Existen dos rubros, donde el manejo de estándares se vuelve crítico: intercambio de metadatos y funciones OLAP. En esta sección, haremos una revisión del progreso que ha tenido el desarrollo de estándares, en ambos rubros.

Con respecto a los metadatos, existen dos organismos internacionales, que se encargan de normar los estándares sobre los metadatos: Meta Data Coalition y Object Management Group.

Meta Data Coalition.- Formado en 1995 por un consorcio de vendedores, y de grupos interesados en la generación de estándares para generación e intercambio de metadatos. Han desarrollado un modelo conocido como OIM por sus siglas en inglés (Open Information Model). En diciembre de 1998, Microsoft se unió a la colación, convirtiéndose así en un gran soporte para el enriquecimiento del modelo. Finalmente en julio de 1999, la Meta Data Coalition aceptó el OIM como el estándar de facto.

Object Management Group.- Otro grupo de vendedores, en los que se incluyen Oracle, IBM, Hewlett-Packard, Sun Microsystems y Unisys. Igualmente pretenden desarrollar estándares, basándose en un foro bastante amplio. En junio del 2000, el Object Management Group develó el modelo CWM por sus siglas en inglés (Common Warehouse Metamodel), como un estándar para el intercambio en Data Warehousing. Tanto la Meta Data Coalition y el Object Management Group, han llegado a acuerdos, donde han establecido un mecanismo de cooperación para unificar criterios, y llegar a un único estándar.

OLAP Council.- Fue fundado en enero de 1995, como un grupo cuya misión es la protección de clientes, además de servir como guía a la industria. Este grupo está abierto a todas aquellas organizaciones que estén interesadas. Han trabajado en un estándar conocido como MDAPI por sus siglas en inglés (Multi-Dimensional Application Programmers Interface). Varios vendedores de aplicaciones OLAP, así como consultores e integradores de sistemas han hecho público su apoyo al MDAPI.

3.4. Funcionalidad OLAP

Como hemos visto en secciones anteriores, los usuarios de un Data Warehouse requieren llevar a cabo diversos análisis, en muchas ocasiones con cálculos bastante complejos; sin embargo las herramientas tradicionales como constructores de consultas y reportes, hojas de cálculo, etc., resultan insuficientes para realizar ésta importante tarea.

El término Online Analitycal Processing, OLAP por sus siglas en inglés; fue creado por el Dr. E. F. Codd, quien es reconocido como el padre del modelo relacional de bases de datos. El Dr. Codd publicó un documento en el año de 1993, cuyo título era: “Providing On-Line Analytical Processing to User Analysts” las 12 reglas para un sistema OLAP. Posteriormente, en el año de 1995 se añadieron 6 reglas adicionales.

OLAP es una nueva categoría de software que permite a los analistas, gerentes y ejecutivos, obtener acceso a datos, de una forma consistente e interactiva, a través de una amplia variedad de vistas, que reflejan una verdadera multidimensionalidad.

Los lineamientos propuestos por el Dr. Codd, deben ser tomados en cuenta, a la hora de elegir una solución OLAP para nuestra organización. A continuación listamos las doce primeras reglas publicadas en 1993:

Vista conceptual multidimensional. Proveer un modelo de datos multidimensional, que es intuitivamente analítico, y además fácil de usar. La forma como los analistas de negocio perciben la organización, tiene una naturaleza multidimensional.

Transparencia. Debemos esforzarnos por hacer transparente para los usuarios, la tecnología detrás de un repositorio de datos, así como las diversas fuentes de datos. Esto nos ayuda a mantener la eficiencia y productividad de los usuarios, a través de interfaces de usuario final que les sean amigables y en la medida de lo posibles familiares.

Accesibilidad. Dar acceso sólo a los datos que se requieren para un análisis específico, presentar una única vista para los usuarios, y que además sea coherente y consistente. Un sistema OLAP, deberá tener un proceso a través del cual sea posible mapear un esquema lógico, con un repositorio de datos heterogéneo.

Desempeño consistente. Asegurar que los usuarios, no experimenten una degradación importante del desempeño en el servicio de generación de reportes, mientras el número de dimensiones, o el tamaño de éstas se incremente.

Arquitectura cliente/servidor. El sistema deberá estar conformado, según los principios del cliente/servidor, para asegurar un desempeño óptimo. El sistema debe ser flexible y adaptable, para permitir añadir usuarios nuevos, con el mínimo esfuerzo.

Dimensionalidad genérica. Contar con una estructura lógica para todas la dimensiones. Asegurar que cada dimensión de datos, es equivalente en estructura y capacidades operacionales. Es decir, la estructura básica de datos, así como las técnicas de acceso, no deben basarse en una única dimensión de datos.

Matriz dinámica de distribución. El sistema deberá ser capaz de deducir de forma dinámica, la distribución de los datos, y ajustar el acceso y almacenamiento a éstos; para mantener un nivel aceptable de desempeño.

Soporte multiusuario. Proveer acceso a los usuarios finales, sin importar si están trabajando en un mismo modelo analítico, o creando nuevos modelos para los mismos datos. Es decir, brindar un acceso concurrente a datos, integridad de datos, así como un acceso seguro.

Operaciones cruzadas con dimensiones. Asegurar que el sistema, cuente con la capacidad de reconocer la diversas jerarquías dentro de las dimensiones, y permitir la navegación hacia arriba y hacia abajo, en cada uno de los cortes o cruces de dimensiones.

Manipulación intuitiva de datos. Permitir una ruta de consolidación de datos, es decir que las operaciones de combinar datos, navegación hacia arriba y hacia abajo, puedan realizarse de forma intuitiva.

Generación flexible de reportes. Consiste en brindarle al analista de negocio, las herramientas que le permitan organizar columnas y renglones en tal forma, que facilite la manipulación, análisis y resumen de la información.

Múltiples niveles de agregación. Permitir un número prácticamente ilimitado, de niveles de agregación entre dimensiones, con cualquier ruta de consolidación.

3.4.1. Principales Características OLAP

Al inicio de la sección discutimos, el origen y reglas básicas de los sistemas OLAP. Ahora nos adentraremos en las características de dichos sistemas, tratando de mantener un lenguaje sencillo.

Las características fundamentales de un sistema OLAP, son:

- Permitir a los analistas de negocio tener una vista multidimensional, de los datos almacenados en el DW.
- Facilitar la generación dinámica de consultas, así como la capacidad de generar análisis complejos.
- Contar con un mecanismo de navegación que permita obtener detalles hacia abajo, e información resumida hacia arriba.
- Generar comparativos.
- Presentar los resultados en una amplia gama de formas, incluyendo gráficos.

Adicionalmente, podemos distinguir entre características básicas y características avanzadas, en el siguiente diagrama (Figura 3.3.), se enumeran todas:

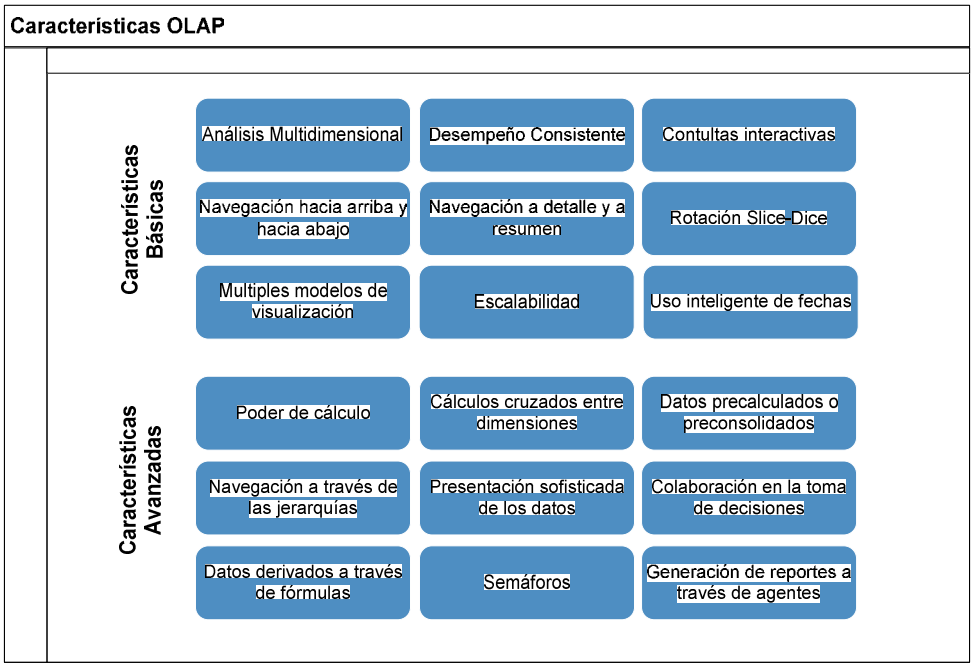
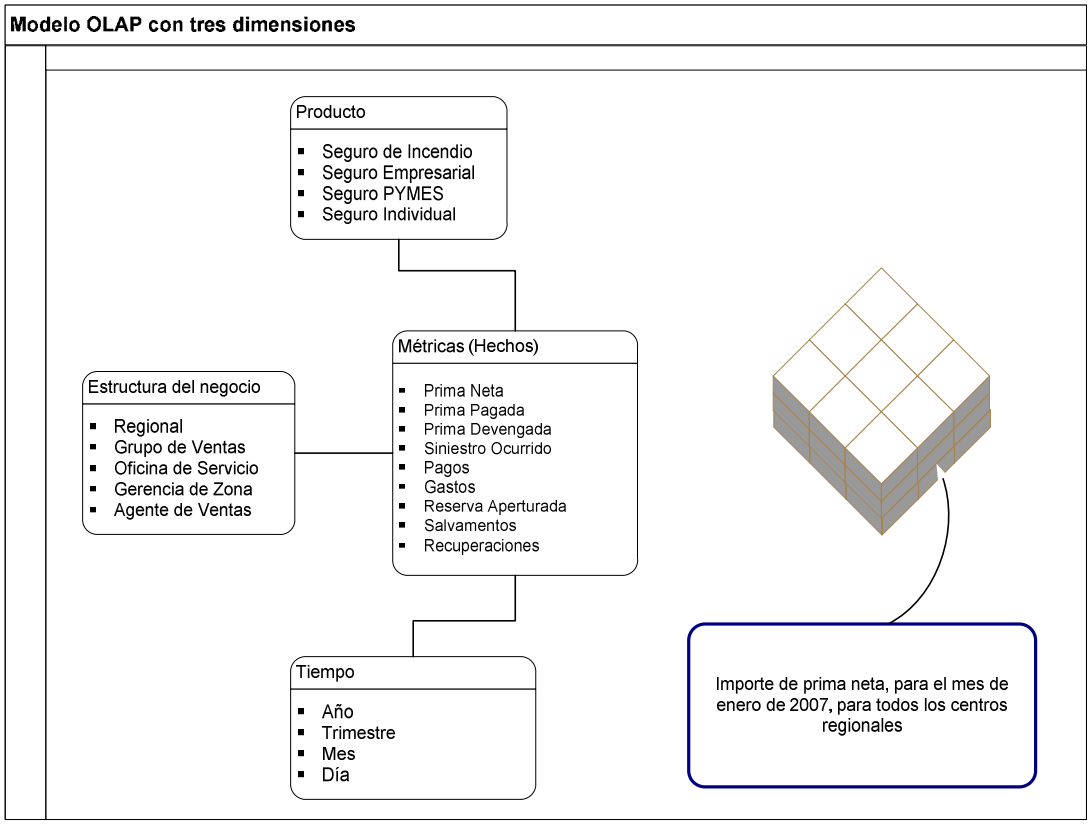


Figura 3.3. Características OLAP

3.4.2. Análisis Dimensional

Ya hemos comentado en las secciones anteriores, que el análisis dimensional es un de los puntos fuertes de los sistemas OLAP. Para obtener una visión más clara de ésta importante funcionalidad del OLAP, veamos un caso a manera de ejemplo.

En el diagrama siguiente (Figura 3.4.) se muestra un modelo estrella, y su representación tridimensional en forma de cubo; donde en el eje X tenemos el producto, luego en el Y tenemos la estructura y finalmente en el eje Z, tendremos el tiempo.



3.5.1. ROLAP

El término ROLAP es la abreviatura de Relational Online Analytical Processing, se basa en el modelo relacional del DBMS subyacente al Data Warehouse; las funciones OLAP se proveen a través la base de datos relacional.

En este modelo el motor de OLAP reside en el cliente, es decir la computadora que se conecta al servidor, para solicitar la información. En la siguiente figura (Figura 3.5.), mostramos la arquitectura básica de un modelo ROLAP:

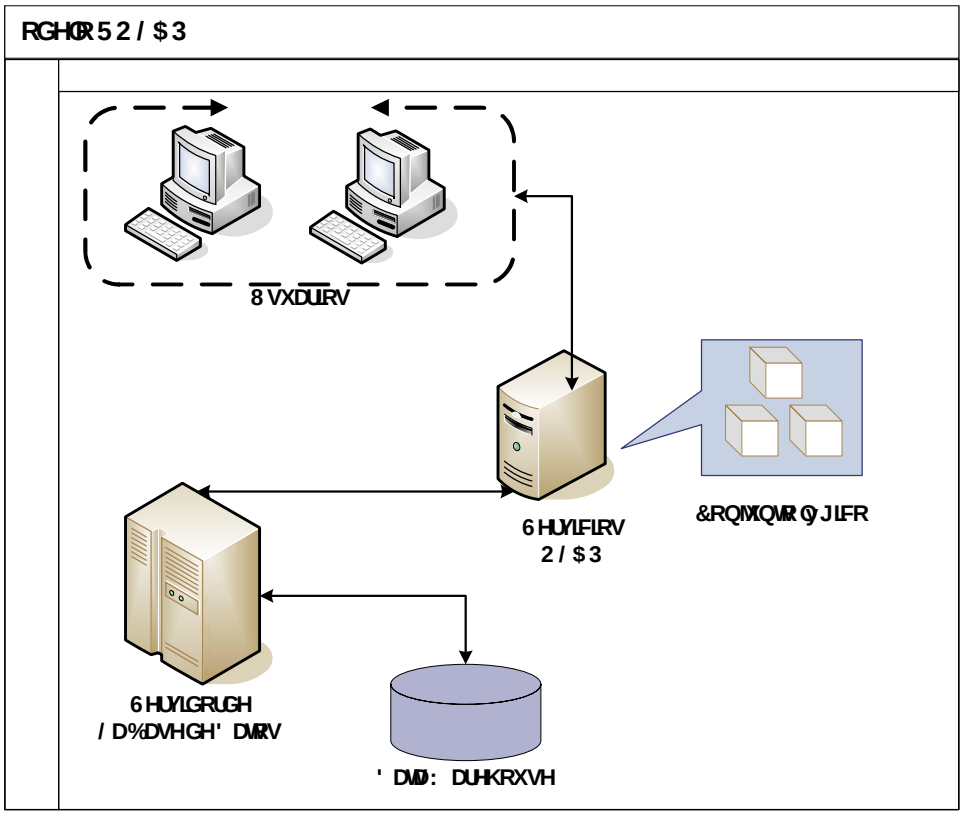


Figura 3.5. Modelo ROLAP

3.5.2. MOLAP

El término MOLAP es la abreviatura de Multidimensional Online Analytical Processing, se implementa a través de datos almacenados de forma multidimensional, es decir, la estructura de datos está diseñada de tal forma que sea posible definir métodos de almacenamiento utilizando un sistema de coordenadas. Generalmente los proveedores de bases de datos multidimensionales (MDDBs) utilizan sistemas propietarios.

En la siguiente figura (Figura 3.6.), mostramos la arquitectura básica de un modelo ROLAP:

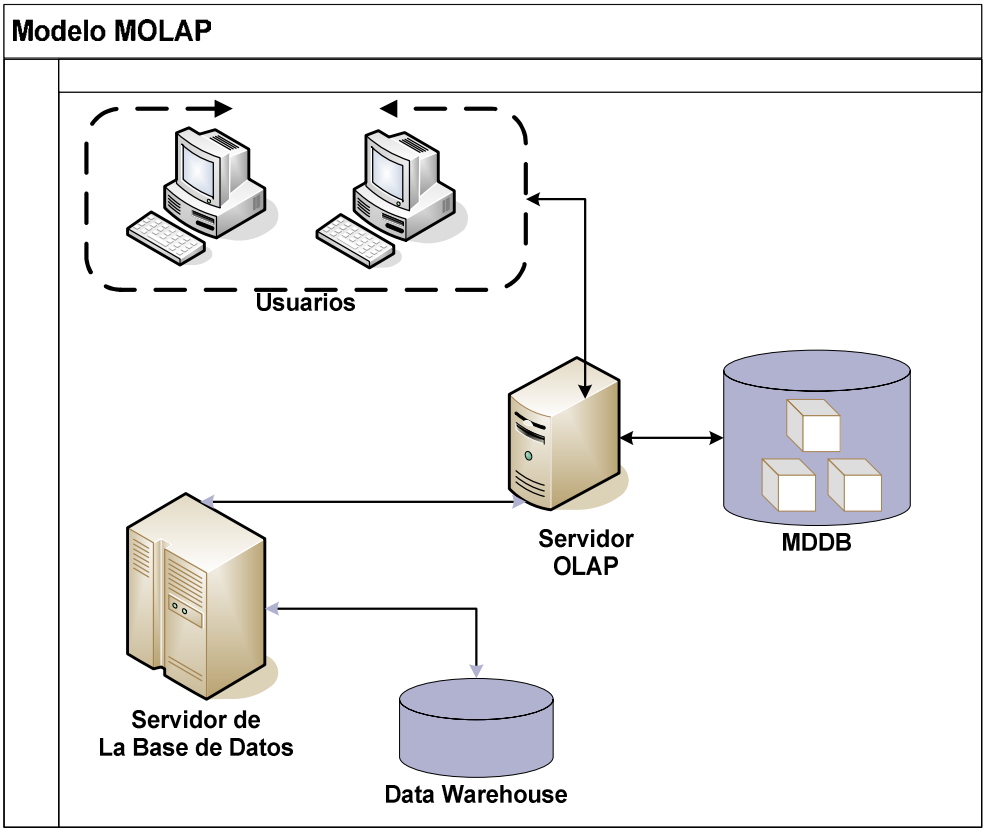


Figura 3.6. Modelo MOLAP

3.6. Cuadro de Mando Integral

El conocer nuestra organización resulta de vital importancia para la toma de decisiones, y la mejor manera de conocerla, es utilizando los KPI (Key Performance Indicators, por sus siglas en inglés).

Los KPI's son métricas que nos ayudan a cuantificar objetivos, es decir, reflejan el desempeño de la organización. Dichas métricas varían dependiendo del tipo de organización, así como del plan estratégico que ésta tenga.

Cuando una organización está en proceso de definición de sus propios KPI's, debemos asegurar que cada uno de ellos sean:

- Específicos
- Medibles
- Alcanzables
- Realistas
- Oportunos

Para que una organización pueda identificar sus KPI's, es necesario que cubra los siguientes requisitos:

- Tener procesos de negocio predefinidos
- Contar con metas claras
- Criterios de desempeño para los procesos de negocio
- Mediciones cuantitativas y cualitativas de los resultados
- Investigar las variaciones que se obtengan en el proceso de muestreo

Los indicadores pueden dividirse en:

- Cuantitativos: Aquellos que pueden representarse como número
- Prácticos: Interactúan con los procesos de negocio
- Direccionales: Nos indican cuando la organización tiene mejoras o no
- Accionables: Indicadores de control para la organización, que pueden disparar un cambio

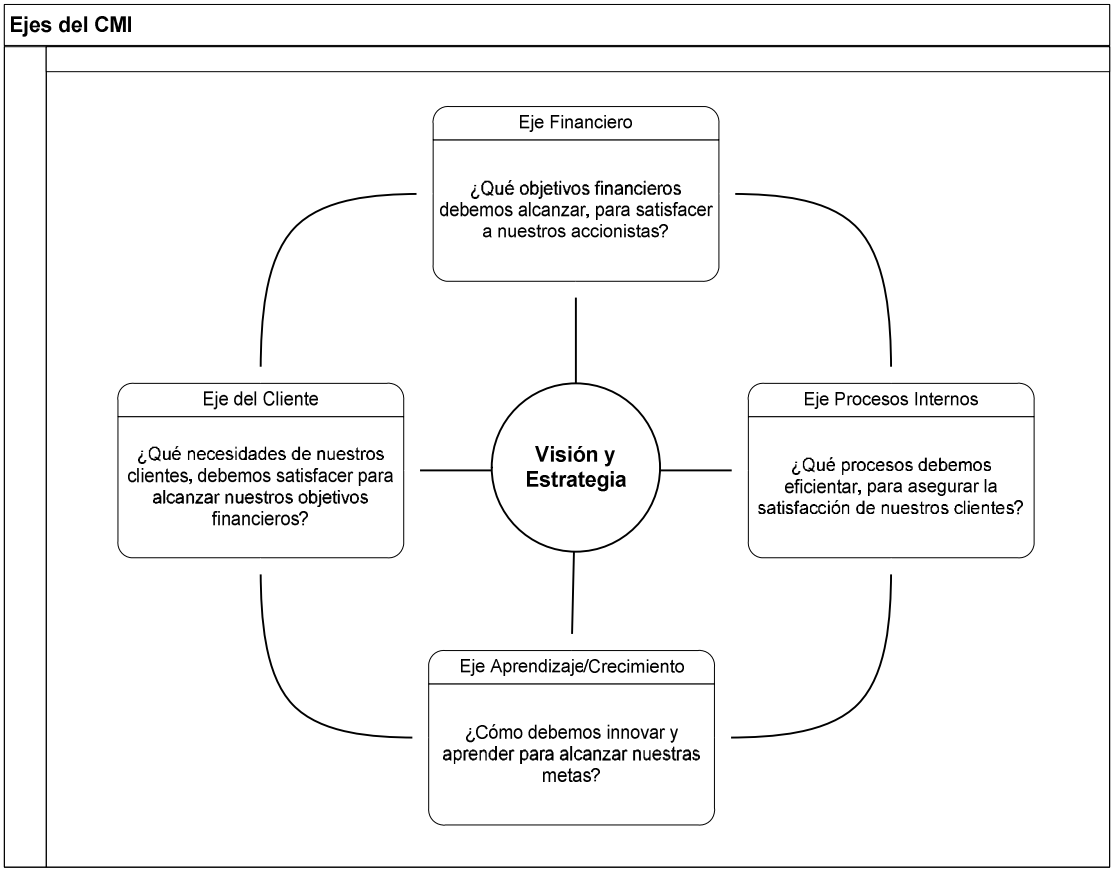
Ahora bien, a este punto ya sabemos qué son los KPI, pero ¿cómo obtenerlos y cómo darles seguimiento? Bueno la respuesta es el cuadro de mando integral o CMI.

3.6.1. Breve historia del CMI

Los orígenes del Cuadro de Mando Integral (CMI), se remontan al año de 1990. Cuando una firma de consultoría llamada Nolan Norton Institute, participó en un estudio aplicado a varias empresas.

Dicho estudio, tenía como propósito medir los resultados de las empresas, utilizando un nuevo paradigma. En dicho estudio David Norton participó como líder del proyecto, y Robert Kaplan de la universidad de Harvard, participó como asesor académico.

El resultado de dicho proyecto, fue el CMI estructurado alrededor de cuatro ejes: financiero, cliente, aprendizaje/crecimiento y procesos internos.


Ejes del CMI
Eje Financiero
¿Qué objetivos financieros debemos alcanzar, para satisfacer a nuestros accionistas?
Eje del Cliente
¿Qué necesidades de nuestros clientes, debemos satisfacer para alcanzar nuestros objetivos financieros?
Visión y Estrategia
Eje Procesos Internos
¿Qué procesos debemos eficientar, para asegurar la satisfacción de nuestros clientes?
Eje Aprendizaje/Crecimiento
¿Cómo debemos innovar y aprender para alcanzar nuestras metas?

Eje de procesos

Primero debemos generar la cadena de valor de la empresa, luego basándonos en la situación del mercado donde participa la organización, identificar los procesos clave. Luego debemos asegurar que los procesos se ajusten a las necesidades de nuestros clientes, definiendo las características que deben cumplir nuestros productos y/o servicios. Finalmente a las actividades identificadas se les asignan los indicadores correspondientes.

Eje de aprendizaje/crecimiento

Este eje que generalmente aparece en cuarto lugar, es el motor de los ejes anteriores. Refleja los conocimientos y habilidades que posee la organización para desarrollar productos, así como para aprender y modificar su rumbo. El uso de la tecnología, y el valor que genera las personas que colaboran en nuestra organización, la disponibilidad de información estratégica para asegurar una toma de decisiones lo más óptima posible, así como una cultura organizacional adecuada, servirán para afianzar las acciones que tomemos en pos del crecimiento.

La utilización del CMI dentro de una organización, ofrece un amplio horizonte de posibilidades; aunque su implementación no se del todo sencilla. Las ventajas derivadas del trabajo con esta importante herramienta, la han colocado entre las más importantes en la actualidad, tanto para gerentes como consultores de negocio.

Capitulo 4

4. DW aplicado en una empresa de seguros

En esta última sección, veremos cómo aplicar los conceptos revisados a lo largo de las secciones anteriores. Cómo diseñar un Data Warehouse para la empresa de seguros.

4.1. Escenario

Supongamos que hemos sido contratados por una empresa, que ofrece seguros de daños sobre propiedad, autos y gastos médicos mayores. Hemos sostenido entrevistas con los gerentes de las áreas de contratación, siniestros, finanzas y ventas. Todos no han externado la necesidad de contar con información confiable y oportuna, para la toma de decisiones. Ya que las exigencias del mercado, y la aparición de nuevos competidores que ofrecen pólizas de seguros a menor precio, debido a que utilizan canales anteriormente inexplorados y mucho más económicos que los tradicionales (como Internet), han forzado a disminuir su participación en el mercado.

Afortunadamente, la compañía aseguradora objeto de nuestro estudio, cuenta actualmente con sistemas operacionales que capturan y procesan el grueso de los datos requeridos. Sin embargo, el problema es que los datos no están integrados; ya que a lo largo del tiempo han surgido fuentes de información muy diversas y dispares, que contienen información acerca de los clientes, productos y canales de distribución.

En la situación actual dentro de los sistemas operacionales legacy, pueden existir registros donde un mismo cliente, puede aparecer referenciado a distintos autos, propiedades, o incluso a distintas coberturas de gastos médicos. Hasta ahora esto había funcionado de forma relativamente aceptable, debido a que cada línea de negocio operaba de forma autónoma; existía poco o nulo interés en compartir datos a través de las diversas líneas de negocio para llevar a cabo una venta cruzada o tener una mayor colaboración. Adicionalmente, los usuarios no tienen la facilidad de acceder a los datos de forma sencilla y cuando lo necesitan.

Por esta razón la organización ha realizado varios esfuerzos, para intentar resolver sus requerimientos de información en el corto plazo y mediano plazo. Como resultado, en muchos de los casos se extrajeron los mismos datos, desde las mismas fuentes de información, para resolver las necesidades de las distintas líneas de negocio sin una visión global de la organización. El resultado no se hizo esperar, cuando los resultados presentados en las reuniones de gerencia, reflejaban números diferentes sobre los indicadores de desempeño de la organización, dependiendo de la fuente de análisis.

Por ello, nuestro trabajo consistirá en transformar un conjunto enorme de datos inconsistentes y desagregados, en un repositorio de datos desde el cuál puedan llevarse a cabo análisis confiables y precisos.

4.2. Macro Procesos en Seguros

La cadena de valor de una empresa de seguros es aparentemente corta y sencilla. Los macro procesos están enfocados a la emisión y administración de pólizas, cobranza de primas y otros pagos, así como procesar y administrar los siniestros.

La compañía aseguradora, está interesada en obtener un mejor entendimiento sobre las métricas generadas por cada uno de éstos procesos. Los usuarios desean analizar información detallada de las transacciones derivadas de la emisión de pólizas, así como de las transacciones derivadas de los siniestros ocurridos. Otra funcionalidad deseable es tener la capacidad de medir la rentabilidad respecto al tiempo, por cobertura, objeto (auto, casa habitación, personas), estado de la república, canal de distribución. Dar seguimiento a la rentabilidad implica, monitorear tanto las ganancias, como los costos incurridos.

Desde luego que una compañía de seguros cuenta con una cadena de procesos amplia, como la inversión de las primas que son cobradas, una amplia lista de proveedores externos como, agentes, ajustadores, talleres, hospitales, clínicas, médicos, etc. No es objetivo del presente trabajo de tesis, analizar cada uno de éstos procesos, para incluirlos en el diseño del DW. Sólo nos centraremos en los procesos clave de la aseguradora.

4.3. Emisión de Pólizas

La cadena de valor de una compañía aseguradora inicia con la administración de las pólizas que emite; el proceso es a grandes rasgos como sigue:

Un agente de seguros vende la póliza a un asegurado, antes de que la póliza sea creada debe establecerse un precio, dicho precio o prima se establece en función de las coberturas que el asegurado desee contratar. Una vez hecho esto, un suscriptor realiza el análisis del riesgo y finalmente toma la responsabilidad de hacer negocio con el asegurado, y se hace la aprobación de la póliza.

Un sistema operacional se encarga de capturar los siguientes tipos de transacciones:

- Creación de póliza
- Modificación de póliza
- Cancelación de póliza (con un motivo)
- Generación de coberturas
- Modificación de coberturas
- Cancelación de coberturas (con un motivo)

La granularidad de una tabla de hechos para transacciones de pólizas, será el código o número de póliza individual. Cada transacción debe estar acompañada de datos de detalle que permitan establecer una descripción en forma dimensional. Las dimensiones asociadas a las transacciones de una póliza son: fecha de transacción, fecha de efecto de la póliza, asegurado, agente, coberturas, objeto asegurado, número de póliza y tipo de transacción de la póliza.

4.4. Tratamiento de Pólizas

Basándonos en la información obtenida a través de entrevistas con los usuarios, optamos por englobar las transacciones que afectan a las pólizas como un único proceso de negocio. Adicionalmente existe la necesidad de analizar las ganancias, obtenidas a partir de las primas asociadas a cada póliza, utilizando una base mensual. Las compañías aseguradoras obtienen esta ganancia a través del tiempo; es decir, la aseguradora sólo puede disponer de la prima, una vez que ha pasado el periodo de tiempo en el que se ha comprometido a darle cobertura al objeto o bien asegurado.

El analizar ésta ganancia con una base mensual, trae una pequeña complicación debido a que deberemos mantener un nivel detallado de las transacciones, y un nivel resumido o acumulado mensual, cabe mencionar que éste no es una simple sumarización de los datos de detalle, es un proceso que vendrá de una fuente separada.

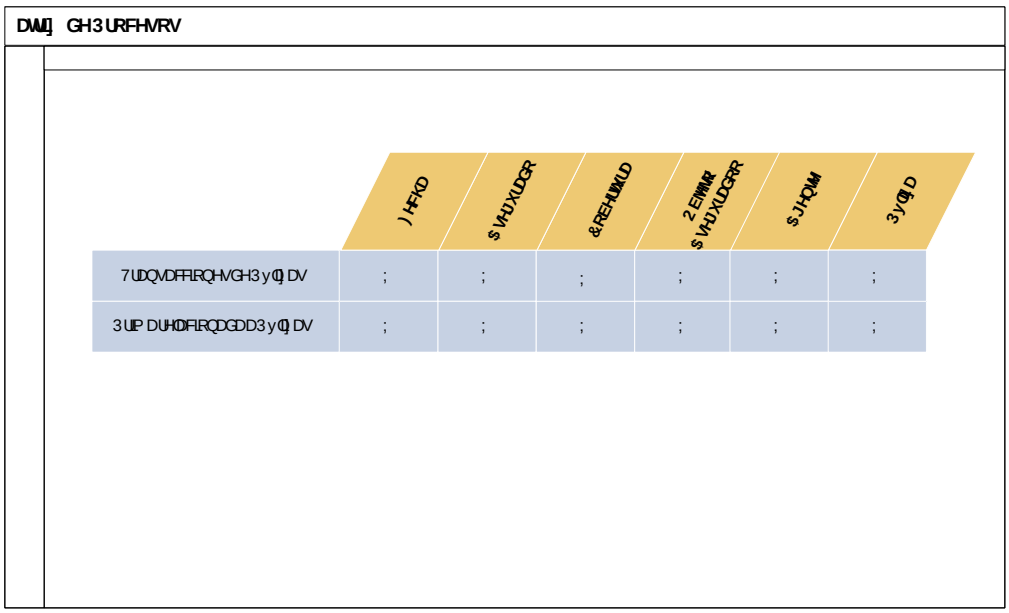


Figura 4.1. Matriz de procesos

En el diagrama anterior (Figura 4.1.), se muestran las dimensiones principales como fecha, asegurado, cobertura, objeto asegurado, agente y póliza. Esta matriz no planea incluir todas las posibles dimensiones, de otra forma fácilmente llegaríamos a tener más de cien columnas.

Ahora analicemos la matriz comenzando por el primer renglón, que se centra en las transacciones necesarias para emitir y modificar una póliza. La póliza será una cabecera que abarca un grupo de coberturas que fueron vendidas al asegurado. Las coberturas para seguros de daños incluyen incendio, inundación y robo/asalto en casa habitación, terremoto y erupción volcánica. Dichas coberturas sólo aplican al o los objetos asegurados, que se encuentran cuidadosamente especificados en la póliza.

4.4.1. Dimensiones y técnicas

En esta sección emplearemos conceptos que fueron revisados en secciones previas, también tendremos la oportunidad de adentrarnos en el detalle de las técnicas empleadas para diseñar y generar tablas de dimensiones, que se adapten mejor a nuestras necesidades.

Existen dos fechas asociadas a las transacciones de pólizas, una es la fecha en que fue realizada la transacción, es decir, cuando la operación se capturó en el sistema operacional. Otra fecha es aquella donde dicho movimiento o transacción toma efecto. Deberemos construir dos dimensiones independientes a partir de dichas fechas, ambas pueden ser implementadas utilizando una misma tabla física; para ello se utilizan vistas con nombres de columna únicos. Una vista es una tabla lógica generada a partir de una instrucción de SQL.

El generar una tabla de dimensión de clientes adecuada, se convierte en elemento crítico; por lo general la dimensión de clientes, resulta ser la más compleja en todo DW; en una organización de buen tamaño, la profundidad de ésta tabla puede llegar a ser de varios millones de registros, así como su extensión puede abarcar varios cientos de atributos e incluso algunas veces puede ser sujeta a actualizaciones en periodos de tiempo muy cortos.

Por ejemplo, una organización que se dedique al marketing puede con facilidad tener más de 3,000 atributos en una tabla de dimensión de clientes. Organizaciones de mayor tamaño, como bancos y secretarías de gobierno, pueden llegar a tener tablas de dimensiones con más de cien millones de registros; una tabla de este tipo frecuentemente es una amalgama de datos que provienen tanto de fuentes internas como externas.

4.4.1.1.1. Análisis del nombre y dirección

Independientemente de que tratemos con personas físicas o morales, en algún lugar debe almacenarse el nombre y dirección de cada uno de los clientes. Por lo general, esto se hace de la forma más simple posible, para facilitar los procesos en el DW. Muchos diseñadores, suponen que el uso de columnas con nombres genéricos, como nombre1, nombre2, nombre3, dirección1, dirección2 y dirección3, puede funcionar bajo cualquier circunstancia. Por desgracia, estos manejos, se vuelven completamente inútiles cuando deseamos segmentar la base de clientes, e incluso tampoco son recomendables si deseamos mantener un entendimiento claro sobre la base de clientes.

Derivado del uso de columnas con nombres genéricos, podemos generar problemas en la calidad de los datos. En la siguiente tabla mostramos algunos ejemplos, del uso de columnas genéricas, en un diseño. Podemos observar que el nombre de la columna es limitado en su descripción, no existe un mecanismo que nos permita manejar títulos, sufijos, etc. No podemos identificar cuál es el primer nombre de la persona, ni cómo deberíamos saludarle en una carta modelo personalizada.

Si observamos la tabla siguiente (Tabla 4.1.) los datos de dirección, teléfono y código postal, tienen abreviaturas que son usadas de forma inconsistente en distintos lugares. Estos campos, podrán tener suficiente espacio para albergar una dirección completa, pero no hay una norma que se ajuste a las reglas postales, que nos asegure que esa dirección estará identificada debidamente.

Atributo de la dimensión	Ejemplo de contenido
Nombre	González Ortiz María Cristina
Dirección-1	Av. Coyoacán #13 int. 24
Dirección-2	Coyoacán, Distrito Federal
Ciudad	México
Estado	Distrito Federal
Código Postal	C.P. 04480
Teléfono	Tel.:56089730

Tabla 4.1. Atributos genéricos de una dimensión

En lugar de utilizar pocos campos de propósito general, deberemos descomponer los atributos de nombre y dirección, en tantas partes como componentes elementales tengan. Luego el proceso de extracción, deberá realizar un análisis sintáctico (parking, en inglés) de los nombres y direcciones, para limpiar los datos lo más posible. De este modo los atributos pueden ser estandarizados, por ejemplo, después del análisis sintáctico, “Av” se convertiría en “Avenida”, e “int” se convertiría en “interior”.

Del mismo modo, se debe validar que el código postal sea correcto, y que coincida con la colonia que está descrita en la dirección.

En la siguiente tabla (Tabla 4.2.), se muestra un ejemplo de cómo pueden descomponerse los atributos de nombre y dirección, para flexibilizar el uso de la base de clientes; y hacerla útil para los propósitos del CRM:

Atributo de la dimensión	Ejemplo de contenido
Salutación	Srita.
Nombre informal	Cristina
Nombre formal	Srita. María Cristina
Nombres	María Cristina
Apellido Paterno	González
Apellido Materno	Ortiz
Nacionalidad	Mexicana
Titulo	Lic.
Tipo de vía	Avenida
Nombre de la vía	Coyoacán
Número	13
Número interior	24
Colonia	Del Valle
Delegación	Coyoacán

Ciudad	México
Estado	Distrito Federal
Código Postal	04480
Teléfono	56089730
Fax	56089730
Correo electrónico	cgonzo @servidor.com
Sitio Web	
Número único de cliente	7346531

Tabla 4.2. Atributos descompuestos de una dimensión

Las personas morales, generalmente tienen varios domicilios, por ejemplo pueden tener una dirección para las oficinas, otra para la bodega de mercancías, etc. Se puede seguir el mismo esquema que mostramos en la tabla anterior, para integrar esta información.

Vale la pena mencionar, que debemos desarrollar un estándar para las direcciones, teléfonos y correos electrónicos, y asegurar que contemos con información lista para usarse, sin necesidad de hacer manejos especiales. Ya que generalmente, estos manejos se realizan desde el sistema operacional, por medio de programas. Como solución alterna algunas organizaciones, tienen procesos de solicitud muy específicos para los capturistas, de forma que se cumpla con estas reglas y los datos viajen de forma estandarizada a las tablas correspondientes.

4.4.1.1.2. Otros atributos comunes a la dimensión de clientes

Desde luego que la lista de atributos relacionados con el cliente, es amplia; sin embargo entre más información descriptiva capturemos, la dimensión de clientes será más robusta, por consiguiente mucho más interesante será el análisis y más valiosos los resultados que éste arroje.

Fechas

Adicionalmente, es necesario agregar diversas fechas, relacionadas con nuestros clientes, tal como fecha miembro desde, fecha de nacimiento, etc. Aunque estas fechas, podrían estar inicialmente con un formato SQL, es muy recomendable que sean convertidas a un formato que nos permita utilizarlas de acuerdo al calendario, de tal forma que sea posible sumarizar, agrupar y segmentar los datos, por fechas que sean relevantes.

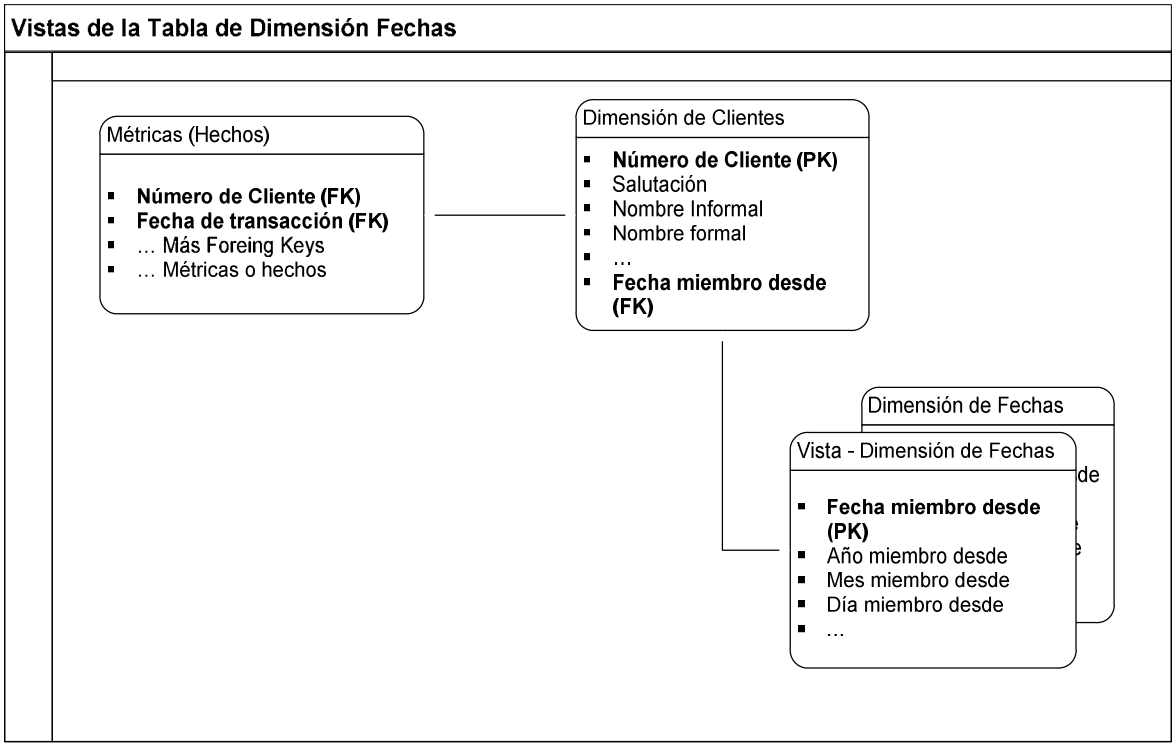
Para realizar esta tarea, es una práctica común, el utilizar una dimensión de fechas, en donde cada una de las fechas que se relacionen con el cliente, serán llaves foráneas en la tabla de dimensión fechas. Con ello logramos un efecto tal, que el sistema se comporta, como si por cada llave foránea, tuviéramos una tabla física independiente; cada una de estas tablas constituye una vista de la tabla dimensión de fechas.

Atributos de segmentación

Algunos de los atributos más útiles, en la dimensión de clientes, son los clasificadores de segmentación, también conocidos como marcadores. Éstos varían bastante, dependiendo del negocio; es decir, no se utilizarán los mismos clasificadores para banca, que para compañías aseguradoras. Sin embargo si existen un grupo de marcadores que son comunes a todo negocio, y están relacionados con los clientes que son personas físicas:

- Género
- Nacionalidad
- Edad, o un clasificador de etapa de vida
- Ingresos, o un clasificador de nivel socio económico
- Estado, por ejemplo nuevo, activo, inactivo, etc.
- Segmento del mercado, o un identificador de las preferencias de compra
- Identificadores que nos digan sus preferencias de pago, probabilidades de morosidad, etc.

Adicionalmente podemos utilizar una fecha como clasificador, en el siguiente diagrama (Figura 4.2.), podemos ver la forma en como se relacionan la tabla de hechos, la dimensión cliente y una vista lógica generada a partir de la dimensión de fechas, donde se descompone la fecha miembro desde, en sus mínimos componentes:



Métricas sumarizadas como atributos

Algunos usuarios están interesados en clasificar clientes, de acuerdo a métricas resumidas, que nos indiquen un rating. Por ejemplo, aplicar un filtro sobre la tabla, que nos de cómo resultado a todos aquellos clientes que gastaron más de cierto monto en el año pasado, cuáles asegurados tuvieron más siniestros, un acumulado del importe total que ha pagado por concepto de primas.

Es casi seguro, que este tipo de clasificadores, se vendarán muy bien con los usuarios finales; ya que no habrá necesidad de que se generen consultas adicionales, para poder conocer dicha información. Cabe aclarar que estos atributos, son utilizados únicamente, como clasificadores, no como resultados que se usen para cálculos.

Si bien el uso de estos atributos, aumentan la flexibilidad de las consultas, además de brindar un plus al usuario final, también tienen su lado oscuro. Ya que agregan una carga extra, a los procesos de ETL, que generan esta información. Debemos incluir también, el esfuerzo de verificar que la información sea consistente, y actualizada. Por esta razón, es recomendable centrarnos en atributos que no requieran una actualización tan constante.

Otra práctica recomendable, es el uso de etiquetas, que nos den un valor más descriptivo; en lugar de almacenar importes como tal; por ejemplo, el usar la etiqueta “Alta Siniestralidad”, en lugar de sólo almacenar un número 1,000. Ya que este tipo de prácticas, minimizan nuestra vulnerabilidad; al no estar bajo la lupa en caso de que el importe almacenado como atributo, no coincidiera con el importe almacenado en la tabla de hechos.

4.4.1.1.3. Tablas dimensión de clientes con rápida variación

Las tablas de dimensión de clientes, cuando contienen varios millones de registros, representan dos grandes retos: el primero es el tiempo de respuesta, al realizar consultas; y el segundo es la imposibilidad de aplicar técnicas de rastreo para identificar cambios. Esto es, que se hace una revisión registro por registro, para verificar qué registros han sufrido cambios.

Desafortunadamente, es muy probable que este tipo de dimensiones, cambien más rápido que tablas de dimensión con tamaño más moderado. El rastrear los cambios en los atributos de una tabla de dimensión, es algo más que una funcionalidad deseable. Por ejemplo, toda compañía aseguradora debe tener información actualizada respecto a los automóviles, y demás bienes que están asegurados, porque resulta crítico tener una “fotografía” que refleje de forma precisa la realidad, cuando sucede un siniestro.

La solución para esta problemática, consiste en separar aquellos atributos, que son consultados constantemente; al igual que aquellos que cambian en periodos muy cortos de tiempo. Al separarlos, generaremos una especie de mini dimensión. Por ejemplo, podríamos crear una mini dimensión para atributos como el género, edad, número de hijos, nivel de ingreso; ya que estos típicamente son requeridos de manera muy continua. En estos casos se recomienda generar un registro, por cada combinación única de género, edad, número de hijos, nivel de ingreso; y no por cliente. Ya que estas son las columnas que estarán sujetas a análisis; además que los usuarios requieren rastrear los cambios realizados a éstos atributos. En contraste, los atributos que no son tan frecuentemente requeridos, y que además no son de tan rápido cambio, los dejaremos en la enorme tabla de dimensión original.

Es importante que al crear mini dimensión, con atributos continuamente variables, hagamos una conversión de éstos valores, a rangos asociados; es decir, forzar a que los atributos tomen un número relativamente pequeño de valores discretos. Ejemplos de este tipo de atributos son el nivel de ingresos o el total de compras. Si no lo hiciéramos así, corremos el riesgo de convertir a nuestra mini dimensión, en una tabla que incluso podría crecer más que la tabla de dimensión de clientes original.

Sin embargo, esto nos restringe a tener un conjunto de rangos predefinidos; cuyos valores no deben cambiar una vez que han sido definidos, ya que resulta muy impráctico. En caso que los usuarios insistan en acceder datos específicos, con sus valores reales, como por ejemplo la calificación de un asegurado en el buró de crédito (actualizado mensualmente), éste tipo de datos deberán agregarse en la tabla de hechos. Adicionalmente, deberemos agregar su representación, en la forma de un nuevo rango de valores dentro de la mini dimensión. En la siguiente tabla (Tabla 4.3.) podemos observar un ejemplo para conformar los rangos predefinidos:

Llave	Edad	Género	Nivel de ingresos mensual
1	20-24	Masculino	<20,000
2	20-24	Masculino	20,000 – 24, 999
3	20-24	Masculino	25,000 – 29,999
…			
15	25-29	Masculino	20,000 – 24,999
16	25-29	Masculino	25,000 – 29,999

Tabla 4.3. Rangos predefinidos de mini dimensión

Al construir dicha tabla de hechos, deberemos agregar dos llaves foráneas: una que relacione la dimensión de cliente y otra que relacione la mini dimensión. Este tipo de diseño, mejora el desempeño de las consultas al proveer un punto de entrada más pequeño hacia la tabla de hechos. En el siguiente diagrama (Figura 4.3.), podemos observar el arreglo dimensional usando una mini dimensión:

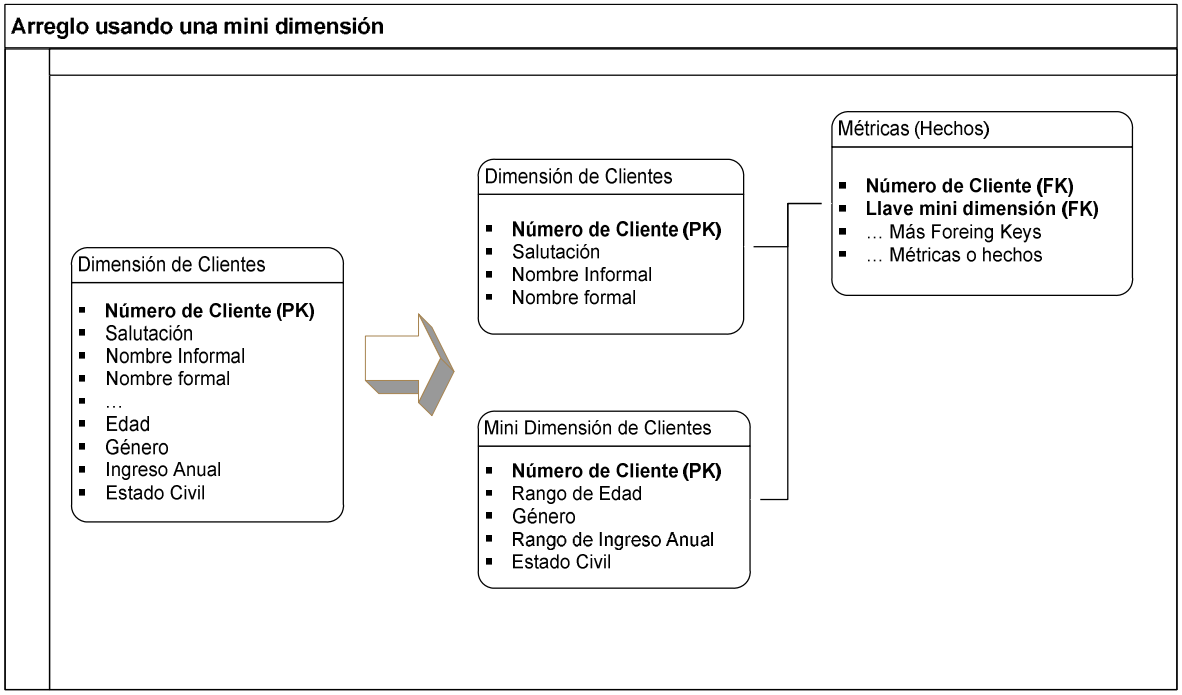


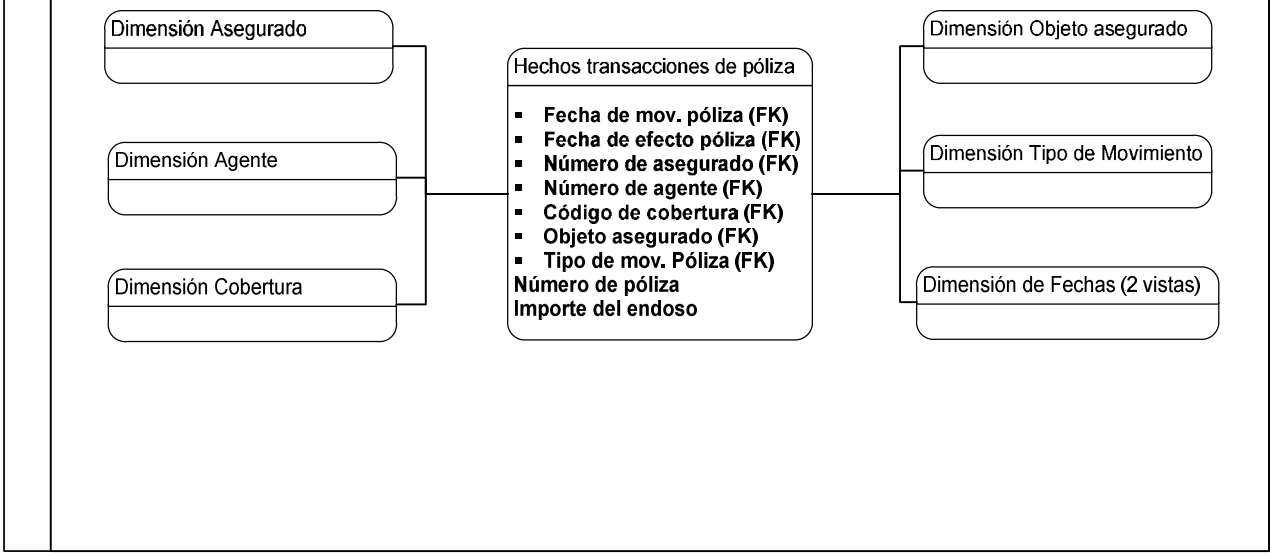
Figura 4.3. Arreglo usando una mini dimensión

4.4.1.1.4. Dimensión Degenerada

La tabla de dimensión de póliza será una dimensión degenerada, ya que contendrá el número de póliza, además de uno o dos atributos adicionales, como el grado de riesgo de la póliza. La información de cabecera asociada a la póliza, será incluida en otras tablas de dimensión; con ello evitamos el tener la información de detalle como el asegurado, fechas y coberturas incluidas en una dimensión de póliza.

Adicionalmente generaremos una dimensión para el tipo de transacción de póliza, que contendrá los distintos tipos de movimientos de póliza, con sus respectivas descripciones. Por lo general este tipo de tabla de dimensión contiene menos de cien registros.

Esquema transacciones de póliza



Matriz de Procesos

	Fecha	Asegurado	Cobertura	Asegurado	Agente	Póliza	Siniestro	Mov. Reserva	Intervinientes
Transacciones de Pólizas	X	X	X	X	X	X			
Prima relacionada a Pólizas	X	X	X	X	X	X			
Transacciones de Siniestros	X	X	X	X	X	X	X	X	X

Figura 4.5. Matriz de procesos actualizada

Diversas tablas de hechos pueden resultar de cada uno de los renglones de esta matriz de procesos, a medida que nos adentremos en la fase de implementación, deberemos tomar un subconjunto de la matriz para darle un nivel de detalle más bajo. Es en este punto donde tenemos la posibilidad de hacer una mejora, por ejemplo agregar atributos que indiquen la granularidad y métricas asociadas a cada tabla de hechos.

Tal como ocurre con la información de emisión de pólizas, el sistema operacional se encarga de capturar cierto tipo de transacciones para las reclamaciones o siniestros. Los tipos de transacciones aplicables son:

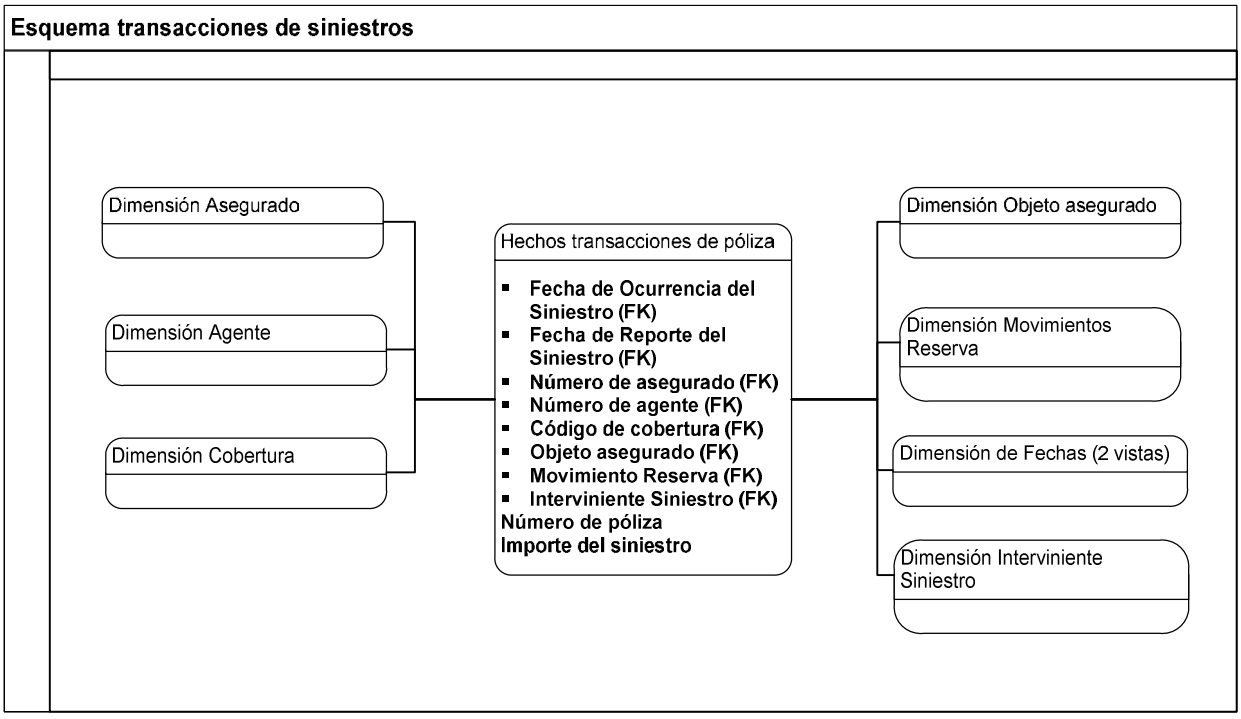
- Abrir siniestro
- Reabrir siniestro
- Cerrar siniestro
- Rechazar siniestro
- Fijar la reserva
- Reajustar la reserva
- Cerrar la reserva
- Fijar la estimación del salvamento
- Recibir pago del salvamento
- Inspección del ajustador
- Entrevista del ajustador
- Litigio abierto
- Litigio cerrado
- Hacer el pago

- Recibir el pago del siniestro

4.6. Dimensiones y técnicas

Las transacciones de siniestros cuentan con dos fechas asociadas, que juegan un rol importante. La primera es la fecha de ocurrencia del siniestro, la segunda es la fecha en que el siniestro es registrado en el sistema. Adicionalmente es importante registrar el empleado que realiza la autorización del pago al siniestro, esto es muy importante para llevar un control adecuado.

La figura 4.6 muestra el esquema de transacciones para siniestros. La dimensión de siniestros contiene la descripción del siniestro de forma codificada, es decir, para conocer la descripción debe cruzarse ésta tabla con un catálogo. La dimensión de movimientos sobre la reserva, registra todos aquellos cambios que sufre la reserva a lo largo del tiempo. Finalmente la dimensión de Intervinientes contiene el registro de todas aquellas entidades que intervienen en un siniestro, tales entidades son: asegurado, tercero, ajustador, profesional y beneficiario.



4.6.1.1. Tabla de dimensión de siniestros

Layout propuesto:

Atributo de la dimensión	Tipo de dato y longitud
Número de siniestro	VARCHAR(10)
Número de póliza	VARCHAR(14)
Versión de la Póliza	NUMERIC(8)
Tipo de Objeto	VARCHAR(10)
Secuencia del Objeto	NUMERIC(8)
Código del Producto Técnico	VARCHAR(10)
Código del Producto Comercial	VARCHAR(10)
Línea de Negocio	VARCHAR(10)
Estado geográfico donde ocurre el siniestro	VARCHAR(35)
Municipio geográfico donde ocurre el siniestro	VARCHAR(35)
Código postal del lugar donde ocurrió el siniestro	VARCHAR(5)
Fecha de alta del siniestro en el sistema	NUMERIC(8)
Fecha de ocurrencia del siniestro	VARCHAR(8)
Hora de ocurrencia del siniestro	VARCHAR(4)
Evento catastrófico del siniestro	VARCHAR(6)
Causa del siniestro (Corto circuito, muerte, etc.)	VARCHAR(4)
Riesgo (Alborotos populares, obra civil, etc.)	VARCHAR(4)
Clave del Riesgo Pericial	VARCHAR(4)

Situación del siniestro (Pendiente, Terminado, Rehabilitado, etc.)	VARCHAR(1)

4.6.1.2. Tabla de dimensión de movimientos sobre la reserva

Layout propuesto:

Atributo de la dimensión	Tipo de dato y longitud
Número de siniestro	VARCHAR(10)
Número de afectado	NUMERIC(8)
Cobertura	VARCHAR(10)
Cobertura afectada	VARCHAR(10)
Moneda del siniestro	VARCHAR(3)
Número de movimiento	VARCHAR(3)
Año - mes contable	NUMERIC(8)
Tipo de reserva (Indemnización, Gasto, Salvamento, Recuperación)	VARCHAR(1)
Tipo de movimiento del siniestro	VARCHAR(3)
Subconcepto (Pérdida, Gastos pérdida, Honorarios Servicios, etc.)	VARCHAR(5)
Estado del movimiento (Anulado, Bloqueado, etc.)	VARCHAR(2)
Código de pago (ejemplo: indemnización al asegurado)	VARCHAR(3)
Clase de reserva (Normal, Condusef, Cnsf, litigio)	VARCHAR(1)
Tipo de movimiento (Reserva,	VARCHAR(1)

Pago, Cobro)	
Fecha contable	NUMERIC(8)
Reserva Indemnización (Moneda Original)	NUMERIC(8)
Reserva Gasto(Moneda Original)	NUMERIC(8)
Reserva Salvamento (Moneda Original)	NUMERIC(8)
Reserva Recuperación Varias (Moneda Original)	NUMERIC(8)
Reserva Recuperación por Coaseguro (Moneda Original)	NUMERIC(8)
Egreso Perdida (Moneda Original)	NUMERIC(8)
Egreso Honorarios Perdida (Moneda Original)	NUMERIC(8)
Egreso Gasto Perdida (Moneda Original)	NUMERIC(8)
Egreso Honorarios Servicio (Moneda Original)	NUMERIC(8)
Egreso Gasto Servicio (Moneda Original)	NUMERIC(8)
Ingreso Salvamento (Moneda Original)	NUMERIC(8)
Ingreso Recuperación Varias (Moneda Original)	NUMERIC(8)
Ingreso Recuperación por Coaseguro (Moneda Original)	NUMERIC(8)
Siniestro Ocurrido (Moneda Original)	NUMERIC(8)
Deducible del siniestro en moneda original	NUMERIC(8)

Coaseguro a cargo del asegurado	NUMERIC(8)
Número de trabajo de tesorería	NUMERIC(8)
Número de movimiento de tesorería	NUMERIC(8)
Código de operación contable	VARCHAR(3)

4.6.1.3. Tabla de dimensión de intervinientes

Layout propuesto:

Atributo de la dimensión	Tipo de dato y longitud
Número de siniestro	VARCHAR(10)
Número de interviniente	NUMERIC(8)
Tipo de interviniente	VARCHAR(3)
Código secuencial de domicilio	VARCHAR(3)
Código de convenio	NUMERIC(10)
Estatus del interviniente	VARCHAR(1)
Indicador de fraude	VARCHAR(1)
Código de profesional	VARCHAR(3)
Fecha de alta de interviniete	NUMERIC(8)
Fecha de baja de interviniete	NUMERIC(8)

4.7. Errores a evitar

Resulta bastante provechoso agregar una última sección donde estableceremos ciertos criterios y límites que los diseñadores no deben traspasar. Hasta el momento hemos presentado conceptos que son aplicables bajo situaciones que comúnmente suceden al diseñar un Data Warehouse específico para una compañía de seguros. A continuación listamos en orden descendente, los diez errores que debemos evitar al diseñar un modelo dimensional; de no evitar las situaciones listadas podríamos estar comprometiendo seriamente el proyecto de implementación del Data Warehose:

Error 10: Colocar atributos de texto utilizados para generar constraints y grupos en una tabla de hechos. Las métricas producto de un proceso de negocio, y obtenidas desde un sistema operacional pertenecen a la tabla de hechos. Los atributos de texto que describen el contexto de las métricas, van en las diversas tablas de dimensión. Finalmente, realizaremos una revisión campo por campo sobre los códigos y elementos seudo numéricos, colocándolos en la tabla de hechos si son más parecidos a una métrica, y por el contrario los colocaremos en una tabla de dimensión si son más parecidos a las descripciones físicas de algo. No debemos dejar el texto verdadero, especialmente los comentarios de campo, en la tabla de hechos. Necesitamos tener éstos atributos de texto fuera del cauce principal del Data Warehouse y si dentro de las tablas de dimensión.

Error 9: Limitar los atributos descriptivos en las tablas de dimensión, para ahorrar espacio. Puede ser que lleguemos a pensar que somos buenos diseñadores por tener el tamaño de las tablas de dimensión bajo control. Sin embargo, en todo Data Warehouse las tablas de dimensión son geométricamente más pequeñas que las tablas de hechos. Teniendo una tabla de dimensión de productos de 100 MB resulta insignificante, si la tabla de hechos es 100 veces de grande. El trabajo como diseñador de DW es proveer tanto contexto descriptivo en cada tabla de dimensión como sea posible. Debemos asegurarnos que cada código esté complementado con texto descriptivo legible. Recordemos que los atributos de texto contenidos en las tablas de dimensión constituyen la interfaz de usuario para explorar datos y generar filtros, adicionalmente serán los títulos de columna en los reportes generados.

Error 8: Jerarquías divididas y niveles de jerarquía en múltiples dimensiones. Una jerarquía es una serie de relaciones muchos a uno, conectadas en cascada. Por ejemplo, muchos productos pueden asociarse una sola marca de fábrica; y muchas marcas de fábrica pueden asociarse a una sola categoría. Si una tabla de dimensión está expresada al nivel más bajo del granularidad (por ejemplo el producto), entonces todos los niveles más altos de la jerarquía puede ser expresados como valores únicos de renglón de producto. Los usuarios entienden jerarquías, nuestro trabajo es presentar las jerarquías de la manera más natural y eficiente. Una jerarquía pertenece a una tabla de dimensión. Finalmente, si existe más de una asociación de forma simultánea para una dimensión, en la mayoría de los casos es perfectamente razonable incluir jerarquías múltiples dentro de la misma tabla de dimensión, mientras que la dimensión haya sido definida con el nivel más bajo de granularidad posible (y las jerarquías son únicamente etiquetadas).

Error 7: Ignorar la necesidad de dar seguimiento a los cambios de atributos. Contrario a la creencia popular, los usuarios de negocio a menudo desean entender el impacto de los cambios que ocurren a un subconjunto de atributos de tablas de dimensión. Es poco probable que los usuarios establezcan tablas de dimensión con atributos que reflejen siempre el estado actual de la organización. Existen

tres técnicas que podemos emplear para tablas de dimensión con atributos de cambio lento, no debemos encasillarnos en una sola técnica. Asimismo, si un grupo de atributos cambia rápidamente, no debemos retrasar la división de la tabla de dimensión, para generar una mini dimensión más volátil. Supongamos que la tabla de dimensión de productos, contiene atributos determinados que llamaremos parámetros estándares. Al inicio del proceso del diseño nos han asegurando que estos parámetros estándares son fijos para toda la vida del producto. Sin embargo, después de dar el segundo vistazo al diseño del DW, descubrimos que dichos atributos cambian varias veces al año para cada producto. Tarde o temprano, deberemos separar la dimensión de producto en dos dimensiones. El nuevo parámetro estándar de la dimensión de producto se guardará en la dimensión original de producto.

Error 6: Solucionar los problemas de desempeño, añadiendo más hardware. Las tabla sumarizadas, o tablas derivadas de resumen, son la manera más rentable de mejorar el desempeño de las consultas. La mayoría de las empresas que comercializan herramientas de consulta, cuentan con soporte para el uso de tablas sumarizadas que dependen de construir modelos dimensionales explícitos. El añadir hardware costoso debe hacerse como parte de un programa de balanceo, que incluye construir tablas derivadas de resumen, crear índices, seleccionar un DBMS eficiente, incrementando el tamaño de la memoria real, la velocidad del CPU y finalmente, agregar paralelismo al nivel de hardware.

Error 5: Uso de claves operacionales para unir las tablas de dimensión con las tablas de hechos. Los diseñadores principiantes son a veces muy literales al diseñar las llaves primarias de las tablas de dimensión, que conectan con las llaves foráneas de las tablas de hechos. Resulta contraproducente declarar un conjunto completo de atributos de cómo llave de la tabla de dimensión y luego utilizarlas como la base para la unión física de la tabla de hechos. Esto incluye la desafortunada práctica de declarar la clave de la dimensión para ser la clave operacional, otra práctica igualmente mala es el uso de una fecha. Lo mejor que se puede hacer es sustituir la llave física por una llave sustituta entera que numere de forma secuencial desde 1 hasta N, donde N es el número total renglones en la tabla de dimensión.

Error 4: Negligencia al declarar y luego no respetar la granularidad de la tabla de hechos. Todos los diseños dimensionales deben comenzar con el proceso de negocio que genera las métricas de desempeño. En segundo lugar, especificar la granularidad exacta de dichos datos. Construir tablas de hechos al nivel más bajo de granularidad. En tercer lugar deberemos rodear dichas métricas con tablas de dimensión que realmente tengan ese mismo nivel de granularidad. Apegarse a la granularidad definida es un paso crucial en el diseño de un modelo dimensional. Un sutil pero serio error en el diseño dimensional es agregar métricas que sean convenientes sólo para la tabla de hechos, tal como filas que describen los totales para una duración extendida o una gran área geográfica. Aunque estos hechos adicionales son bien conocidos y aunque aparentemente simplifiquen algunas aplicaciones, causan estragos, debido a que todas las sumarizaciones automáticas que se realicen a través de las tablas de dimensiones, producirán resultados incorrectos. Cada nivel de granularidad requiere su propia tabla de hechos.

Error 3: Basar el modelo dimensional en un reporte específico. Un modelo dimensional no tiene nada ver con un reporte propuesto. Mejor dicho, es un modelo sobre los procesos de negocio. Las métricas son la base de las tablas de hechos. Las dimensiones que son apropiadas para una tabla de hechos dada, son aquellas que dan el contexto físico que describe las circunstancias de las métricas. Un

modelo dimensional debe estar basado en la física de los procesos de negocio, con lo cual se vuelve independiente de la forma como un usuario define un reporte.

Error 2: Esperar que los usuarios realicen consultas de datos al más bajo nivel en un formato normalizado. Los datos de más bajo-nivel son siempre los más dimensionales y deben ser la base del diseño dimensional. Los datos que han sido agregados en todas direcciones, han sido privados de algunas de sus dimensiones. No es posible construir un data mart con datos agregados y esperar que los usuarios naveguen con la tercera forma normal para obtener los datos de detalle. Los modelos normalizados pueden ser útiles para segmentar los datos, pero nunca deben ser utilizados para presentar los datos a los usuarios del negocio.

Error 1: Fallar al conformar hechos y dimensiones a través de tablas de hechos separadas. Si contamos con una métrica tal como el importe de ganancias sobre la prima, disperso en dos o más data marts, y que además proviene de diversos sistemas fuente; entonces debemos poner especial cuidado para asegurar que las definiciones técnicas de estas métricas concuerdan exactamente. Si las definiciones no concuerdan exactamente, entonces no deben ser ambas referidas como la misma métrica. Finalmente, si dos o más tablas de hechos tienen la misma dimensión, entonces debemos ser muy metódicos para asegurar que estas dimensiones sean idénticas o en su defecto los subconjuntos deben ser cuidadosamente elegidos.

Conclusiones

El Data Warehouse se ha convertido en una base sólida para el crecimiento sostenido de muchas organizaciones alrededor del mundo, convirtiéndose en una ventaja competitiva que debe ser aprovechada al máximo. Permite convertir datos dispersos provenientes de muy diversas fuentes, en información relevante para la organización.

Sin embargo el éxito de un Data Warehouse depende en gran medida de un diseño adecuado, que permita cubrir las necesidades de información de la organización y de las diversas áreas que la componen. Para cumplir esta importante misión, el diseñador debe poner especial atención a los detalles ocultos en los requerimientos de los usuarios.

Para lograr lo anterior, es muy importante que el diseñador domine los conceptos y técnicas recomendadas, una vez hecho esto estaremos en posición de concentrarnos en otras actividades, igualmente importantes, que ocurren a lo largo del ciclo de vida de un proyecto de implementación de un Data Warehouse.

En la sección final se citan ejemplos puntuales para el diseño de modelos dimensionales aplicados a una empresa de seguros en México, los cuales representan la culminación de los conceptos y técnicas abordados a lo largo del presente trabajo de tesis, así como la aplicación de lecciones aprendidas y de la experiencia obtenida a lo largo de diversos proyectos de implementación de Data Warehouses.

Bibliografía

William H. Inmon, Building the Data Warehouse. John Wiley & Sons, U.S.A. 1992.

Imhoff, et al, Mastering Data Warehouse Design. John Wiley & Sons, U.S.A. 2003.

Ralph Kimball & Margy Ross. The Data Warehouse Tool Kit. John Wiley & Sons, U.S.A. 2002.

Montoya Manfredi, Ulises. Derecho Comercial Tomo II. Cultural Cuzco S.A. 1986.

Garrigues, Joaquín. Curso de Derecho Mercantil Tomo IV. Editorial Temis 1987.

Rodríguez Pastor, Carlos. Derecho de Seguros y Reaseguros. Fundación MJ Bustamante de la Fuente 1987.

AMIS. Tipos de Seguros. Disponible en:
http://www.amis.org.mx/informaweb/Documentos/Archivos/sector_TipoSeguro.html

Bill Inmon & Claudia Imhoff, Building the Operational Data Store. John Wiley & Sons, U.S.A. 1996.

Ralph Kimball & Joe Caserta, The Data Warehouse ETL Toolkit. John Wiley & Sons, U.S.A. 2004.

Glosario

B

BASE DE DATOS (DATA BASE)

Conjunto de datos no redundantes, almacenados en un soporte informático, organizados de forma independiente de su utilización y accesibles simultáneamente por distintos usuarios y aplicaciones. La diferencia de una BD respecto a otro sistema de almacenamiento de datos es que éstos se almacenan en la BD de forma que cumplen tres requisitos básicos: no redundancia, independencia y concurrencia.

BLOB (Binary Large Object)

Objeto binario grande. Entre los tipos de datos que contienen los campos BLOB están: binarios, memo, memo con formato, de imagen, de sonido y OLE.

C

CLIENTE/SERVIDOR

Arquitectura de sistemas de información en la que los procesos de una aplicación se dividen en componentes que se pueden ejecutar en máquinas diferentes. Modo de funcionamiento de una aplicación en la que se diferencian dos tipos de procesos y su soporte se asigna a plataformas diferentes.

CODIFICACION

a) Transformación de un mensaje en forma codificada, es decir, especificación para la asignación unívoca de los caracteres de un repertorio (alfabeto, juego de caracteres) a los de otro repertorio.

b) Conversión de un valor analógico en una señal digital según un código prefijado.

D

DETECCION DE DESVIACION

Normalmente, para la detección de desviación en bases de datos grandes se usa la información explícita externa a los datos, así como las limitaciones de integridad o modelos predefinidos. En un método lineal por contraste, se enfoca el problema desde el interior de los datos, usando la redundancia implícita de los datos. Aquí se simula un mecanismo familiar a los seres humanos: después de ver una serie de datos similares, un elemento que perturba la serie se considera una excepción.

DICCIONARIO DE DATOS

Descripción lógica de los datos para el usuario. Reúne la información sobre los datos almacenados en la BD (descripciones, significado, estructuras, consideraciones de seguridad, edición y uso de las aplicaciones, etc.).

DIRECTORIO DE DATOS

Es un subsistema del sistema de gestión de base de datos que describe dónde y cómo se almacenan los datos en la BD (modo de acceso y características físicas de los mismos).

DRILL-DOWN

Obtención de información más detallada sobre un conjunto de información en el cual se está trabajando. Ejemplo: Si se está mirando el Activo, obtener todas las cuentas del activo.

E

EXTRANET

Constituye un servicio de comunicación orientado a un público focalizado sobre el formato de los sistemas Web, operando sobre la red Internet. Ejemplo: Una casa de ventas de productos varios, implementa un sistema de Ofertas, Consulta a Catálogos, Bancos de Datos y Compras a sus clientes preferenciales.

G

GUI

Graphic User Interface, es la interfase gráfica, a través de la cual, los usuarios interactúan con un sistema o aplicación de software.

I

IN-HOUSE

Aplicable a la realización de un servicio de outsourcing en las instalaciones de la organización que contrata el servicio.

INCONSISTENCIA

El contenido de una base de datos es inconsistente si dos datos que deberían ser iguales no lo son. Por ejemplo, un empleado aparece en una tabla como activo y en otra como jubilado.

INTEGRIDAD

Condición de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, sólo por el personal autorizado.

INTERNET

Término usado para referirse a la red más grande del mundo, que conecta miles de redes con alcance mundial. Está creando una cultura que basándose en la simplicidad, investigación y estandarización fundamentado en usos de la vida real, está cambiando la forma de ver y hacer

muchas de las tareas actuales. Mucha de la tecnología de punta en redes está proviniendo de la comunidad Internet.

INTRANET

Constituye un servicio de comunicación de los sistemas de información corporativos orientados a su personal, sobre el formato de los sistemas Web, operando sobre la red Internet. Ejemplo: El sistema contable de una empresa de ventas de productos de ferretería, tipo Home Center.

M

MAP

a) Conjunto de datos

b) Lista de datos u objetos, tal como actualmente están almacenados en memoria o en disco.

c) Transferir un conjunto de objetos de un lugar a otro. Por ejemplo, los módulos de programas en el disco son proyectados ("mapeados") en la memoria. Una imagen gráfica en memoria es proyectada en la pantalla.

d) Relacionar un conjunto de objetos con otro. Por ejemplo, una estructura de base de datos lógica se proyecta sobre la base de datos física.

Mapping.- Proyección, correspondencia, transformación.

MODELAMIENTO PREDICTIVO (Inteligencia Artificial)

Las herramientas de modelado predictivo permiten realizar relaciones complejas o modelos desde un archivo de datos.

Una de las principales diferencias entre los modelos estadísticos y los modelos de inteligencia artificial, es cómo miden su error. Los primeros miden el error relativo tal como el modelo "adapta" los datos, mientras que los segundos, miden el error relativo a los datos aún invisibles (Error predictivo).

Segundo, los modelos estadísticos tienen dificultades al dar datos contradictorios o desordenados, es decir, los datos deben estar limpios y deben existir las correlaciones consistentes. Viceversa, las herramientas de inteligencia artificial buscan "generalizar" relaciones para proporcionar el resultado más probable.

El modelado abductivo (argumento en que la premisa mayor es evidente y la menor probable, pero más creíble que la conclusión) usa funciones poli nómicas para describir las relaciones al interior de los datos. Esta metodología facilita una variable de entrada para ser ponderado más de una vez. Adicional, sólo se incluyen los términos que significativamente contribuyen al rendimiento.

Los modelos predictivos pueden usarse para el soporte de decisión o presentando sub-rutinas para desarrollar aplicaciones predictivas a clientes. Las capacidades de los modelos predictivos pueden mejorarse si los archivos de datos se mejoran con tantas variables de entrada como sea posible.

P

PROGRAMACION GENETICA (PG)

El paradigma de Programación Genética, propuesto por John Koza, es una extensión de los algoritmos genéticos que difiere de éstos en la forma en que representa a los individuos de la población, pues utiliza programas de computadora en lugar de cadenas de longitud fija.

La meta de la PG es lograr que las computadoras aprendan a resolver problemas sin ser explícitamente programadas, generando soluciones a problemas a partir de la inducción de programas. El programador no especifica el tamaño, forma y complejidad estructural de los programas-solución, sino que los programas evolucionan hasta generar soluciones satisfactorias.

Dentro del espacio de posibles programas de computadora, la inducción de programas involucra el descubrimiento inductivo de un programa que produzca alguna salida deseada, cuando se le presenta alguna entrada en particular. Y esto es precisamente lo que la metodología de PG realiza de una manera sistematizada.

Con base en este planteamiento, un programa puede ser llamado una fórmula, un plan, una estrategia de control, un procedimiento computacional, etc. Similarmente, las entradas del programa pueden ser llamadas variables independientes, variables de estado, valores de sensores, argumentos de una función, etc. A su vez las salidas del programa pueden denominarse variables dependientes, un movimiento, un actuador, el valor regresado por una función, etc.

En programación genética, poblaciones de cientos, miles y decenas de miles de programas o más, se desarrollan genéticamente. Este desarrollo se hace usando el principio darwiniano de supervivencia del más apto y las operaciones genéticas primarias de Reproducción y Cruce (o recombinación sexual).

Características.

a) La forma de árbol de los programas de computadora. Para evitar el crecimiento descontrolado de un programa, en muchos casos pueden encapsularse subárboles en hojas individuales.

b) En cada etapa de este proceso altamente paralelo, descentralizado y localmente controlado, el estado consiste únicamente de la población actual de individuos.

c) La variabilidad dinámica de los programas en la búsqueda de la solución. A menudo, es difícil y no natural tratar de especificar o restringir el tamaño y forma de una solución eventual de antemano. Más aún, el hacerlo reduce el tamaño de la ventana por la cual el sistema ve al mundo, pudiendo evitar encontrar la solución final o, peor aún, encontrar una solución predeterminada.

d) Ausencia o un menor pre-procesamiento de entradas y post-procesamiento de salidas. Típicamente, las entradas, los resultados intermedios y las salidas son expresados de manera directa en la terminología natural del dominio del problema. Los programas producidos por la PG consisten de funciones que son naturales al dominio del problema.

e) En la PG las estructuras que sobreviven a la adaptación son activas. Estas no son códigos pasivos (cromosomas) de la solución de un problema. Las estructuras de PG son estructuras activas capaces de ser ejecutadas en su forma actual.

En conclusión, la programación genética sistematiza el problema de inducción de programas, es decir, la generación automática de un programa que solucione un problema dado. La importancia de la inducción de programas se hace evidente al observar que todos los problemas se pueden reformular como un programa de computadora.

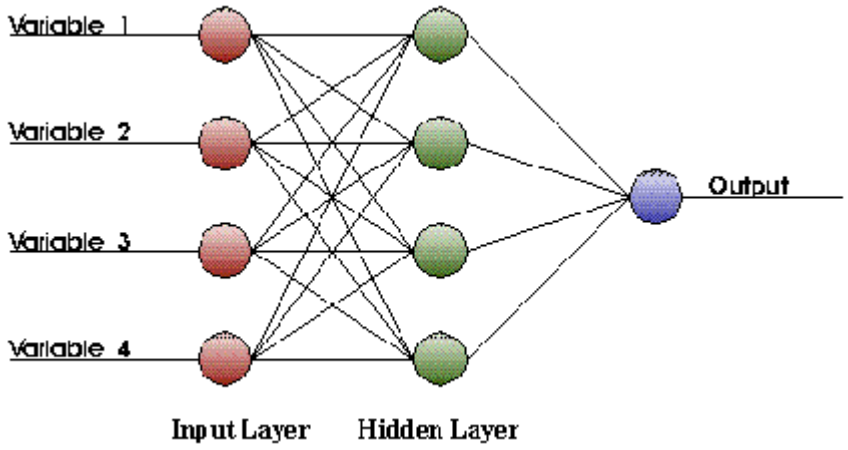
La metodología de PG proporciona características muy importantes para el diseño, de manera robusta, de sistemas que actúen sobre condiciones inestables en ambientes cambiantes.

R

RED NEURONAL ARTIFICIAL

Son abstracciones más o menos complejas que tratan de emular el funcionamiento de las redes neuronales del cerebro humano. La mayoría de las veces son modelos teóricos que se plasman en programas de ordenador y unas pocas modelos sobre silicio para aprovechar la velocidad de proceso paralelo de estas arquitecturas.

Las neuronas individuales se conectan con otros para formar una "Red" de conexiones. La conexión individual entre dos neuronas se pondera para proveer su contribución al pronóstico del rendimiento deseado.



REDUNDANCIA

Repetición de los mismos datos en varios lugares.

REPOSITORIO

Base de datos central en herramientas de ayuda al desarrollo. El repositorio amplía el concepto de diccionario de datos para incluir toda la información que se va generando a lo largo del ciclo de vida del sistema, como por ejemplo:

Componentes de análisis y diseño (diagramas de flujo de datos, diagramas entidad-relación, esquemas de bases de datos, diseños de pantallas, etc.), estructuras de programas, algoritmos, etc. En algunas referencias se le denomina Diccionario de recursos de información.

S

SCORING

Calificación que se le da a un grupo de clientes/productos que mide la propensión a compras, ventas, retiro, llegadas, etc.

SISTEMA DE GESTION DE BASE DE DATOS

Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos. Acepta pedidos de datos desde un programa de aplicación y le ordena al sistema operativo transferir los datos apropiados.

Cuando se usa un sistema de gestión de base de datos, SGDB, (en inglés DBMS), los sistemas de información pueden ser cambiados más fácilmente a medida que cambien los requerimientos de la organización. Nuevas categorías de datos pueden agregarse a la base de datos sin dañar el sistema existente.

SISTEMA DE INFORMACION (SI)

Conjunto de elementos físicos, lógicos, de comunicación, datos y personal que, interrelacionados, permiten el almacenamiento, transmisión y proceso de la información.

SOLARIS

Sistema operativo multiproceso, multiprograma y multiusuario. Software diseñado por AT&T para ingeniería de telecomunicación. Ha sido el primer sistema operativo concebido con independencia de los fabricantes. Posee una gran facilidad para adaptarse a ordenadores con diferentes arquitecturas, siendo ampliamente autónomo respecto del hardware. Está escrito en lenguaje de alto nivel C.

SQL (Structured Query Language)

Lenguaje de interrogación normalizado para bases de datos relacionales. El SQL es un lenguaje de alto nivel, no procedural, normalizado, que permite la consulta y actualización de los datos de BD relacionales. Se ha convertido en el estándar para acceder a BD relacionales. La primera versión se aprobó como norma ISO en 1987 y la segunda, conocida como SQL2 y vigente actualmente, en 1992.

Actualmente se trabaja en la norma SQL3 que soportará bases de datos orientadas a objeto y bases de datos activas. El SQL facilita un lenguaje de definición de datos y un lenguaje de manipulación de datos. Además, incluye una interfase que permite el acceso y manipulación de la BD a usuarios finales.

T

TERABYTE (TB)

Unidad de medida que equivale a 1024 GB.