MODELO DE DESARROLLO PARA EL ALMACENAMIENTO, ANÁLISIS E INTERPRETACIÓN DE INFORMACIÓN APLICADO EN UN SISTEMA DE INFORMACIÓN PARA VIGILANCIA EPIDEMIOLÓGICA

"MAISIVE"

ANEXO A

MARCO CONCEPTUAL

Ricardo Emilio Lombana Quiñones
Richard Andrés Rojas Rosero

Director: Ing. Diego Mauricio López Gutiérrez

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de Investigación en Ingeniería de Sistemas Telemáticos

Popayán, Febrero 2004

TABLA DE CONTENIDO

INDICE DE TABLAS

INDICE DE ILUSTRACIONES

1. DATA WAREHOUSING

1.1. CONCEPTO DE DATA WAREHOUSING

Data warehousing es el centro de la arquitectura para los sistemas de información en la década de los '90. Soporta el procesamiento informático al proveer una plataforma sólida, a partir de los datos históricos para hacer el análisis de información. Facilita la integración de sistemas de aplicación no integrados. Organiza y almacena los datos que se necesitan para el procesamiento analítico, informático sobre una amplia perspectiva de tiempo.

La innovación de la Tecnología de Información dentro de un ambiente data warehousing, puede permitir a cualquier organización hacer un uso más óptimo de los datos, como un ingrediente clave para un proceso de toma de decisiones más efectivo. Las organizaciones tienen que aprovechar sus recursos de información para crear la información de la operación del negocio, pero deben considerarse las estrategias tecnológicas necesarias para la implementación de una arquitectura completa de data warehouse.

Se puede caracterizar un data warehouse haciendo un contraste de cómo los datos de un negocio almacenados en un data warehouse, difieren de los datos operacionales usados por las aplicaciones de producción.

Base de Datos Operacional	Data Warehouse
Datos Operacionales	Datos del negocio para Información
Orientado a la aplicación	Orientado al sujeto
Actual	Actual + histórico
Detallada	Detallada + más resumida
Cambia continuamente	Estable

Tabla 1. Concepto de Data warehouse

Diferentes tipos de información

El ingreso de datos en el data warehouse viene desde el ambiente operacional en casi todos los casos. El data warehouse es siempre un almacén de datos transformados y separados físicamente de la aplicación donde se encontraron los datos en el ambiente operacional.

Un data warehouse se crea al extraer datos desde una o más bases de datos de aplicaciones operacionales. La data extraída es transformada para eliminar inconsistencias y resumir si es necesario y luego, cargadas en el data warehouse. El proceso de transformar, crear el detalle de tiempo variante, resumir y combinar los extractos de datos, ayudan a crear el ambiente para el acceso a la información Institucional. Este nuevo enfoque ayuda a las personas individuales, en todos los niveles de la empresa, a efectuar su toma de decisiones con más responsabilidad.

1.2. SISTEMAS DE INFORMACIÓN

Los sistemas de información se han dividido de acuerdo al siguiente esquema:

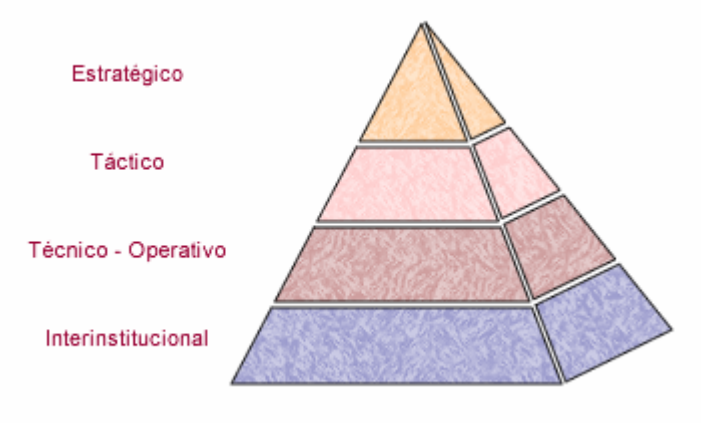

Figura 1. Sistemas de Información

Sistemas Estratégicos: Orientados a soportar la toma de decisiones, facilitan la labor de la dirección, proporcionándole un soporte básico, en forma de mejor información, para la toma de

decisiones. Se caracterizan porque son sistemas sin carga periódica de trabajo, es decir, su utilización no es predecible, al contrario de los casos anteriores, cuya utilización es periódica.

Destacan entre estos sistemas: los Sistemas de Información Gerencial (MIS), Sistemas de Información Ejecutivos (EIS), Sistemas de Información Georeferencial (GIS), Sistemas de Simulación de Negocios (BIS y que en la práctica son sistemas expertos o de Inteligencia Artificial-AI).

Sistemas Tácticos: Diseñados para soportar las actividades de coordinación de actividades y manejo de documentación, definidos para facilitar consultas sobre información almacenada en el sistema, proporcionar informes y, en resumen, facilitar la gestión independiente de la información por parte de los niveles intermedios de la organización.
Destacan entre ellos: los Sistemas Ofimáticos (OA), Sistemas de Transmisión de Mensajería (E-mail y Fax Server), coordinación y control de tareas (Work Flow) y tratamiento de documentos (Imagen, Trámite y Bases de Datos Documentarios).

Sistemas Técnico-Operativos: Que cubren el núcleo de operaciones tradicionales de captura masiva de datos (Data Entry) y servicios básicos de tratamiento de datos, con tareas predefinidas (contabilidad, facturación, almacén, presupuesto, personal y otros sistemas administrativos). Estos sistemas están evolucionando con la irrupción de censores, autómatas, sistemas multimedia, bases de datos relacionales más avanzadas y data warehousing.

Sistemas Interinstitucionales: Este último nivel de sistemas de información recién está surgiendo, es consecuencia del desarrollo organizacional orientado a un mercado de carácter global, el cual obliga a pensar e implementar estructuras de comunicación más estrechas entre la organización y el mercado (Empresa Extendida, Organización Inteligente e Integración Organizacional), todo esto a partir de la generalización de las redes informáticas de alcance nacional y global (INTERNET), que se convierten en vehículo de comunicación entre la organización y el mercado, no importa dónde esté la organización (INTRANET), el mercado de la institución (EXTRANET) y el mercado (Red Global).

Sin embargo, la tecnología data warehousing basa sus conceptos y diferencias entre dos tipos fundamentales de sistemas de información en todas las organizaciones: los sistemas técnico-operacionales y los sistemas de soporte de decisiones. Este último es la base de un data warehouse.

1.2.1. SISTEMAS TÉCNICO-OPERACIONALES

Como indica su nombre, son los sistemas que ayudan a manejar la empresa con sus operaciones cotidianas. Estos son los sistemas que operan sobre el "backbone" (columna vertebral) de cualquier empresa o institución, entre las que se tiene sistemas de ingreso de órdenes, inventario, fabricación, planilla y contabilidad, entre otros.

Debido a su volumen e importancia en la organización, los sistemas operacionales siempre han sido las primeras partes de la empresa a ser computarizados. A través de los años, estos sistemas operacionales se han extendido, revisado, mejorado y mantenido al punto que hoy, ellos son completamente integrados en la organización.

Desde luego, la mayoría de las organizaciones grandes de todo el mundo, actualmente no podrían operar sin sus sistemas operacionales y los datos que estos sistemas mantienen.

1.2.2. SISTEMAS DE SOPORTE DE DECISIONES

Por otra parte, hay otras funciones dentro de la empresa que tienen que ver con el planeamiento, previsión y administración de la organización. Estas funciones son también críticas para la supervivencia de la organización, especialmente en nuestro mundo de rápidos cambios.
Las funciones como "planificación de marketing", "planeamiento de ingeniería" y "análisis financiero", requieren, además, de sistemas de información que los soporte. Pero estas funciones son diferentes de las operacionales y los tipos de sistemas y la información requerida son también diferentes. Las funciones basadas en el conocimiento son los sistemas de soporte de decisiones.

Estos sistemas están relacionados con el análisis de los datos y la toma de decisiones, frecuentemente, decisiones importantes sobre cómo operará la empresa, ahora y en el futuro. Estos sistemas no sólo tienen un enfoque diferente al de los operacionales, sino que, por lo general, tienen un alcance diferente.

Mientras las necesidades de los datos operacionales se enfocan normalmente hacia una sola área, los datos para el soporte de decisiones, con frecuencia, toma un número de áreas diferentes y necesita cantidades grandes de datos operacionales relacionadas. Son estos sistemas sobre los se basa la tecnología data warehousing.

1.3. DIFERENCIAS ENTRE UN SISTEMA OPERACIONAL (OLTP) Y UN ALMACEN DE DATOS (DATA WAREHOUSE)

Los sistemas tradicionales de transacciones y las aplicaciones de Data Warehousing son polos opuestos en cuanto a sus requerimientos de diseño y sus características de operación. Es de suma importancia comprender perfectamente estas diferencias para evitar caer en el diseño de un Data Warehouse como si fuera una aplicación de transacciones en línea (OLTP).

Las aplicaciones de OLTP están organizadas para ejecutar las transacciones para los cuales fueron hechos, como por ejemplo: mover dinero entre cuentas, un cargo o abono, una devolución de inventario, etc. Por otro lado, un Data Warehouse está organizado en base a conceptos, como por ejemplo: clientes, facturas, productos, etc.

Otra diferencia radica en el número de usuarios. Normalmente, el número de usuarios de un Data Warehouse es menor al de un OLTP. Es común encontrar que los sistemas transaccionales son accesados por cientos de usuarios simultáneamente, mientras que los Data Warehouse sólo por decenas. Los sistemas de OLTP realizan cientos de transacciones por segundo mientras que una sola consulta de un Data Warehouse puede tomar minutos. Otro factor es que frecuentemente los sistemas transaccionales son menores en tamaño a los Data Warehouses, esto es debido a que un Data Warehouse puede estar formado por información de varios OLTP´s.

Existen también diferencia en el diseño, mientras que el de un OLPT es extremadamente normalizado, el de un Data Warehouse tiende a ser desnormalizado. El OLTP normalmente está formado por un número mayor de tablas, cada una con pocas columnas, mientras que en un Data Warehouse el número de tablas es menor, pero cada una de éstas tiende a ser mayor en número de columnas.

Los OLTP son continuamente actualizados por los sistemas operacionales del día con día, mientras que los Data Warehouse son actualizados en batch de manera periódica.

En la Tabla 2 se enmarcan las principales diferencias entre los sistemas operacionales y las aplicaciones de Data warehouse.

Sistema Operacional	Almacén de Datos
Almacena datos actuales	Almacena datos históricos
Almacena datos de detalle	Almacena datos de detalle y datos agregados a distintos niveles
Bases de datos medianas (100Mb – 1Gb)	Bases de datos grandes (100Gb – 1Tb)
Los datos son dinámicos (actualizables)	Los datos son estáticos
Los procesos (transacciones) son repetitivos	Los procesos no son previsibles
El número de transacciones es elevado	El número de transacciones es medio o bajo
Tiempo de respuesta pequeño (segundos)	Tiempo de respuesta variable (segundos – horas)
Dedicado al procesamiento de transacciones	Dedicado al análisis de datos
Orientado a los procesos de la organización	Orientado a la información relevante
Soporta decisiones diarias	Soporta decisiones estratégicas
Sirve a muchos usuarios (administrativos)	Sirve a técnicos de dirección

Tabla 2. Diferencias entre un sistema operacional y un Data warehouse

1.4. CARACTERÍSTICAS DE UN DATA WAREHOUSE

Entre las principales características se tienen:

- Orientado al tema
- Integrado
- De tiempo variable
- No volátil

1.4.1. ORIENTADO A TEMAS

Una primera característica del data warehouse es que la información se clasifica en base a los aspectos que son de interés para la empresa. Siendo así, los datos tomados están en contraste con los clásicos procesos orientados a las aplicaciones. En la Figura 2 se muestra el contraste entre los dos tipos de orientaciones.

El ambiente operacional se diseña alrededor de las aplicaciones y funciones tales como préstamos, ahorros, tarjeta bancaria y depósitos para una institución financiera. Por ejemplo, una aplicación de ingreso de órdenes puede acceder a los datos sobre clientes, productos y cuentas. La base de datos combina estos elementos en una estructura que acomoda las necesidades de la aplicación.

En el ambiente data warehousing se organiza alrededor de sujetos tales como cliente, vendedor, producto y actividad. Por ejemplo, para un fabricante, éstos pueden ser clientes, productos, proveedores y vendedores. Para una universidad pueden ser estudiantes, clases y profesores. Para un hospital pueden ser pacientes, personal médico, medicamentos, etc.

La alineación alrededor de las áreas de los temas afecta el diseño y la implementación de los datos encontrados en el data warehouse. Las principales áreas de los temas influyen en la parte más importante de la estructura clave.

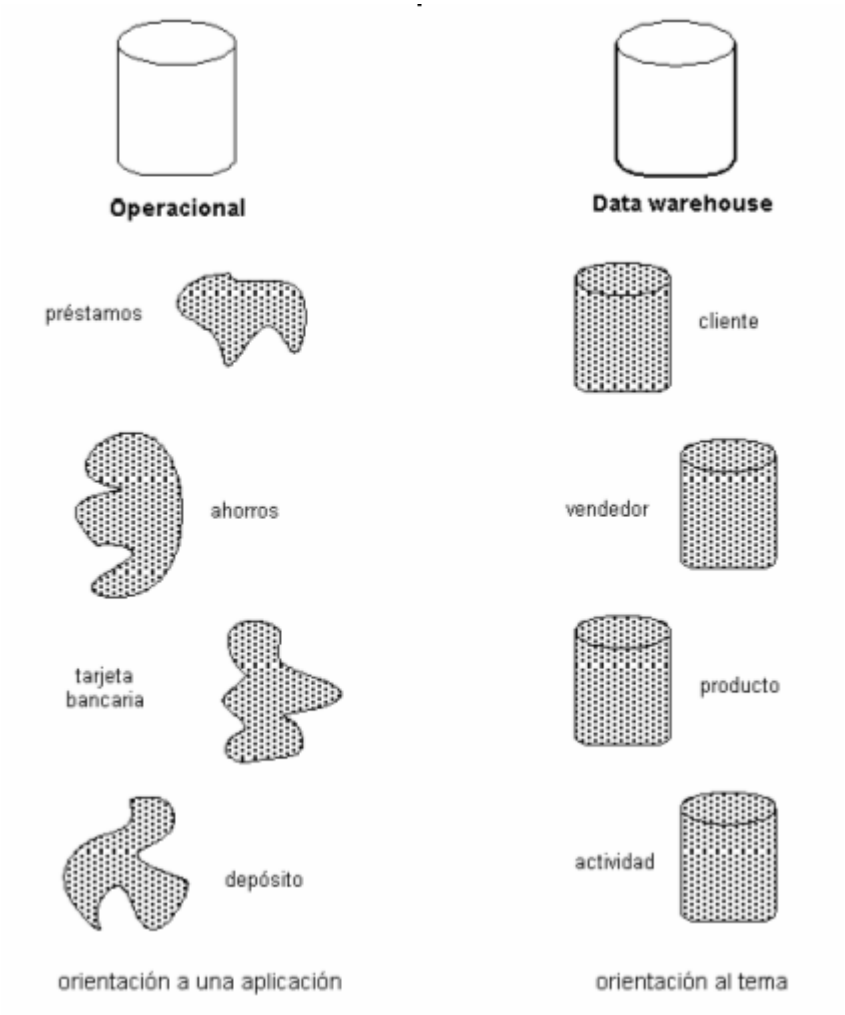

Figura 2. Orientación del Data warehouse al tema

Las aplicaciones están relacionadas con el diseño de la base de datos y del proceso. En data warehousing se enfoca el modelamiento de datos y el diseño de la base de datos. El diseño del proceso (en su forma clásica) no es separado de este ambiente.

Las diferencias entre la orientación de procesos y funciones de las aplicaciones y la orientación a temas, radican en el contenido de la data a nivel detallado. En el data warehouse se excluye la información que no será usada por el proceso de sistemas de soporte de decisiones, mientras que la información de las orientadas a las aplicaciones, contiene datos para satisfacer de inmediato los

requerimientos funcionales y de proceso, que pueden ser usados o no por el analista de soporte de decisiones.

Otra diferencia importante está en la interrelación de la información. Los datos operacionales mantienen una relación continua entre dos o más tablas basadas en una regla comercial que está vigente. Las del data warehouse miden un espectro de tiempo y las relaciones encontradas en el data warehouse son muchas. Muchas de las reglas comerciales (y sus correspondientes relaciones de datos) se representan en el data warehouse, entre dos o más tablas.

1.4.2. INTEGRACIÓN

El aspecto más importante del ambiente data warehousing es que la información encontrada al interior está siempre integrada.

La integración de datos se muestra de muchas maneras: en convenciones de nombres consistentes, en la medida uniforme de variables, en la codificación de estructuras consistentes, en atributos físicos de los datos consistentes, fuentes múltiples y otros.

El contraste de la integración encontrada en el data warehouse con la carencia de integración del ambiente de aplicaciones, se muestran en la Figura 3, con diferencias bien marcadas.

A través de los años, los diseñadores de las diferentes aplicaciones han tomado sus propias decisiones sobre cómo se debería construir una aplicación. Los estilos y diseños personalizados se muestran de muchas maneras.

Se diferencian en la codificación, en las estructuras claves, en sus características físicas, en las convenciones de nombramiento y otros. La capacidad colectiva de muchos de los diseñadores de aplicaciones, para crear aplicaciones inconsistentes, es fabulosa. La Figura 3 mencionada, muestra algunas de las diferencias más importantes en las formas en que se diseñan las aplicaciones.

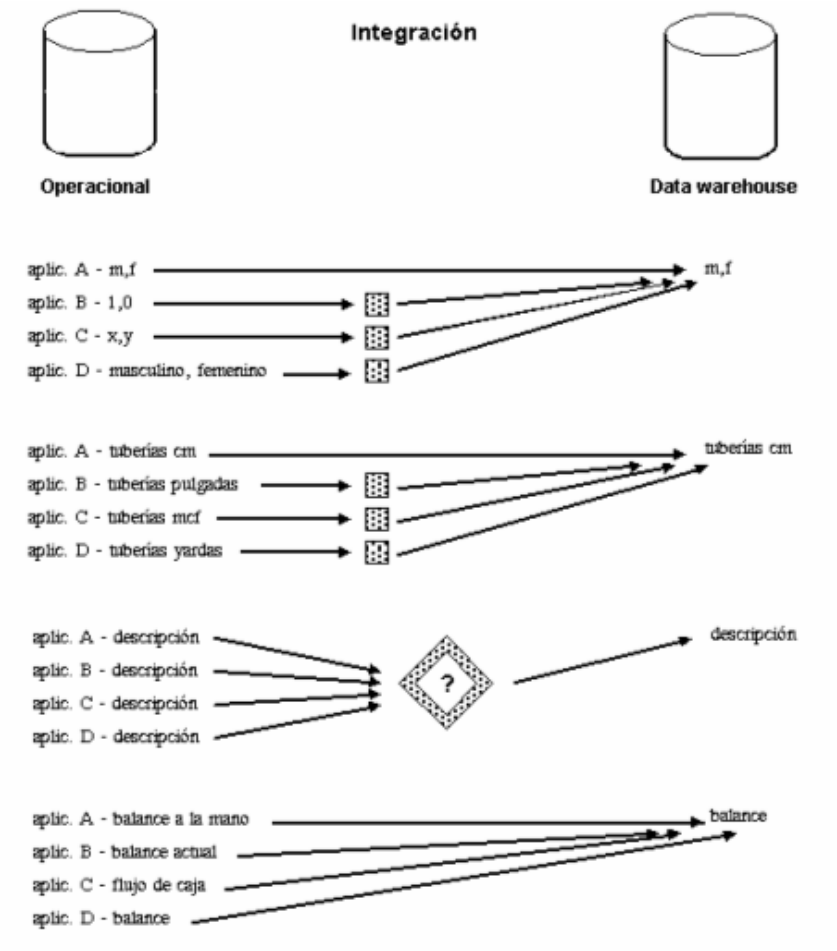

Figura 3. Integración de los datos en un data warehouse

Codificación: Los diseñadores de aplicaciones codifican el campo GENERO en varias formas. Un diseñador representa GENERO como una "M" y una "F", otros como un "1" y un "0", otros como una "X" y una "Y" e inclusive, como "masculino" y "femenino".

No importa mucho cómo el GENERO llega al data warehouse. Probablemente "M" y "F" sean tan buenas como cualquier otra representación. Lo importante es que sea de cualquier fuente de donde venga, el GENERO debe llegar al data warehouse en un estado integrado uniforme.

Por lo tanto, cuando el GENERO se carga en el data warehouse desde una aplicación, donde ha sido representado en formato "M" y "F", los datos deben convertirse al formato del data warehouse.

Medida de atributos: Los diseñadores de aplicaciones miden las unidades de medida de las tuberías en una variedad de formas. Un diseñador almacena los datos de tuberías en centímetros, otros en pulgadas, otros en millones de pies cúbicos por segundo y otros en yardas.

Al dar medidas a los atributos, la transformación traduce las diversas unidades de medida usadas en las diferentes bases de datos para transformarlas en una medida estándar común.

Cualquiera que sea la fuente, cuando la información de la tubería llegue al data warehouse necesitará ser medida de la misma manera.

Convenciones de Nombramiento: El mismo elemento es frecuentemente referido por nombres diferentes en las diversas aplicaciones. El proceso de transformación asegura que se use preferentemente el nombre de usuario.

Fuentes Múltiples: El mismo elemento puede derivarse desde fuentes múltiples. En este caso, el proceso de transformación debe asegurar que la fuente apropiada sea usada, documentada y movida al depósito.

Tal como se muestra en la figura, los puntos de integración afectan casi todos los aspectos de diseño - las características físicas de los datos, la disyuntiva de tener más de una de fuente de datos, el problema de estándares de denominación inconsistentes, formatos de fecha inconsistentes y otros.

Cualquiera que sea la forma del diseño, el resultado es el mismo - la información necesita ser almacenada en el data warehouse en un modelo globalmente aceptable y singular, aun cuando los sistemas operacionales subyacentes almacenen los datos de manera diferente.

Cuando el analista de sistema de soporte de decisiones observe el data warehouse, su enfoque deberá estar en el uso de los datos que se encuentre en el depósito, antes que preguntarse sobre la confiabilidad o consistencia de los datos.

1.4.3. DE TIEMPO VARIANTE

Toda la información del data warehouse es requerida en algún momento. Esta característica básica de los datos en un depósito, es muy diferente de la información encontrada en el ambiente operacional. En éstos, la información se requiere al momento de acceder. En otras palabras, en el ambiente operacional, cuando usted accesa a una unidad de información, usted espera que los valores requeridos se obtengan a partir del momento de acceso.

Como la información en el data warehouse es solicitada en cualquier momento (es decir, no "ahora mismo"), los datos encontrados en el depósito se llaman de "tiempo variante".

Los datos históricos son de poco uso en el procesamiento operacional. La información del depósito por el contraste, debe incluir los datos históricos para usarse en la identificación y evaluación de tendencias. (Ver Figura 4).

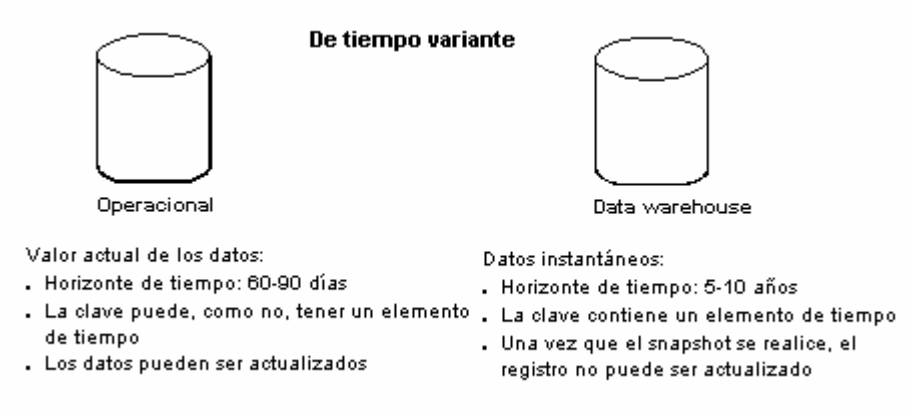

Figura 4. Tiempo variante en el data warehouse

El tiempo variante se muestra de varias maneras:

1. La más simple es que la información representa los datos sobre un horizonte largo de tiempo - desde cinco a diez años. El horizonte de tiempo representado para el ambiente operacional es mucho más corto - desde valores actuales hasta sesenta a noventa días.

Las aplicaciones que tienen un buen rendimiento y están disponibles para el procesamiento de transacciones, deben llevar una cantidad mínima de datos si tienen cualquier grado de flexibilidad. Por ello, las aplicaciones operacionales tienen un corto horizonte de tiempo, debido al diseño de aplicaciones rígidas.

2. La segunda manera en la que se muestra el tiempo variante en el data warehouse está en la estructura clave. Cada estructura clave en el data warehouse contiene, implícita o explícitamente, un elemento de tiempo como día, semana, mes, etc.

   El elemento de tiempo está casi siempre al pie de la clave concatenada, encontrada en el data warehouse. En ocasiones, el elemento de tiempo existirá implícitamente, como el caso en que un archivo completo se duplica al final del mes, o al cuarto.

3. La tercera manera en que aparece el tiempo variante es cuando la información del data warehouse, una vez registrada correctamente, no puede ser actualizada. La información del data warehouse es, para todos los propósitos prácticos, una serie larga de "snapshots" (vistas instantáneas).

   Por supuesto, si los snapshots de los datos se han tomado incorrectamente, entonces pueden ser cambiados. Asumiendo que los snapshots se han tomado adecuadamente, ellos no son alterados una vez hechos. En algunos casos puede ser no ético, e incluso ilegal, alterar los snapshots en el data warehouse. Los datos operacionales, siendo requeridos a partir del momento de acceso, pueden actualizarse de acuerdo a la necesidad.

1.4.4. NO VOLÁTIL

La información es útil sólo cuando es estable. Los datos operacionales cambian sobre una base momento a momento. La perspectiva más grande, esencial para el análisis y la toma de decisiones, requiere una base de datos estable.

En la Figura 5 se muestra que la actualización (insertar, borrar y modificar), se hace regularmente en el ambiente operacional sobre una base de registro por registro. Pero la manipulación básica de los datos que ocurre en el data warehouse es mucho más simple. Hay dos únicos tipos de operaciones: la carga inicial de datos y el acceso a los mismos. No hay actualización de datos (en el sentido general de actualización) en el depósito, como una parte normal de procesamiento.

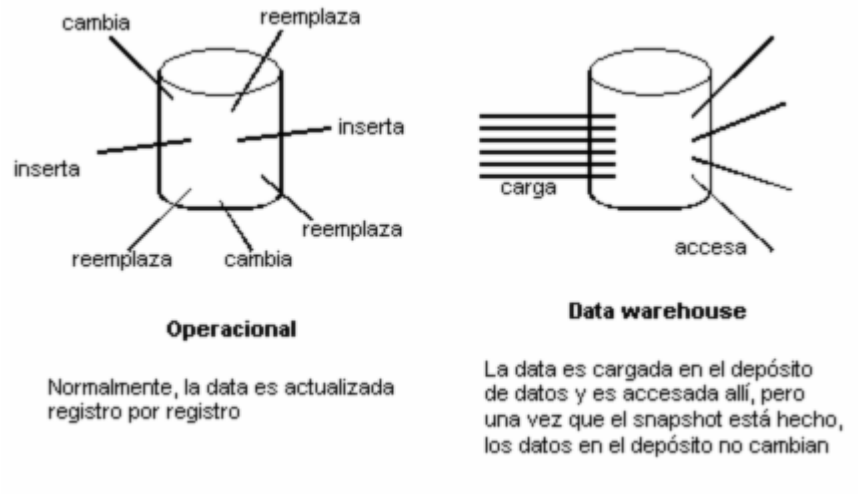

Figura 5. Data no volátil en un data warehouse

Hay algunas consecuencias muy importantes de esta diferencia básica, entre el procesamiento operacional y del data warehouse. En el nivel de diseño, la necesidad de ser precavido para actualizar las anomalías no es un factor en el data warehouse, ya que no se hace la actualización de datos. Esto significa que en el nivel físico de diseño, se pueden tomar libertades para optimizar el acceso a los datos, particularmente al usar la normalización y denormalización física.

Otra consecuencia de la simplicidad de la operación del data warehouse está en la tecnología subyacente, utilizada para correr los datos en el depósito. Teniendo que soportar la actualización de registro por registro en modo on-line (como es frecuente en el caso del procesamiento operacional) requiere que la tecnología tenga un fundamento muy complejo debajo de una fachada de simplicidad.

La tecnología permite realizar backup y recuperación, transacciones e integridad de los datos y la detección y solución al estancamiento que es más complejo. En el data warehouse no es necesario el procesamiento.

La fuente de casi toda la información del data warehouse es el ambiente operacional. A simple vista, se puede pensar que hay redundancia masiva de datos entre los dos ambientes. Desde luego, la primera impresión de muchas personas se centra en la gran redundancia de datos, entre el ambiente operacional y el ambiente de data warehouse. Dicho razonamiento es superficial y demuestra una carencia de entendimiento con respecto a qué ocurre en el data warehouse. De hecho, hay una mínima redundancia de datos entre ambos ambientes.

Se debe considerar lo siguiente:

- Los datos se filtran cuando pasan desde el ambiente operacional al de depósito. Existe mucha data que nunca sale del ambiente operacional. Sólo los datos que realmente se necesitan ingresarán al ambiente de data warehouse.
- El horizonte de tiempo de los datos es muy diferente de un ambiente al otro. La información en el ambiente operacional es más reciente con respecto a la del data warehouse. Desde la perspectiva de los horizontes de tiempo únicos, hay poca superposición entre los ambientes operacional y de data warehouse.
- El data warehouse contiene un resumen de la información que no se encuentra en el ambiente operacional.

Los datos experimentan una transformación fundamental cuando pasa al data warehouse. La mayor parte de los datos se alteran significativamente al ser seleccionados y movidos al data warehouse. Dicho de otra manera, la mayoría de los datos se alteran física y radicalmente cuando se mueven al depósito. No es la misma data que reside en el ambiente operacional desde el punto de vista de integración.

En vista de estos factores, la redundancia de datos entre los dos ambientes es una ocurrencia rara, que resulta en menos de 1%.

1.5. ESTRUCTURA DEL DATA WAREHOUSE

Los data warehouses tienen una estructura distinta. Hay niveles diferentes de esquematización y detalle que delimitan el data warehouse. La estructura de un data warehouse se muestra en la Figura 6.

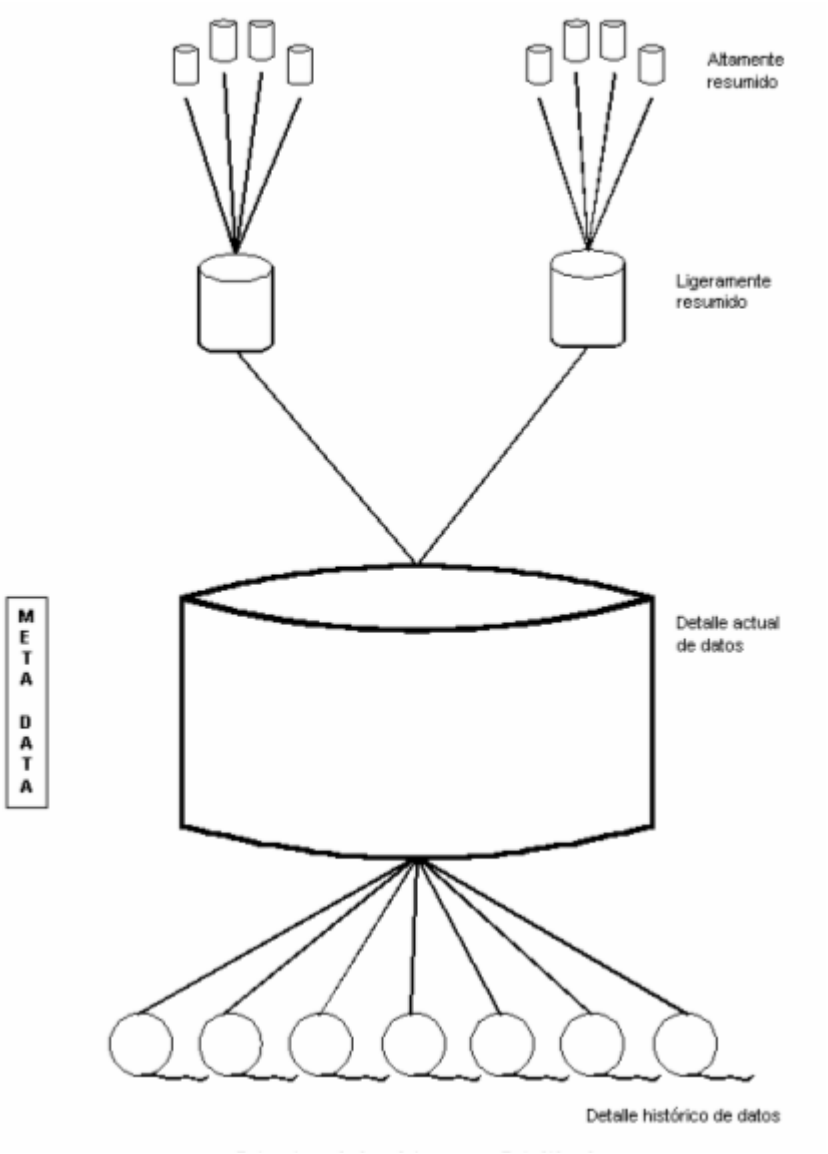

Figura 6. Estructura de datos en un data warehouse

En la figura, se muestran los diferentes componentes del data warehouse y son:

- Detalle de datos actuales
- Detalle de datos antiguos
- Datos ligeramente resumidos
- Datos completamente resumidos
- Meta data

Detalle de datos actuales: En gran parte, el interés más importante radica en el detalle de los datos actuales, debido a que:

- Refleja las ocurrencias más recientes, las cuales son de gran interés
- Es voluminoso, ya que se almacena al más bajo nivel de granularidad.
- Casi siempre se almacena en disco, el cual es de fácil acceso, aunque su administración sea costosa y compleja.

Detalle de datos antiguos: La data antigua es aquella que se almacena sobre alguna forma de almacenamiento masivo. No es frecuentemente accesada y se almacena a un nivel de detalle, consistente con los datos detallados actuales. Mientras no sea prioritario el almacenamiento en un medio de almacenaje alterno, a causa del gran volumen de datos unido al acceso no frecuente de los mismos, es poco usual utilizar el disco como medio de almacenamiento.

Datos ligeramente resumidos: La data ligeramente resumida es aquella que proviene desde un bajo nivel de detalle encontrado al nivel de detalle actual. Este nivel del data warehouse casi siempre se almacena en disco. Los puntos en los que se basa el diseñador para construirlo son: Que la unidad de tiempo se encuentre sobre la esquematización hecha, Qué contenidos (atributos) tendrá la data ligeramente resumida.

Datos completamente resumidos: El siguiente nivel de datos encontrado en el data warehouse es el de los datos completamente resumidos. Estos datos son compactos y fácilmente accesibles.

A veces se encuentra en el ambiente de data warehouse y en otros, fuera del límite de la tecnología que ampara al data warehouse. (De todos modos, los datos completamente resumidos son parte del data warehouse sin considerar donde se alojan los datos físicamente.)

Metadata: El componente final del data warehouse es el de la metadata. De muchas maneras la metadata se sitúa en una dimensión diferente al de otros datos del data warehouse, debido a que su contenido no es tomado directamente desde el ambiente operacional.

La metadata juega un rol especial y muy importante en el data warehouse y es usada como:

- Un directorio para ayudar al analista a ubicar los contenidos del data warehouse.
- Una guía para el mapping de datos de cómo se transforma, del ambiente operacional al de data warehouse.
- Una guía de los algoritmos usados para la esquematización entre el detalle de datos actual, con los datos ligeramente resumidos y éstos, con los datos completamente resumidos, etc.

La metadata contiene (al menos):

- La estructura de los datos
- Los algoritmos usados para la esquematización
- El mapping desde el ambiente operacional al data warehouse

La información adicional que no se esquematiza es almacenada en el data warehouse. En muchas ocasiones, allí se hará el análisis y se producirá un tipo u otro de resumen. El único tipo de esquematización que se almacena permanentemente en el data warehouse, es el de los datos que son usados frecuentemente. En otras palabras, si un analista produce un resumen que tiene una probabilidad muy baja de ser usado nuevamente, entonces la esquematización no es almacenada en el data warehouse.

1.6. ARQUITECTURA DE UN DATA WAREHOUSE

Una de las razones por las que el desarrollo de un data warehouse crece rápidamente, es que realmente es una tecnología muy entendible. De hecho, data warehousing puede representar mejor la estructura amplia de una empresa para administrar los datos informacionales dentro de la organización. A fin de comprender cómo se relacionan todos los componentes involucrados en una estrategia data warehousing, es esencial tener una Arquitectura Data Warehouse.

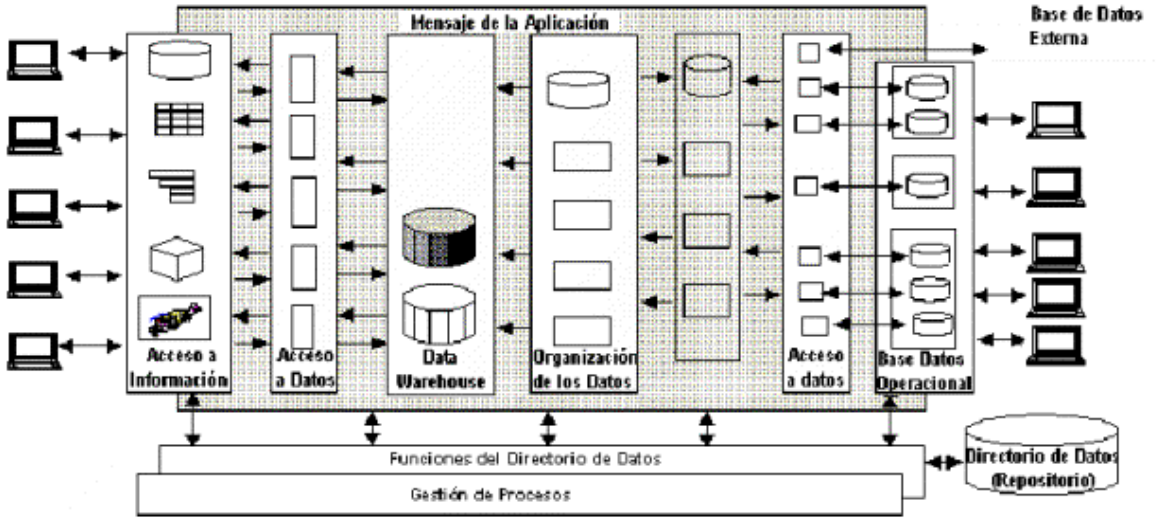

Figura 7. Arquitectura de un data warehouse

1.6.1. ELEMENTOS CONSTITUYENTES DE UNA ARQUITECTURA DATA WAREHOUSE

Una Arquitectura Data Warehouse (Data Warehouse Architecture - DWA) es una forma de representar la estructura total de datos, comunicación, procesamiento y presentación, que existe para los usuarios finales que disponen de una computadora dentro de la empresa.

La arquitectura se constituye de un número de partes interconectadas:

- Base de datos operacional / Nivel de base de datos externo
- Nivel de acceso a la información
- Nivel de acceso a los datos
- Nivel de directorio de datos (Metadata)
- Nivel de gestión de proceso
- Nivel de mensaje de la aplicación
- Nivel de data warehouse
- Nivel de organización de datos

1.6.1.1. Base de datos operacional / Nivel de base de datos externo

Los sistemas operacionales procesan datos para apoyar las necesidades operacionales críticas. Para hacer eso, se han creado las bases de datos operacionales históricas que proveen una estructura de procesamiento eficiente, para un número relativamente pequeño de transacciones comerciales bien definidas.

Sin embargo, a causa del enfoque limitado de los sistemas operacionales, las bases de datos diseñadas para soportar estos sistemas, tienen dificultad al acceder a los datos para otra gestión o propósitos informáticos.

Esta dificultad en acceder a los datos operacionales es amplificada por el hecho que muchos de estos sistemas tienen de 10 a 15 años de antigüedad. El tiempo de algunos de estos sistemas significa que la tecnología de acceso a los datos disponible para obtener los datos operacionales, es así mismo antigua.

Ciertamente, la meta del data warehousing es liberar la información que es almacenada en bases de datos operacionales y combinarla con la información desde otra fuente de datos, generalmente externa.

Cada vez más, las organizaciones grandes adquieren datos adicionales desde bases de datos externas. Esta información incluye tendencias demográficas, econométricas, adquisitivas y

competitivas (que pueden ser proporcionadas por Instituciones Oficiales - INEI). Internet o también llamada "information superhighway" (supercarretera de la información) provee el acceso a más recursos de datos todos los días.

1.6.1.2. Nivel de acceso a la información

El nivel de acceso a la información de la arquitectura data warehouse, es el nivel del que el usuario final se encarga directamente. En particular, representa las herramientas que el usuario final normalmente usa día a día. Por ejemplo: Excel, Lotus 1-2-3, Focus, Access, SAS, etc.

Este nivel también incluye el hardware y software involucrados en mostrar información en pantalla y emitir reportes de impresión, hojas de cálculo, gráficos y diagramas para el análisis y presentación. Hace dos décadas que el nivel de acceso a la información se ha expandido enormemente, especialmente a los usuarios finales quienes se han volcado a los PCs monousuarios y los PCs en redes.

Actualmente, existen herramientas más y más sofisticadas para manipular, analizar y presentar los datos, sin embargo, hay problemas significativos al tratar de convertir los datos tal como han sido recolectados y que se encuentran contenidos en los sistemas operacionales en información fácil y transparente para las herramientas de los usuarios finales. Una de las claves para esto es encontrar un lenguaje de datos común que puede usarse a través de toda la empresa.

1.6.1.3. Nivel de acceso a los datos

El nivel de acceso a los datos de la arquitectura data warehouse está involucrado con el nivel de acceso a la información para conversar en el nivel operacional. En la red mundial de hoy, el lenguaje de datos común que ha surgido es SQL. Originalmente, SQL fue desarrollado por IBM como un lenguaje de consulta, pero en los últimos veinte años ha llegado a ser el estándar para el intercambio de datos.

Uno de los adelantos claves de los últimos años ha sido el desarrollo de una serie de "filtros" de acceso a datos, tales como EDA/SQL para acceder a casi todo los Sistemas de Gestión de Base de Datos (Data Base Management Systems - DBMSs) y sistemas de archivos de datos, relacionales o no. Estos filtros permiten a las herramientas de acceso a la información, acceder también a la data almacenada en sistemas de gestión de base de datos que tienen veinte años de antigüedad.

El nivel de acceso a los datos no solamente conecta DBMSs diferentes y sistemas de archivos sobre el mismo hardware, sino también a los fabricantes y protocolos de red. Una de las claves de una estrategia data warehousing es proveer a los usuarios finales con "acceso a datos universales".

El acceso a los datos universales significa que, teóricamente por lo menos, los usuarios finales sin tener en cuenta la herramienta de acceso a la información o ubicación, deberían ser capaces de acceder a cualquier o todos los datos en la empresa que es necesaria para ellos, para hacer su trabajo.

El nivel de acceso a los datos entonces es responsable de la interfaces entre las herramientas de acceso a la información y las bases de datos operacionales. En algunos casos, esto es todo lo que un usuario final necesita. Sin embargo, en general, las organizaciones desarrollan un plan mucho más sofisticado para el soporte del data warehousing.

1.6.1.4. Nivel de Directorio de Datos (Metadata)

A fin de proveer el acceso a los datos universales, es absolutamente necesario mantener alguna forma de directorio de datos o repositorio de la información metadata. La metadata es la información alrededor de los datos dentro de la empresa.

Las descripciones de registro en un programa COBOL son metadata. También lo son las sentencias DIMENSION en un programa FORTRAN o las sentencias a crear en SQL.

A fin de tener un depósito totalmente funcional, es necesario tener una variedad de metadata disponibles, información sobre las vistas de datos de los usuarios finales e información sobre las bases de datos operacionales. Idealmente, los usuarios finales deberían de acceder a los datos desde el data warehouse (o desde las bases de datos operacionales), sin tener que conocer dónde residen los datos o la forma en que se han almacenados.

1.6.1.5. Nivel de Gestión de Procesos

El nivel de gestión de procesos tiene que ver con la programación de diversas tareas que deben realizarse para construir y mantener el data warehouse y la información del directorio de datos. Este nivel puede depender del alto nivel de control de trabajo para muchos procesos (procedimientos) que deben ocurrir para mantener el data warehouse actualizado.

1.6.1.6. Nivel de Mensaje de la Aplicación

El nivel de mensaje de la aplicación tiene que ver con el transporte de información alrededor de la red de la empresa. El mensaje de aplicación se refiere también como "subproducto", pero puede involucrar sólo protocolos de red. Puede usarse por ejemplo, para aislar aplicaciones operacionales o estratégicas a partir del formato de datos exacto, recolectar transacciones o los mensajes y entregarlos a una ubicación segura en un tiempo seguro.

1.6.1.7. Nivel Data Warehouse (Físico)

En el data warehouse (núcleo) es donde ocurre la data actual, usada principalmente para usos estratégicos. En algunos casos, uno puede pensar del data warehouse simplemente como una vista lógica o virtual de datos. En muchos ejemplos, el data warehouse puede no involucrar almacenamiento de datos.

En un data warehouse físico, copias, en algunos casos, muchas copias de datos operacionales y/o externos, son almacenados realmente en una forma que es fácil de acceder y es altamente flexible. Cada vez más, los data warehouses son almacenados sobre plataformas cliente/servidor, pero por lo general se almacenan sobre mainframes.

1.6.1.8. Nivel de Organización de Datos

El componente final de la arquitectura data warehouse es la organización de los datos. Se llama también gestión de copia o réplica, pero de hecho, incluye todos los procesos necesarios como seleccionar, editar, resumir, combinar y cargar datos en el depósito y acceder a la información desde bases de datos operacionales y/o externas.

La organización de datos involucra con frecuencia una programación compleja, pero cada vez más, están creándose las herramientas data warehousing para ayudar en este proceso. Involucra también programas de análisis de calidad de datos y filtros que identifican modelos y estructura de datos dentro de la data operacional existente.

1.6.2. OPERACIONES EN UN DATA WAREHOUSE

En la Figura 8 se muestra algunos de los tipos de operaciones que se efectúan dentro de un ambiente data warehousing.

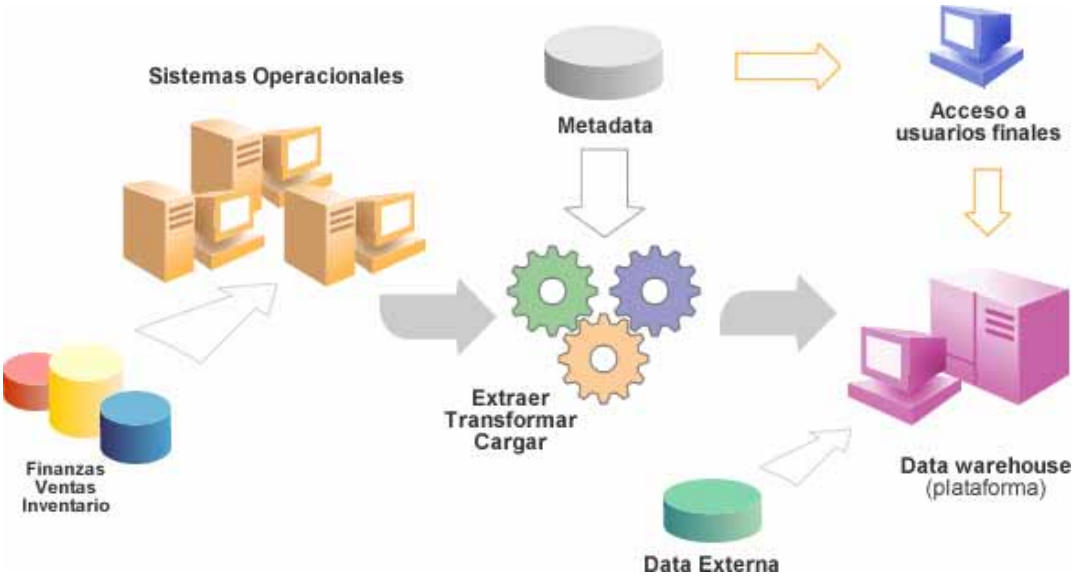

Figura 8. Operaciones en un data warehouse

1.6.2.1. Sistemas Operacionales

Los datos administrados por los sistemas de aplicación operacionales son la fuente principal de datos para el data warehouse.

Las bases de datos operacionales se organizan como archivos indexados (UFAS, VSAM), bases de datos de redes/jerárquicas (I-D-S/II, IMS, IDMS) o sistemas de base de datos relacionales (DB2, Oracle, Informix, etc.). Según las encuestas, aproximadamente del 70% a 80% de las bases de datos de las empresas se organizan usando DBMSs no relacional.

1.6.2.2. Extracción, Transformación y Carga de los Datos

Se requieren herramientas de gestión de datos para extraer datos desde bases de datos y/o archivos operacionales, luego es necesario manipular o transformar los datos antes de cargar los resultados en el data warehouse.

Tomar los datos desde varias bases de datos operacionales y transformarlos en datos requeridos para el depósito, se refiere a la transformación o a la integración de datos. Las bases de datos operacionales, diseñadas para el soporte de varias aplicaciones de producción, frecuentemente difieren en el formato.

Los mismos elementos de datos, si son usados por aplicaciones diferentes o administrados por diferentes software DBMS, pueden definirse al usar nombres de elementos inconsistentes, que tienen formatos inconsistentes y/o ser codificados de manera diferente. Todas estas inconsistencias deben resolverse antes que los elementos de datos sean almacenados en el data warehouse.

1.6.2.3. Metadata

Otro paso necesario es crear la metadata. La metadata (es decir, datos acerca de datos) describe los contenidos del data warehouse. La metadata consiste de definiciones de los elementos de datos en el depósito, sistema(s) del (os) elemento(s) fuente. Como la data, se integra y transforma antes de ser almacenada en información similar.

1.6.2.4. Acceso de usuario final

Los usuarios accesan al data warehouse por medio de herramientas de productividad basadas en GUI (Graphical User Interface - Interfase gráfica de usuario). Pueden proveerse a los usuarios del data warehouse muchos de estos tipos de herramientas.

Estos pueden incluir software de consultas, generadores de reportes, procesamiento analítico en línea, herramientas data/visual mining, etc., dependiendo de los tipos de usuarios y sus requerimientos particulares. Sin embargo, una sola herramienta no satisface todos los requerimientos, por lo que es necesaria la integración de una serie de herramientas.

1.6.2.5. Plataforma del data warehouse

La plataforma para el data warehouse es casi siempre un servidor de base de datos relacional. Cuando se manipulan volúmenes muy grandes de datos puede requerirse una configuración en bloque de servidores UNIX con multiprocesador simétrico (SMP) o un servidor con procesador paralelo masivo (MPP) especializado.

Los extractos de la data integrada/transformada se cargan en el data warehouse. Uno de los más populares RDBMSs disponibles para data warehousing sobre la plataforma UNIX (SMP y MPP) generalmente es Teradata. La elección de la plataforma es crítica. El depósito crecerá y hay que comprender los requerimientos después de 3 o 5 años.

Muchas de las organizaciones quieran o no escogen una plataforma por diversas razones: el Sistema X es nuestro sistema elegido o el Sistema Y está ya disponible sobre un sistema UNIX que nosotros ya tenemos. Uno de los errores más grandes que las organizaciones cometen al seleccionar la plataforma, es que ellos presumen que el sistema (hardware y/o DBMS) escalará con los datos.

El sistema de depósito ejecuta las consultas que se pasa a los datos por el software de acceso a los datos del usuario. Aunque un usuario visualiza las consultas desde el punto de vista de un GUI, las consultas típicamente se formulan como pedidos SQL, porque SQL es un lenguaje universal y el estándar de hecho para el acceso a datos.

1.6.2.6. Datos Externos

Dependiendo de la aplicación, el alcance del data warehouse puede extenderse por la capacidad de acceder a la data externa. Por ejemplo, los datos accesibles por medio de servicios de computadora en línea (tales como CompuServe y America On Line) y/o vía Internet, pueden estar disponibles a los usuarios del data warehouse.

Evolución del Depósito

Construir un data warehouse es una tarea grande. No es recomendable emprender el desarrollo del data warehouse de la empresa como un proyecto cualquiera. Más bien, se recomienda que los requerimientos de una serie de fases se desarrollen e implementen en modelos consecutivos que permitan un proceso de implementación más gradual e iterativo.

No existe ninguna organización que haya triunfado en el desarrollo del data warehouse de la empresa, en un sólo paso. Muchas, sin embargo, lo han logrado luego de un desarrollo paso a paso. Los pasos previos evolucionan conjuntamente con la materia que está siendo agregada.

Los datos en el data warehouse no son volátiles y es un repositorio de datos de sólo lectura (en general). Sin embargo, pueden añadirse nuevos elementos sobre una base regular para que el contenido siga la evolución de los datos en la base de datos fuente, tanto en los contenidos como en el tiempo.

Uno de los desafíos de mantener un data warehouse, es idear métodos para identificar datos nuevos o modificados en las bases de datos operacionales. Algunas maneras para identificar estos datos incluyen insertar fecha/tiempo en los registros de base de datos y entonces crear copias de registros actualizados y copiar información de los registros de transacción y/o base de datos diarias.

Estos elementos de datos nuevos y/o modificados son extraídos, integrados, transformados y agregados al data warehouse en pasos periódicos programados. Como se añaden las nuevas ocurrencias de datos, los datos antiguos son eliminados. Por ejemplo, si los detalles de un sujeto particular se mantienen por 5 años, como se agregó la última semana, la semana anterior es eliminada.

1.7. TRANSFORMACIÓN DE DATOS Y METADATA

1.7.1. TRANSFORMACIÓN DE DATOS

Uno de los desafíos de cualquier implementación de data warehouse, es el problema de transformar los datos. La transformación se encarga de las inconsistencias en los formatos de datos y la codificación, que pueden existir dentro de una base de datos única y que casi siempre existen cuando múltiples bases de datos contribuyen al data warehouse.

En la Figura 9 se ilustra una forma de inconsistencia, en la cual el género se codifica de manera diferente en tres bases de datos diferentes. Los procesos de transformación de datos se desarrollan para direccionar estas inconsistencias.

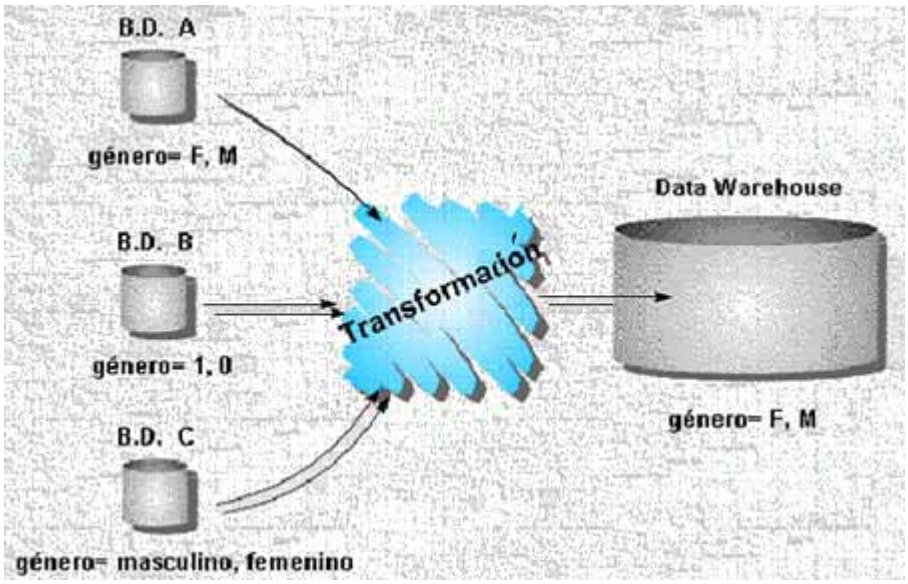

Figura 9. Transformación de datos

La transformación de datos también se encarga de las inconsistencias en el contenido de datos. Una vez que se toma la decisión sobre que reglas de transformación serán establecidas, deben crearse e incluirse las definiciones en las rutinas de transformación.

Se requiere una planificación cuidadosa y detallada para transformar datos inconsistentes en conjuntos de datos conciliables y consistentes para cargarlos en el data warehouse.

1.7.2. METADATA

Otro aspecto de la arquitectura de data warehouse es crear soporte a la metadata. Metadata es la información sobre los datos que se alimenta, se transforma y existe en el data warehouse. Metadata es un concepto genérico, pero cada implementación de la metadata usa técnicas y métodos específicos.

Estos métodos y técnicas son dependientes de los requerimientos de cada organización, de las capacidades existentes y de los requerimientos de interfaces de usuario. Hasta ahora, no hay normas para la metadata, por lo que la metadata debe definirse desde el punto de vista del software data warehousing, seleccionado para una implementación específica.

Típicamente, la metadata incluye los siguientes ítems:

- Las estructuras de datos que dan una visión de los datos al administrador de datos.
- Las definiciones del sistema de registro desde el cual se construye el data warehouse.
- Las especificaciones de transformaciones de datos que ocurren tal como la fuente de datos se replica al data warehouse.
- El modelo de datos del data warehouse (es decir, los elementos de datos y sus relaciones).
- Un registro de cuando los nuevos elementos de datos se agregan al data warehouse y cuando los elementos de datos antiguos se eliminan o se resumen.
- Los niveles de sumarización, el método de sumarización y las tablas de registros de su data warehouse.

Algunas implementaciones de la metadata también incluyen definiciones de la(s) vista(s) presentada(s) a los usuarios del data warehouse. Típicamente, se definen vistas múltiples para favorecer las preferencias variadas de diversos grupos de usuarios. En otras implementaciones, estas descripciones se almacenan en un Catálogo de Información.

Los esquemas y subesquemas para bases de datos operacionales, forman una fuente óptima de entrada cuando se crea la metadata. Hacer uso de la documentación existente, especialmente cuando está disponible en forma electrónica, puede acelerar el proceso de definición de la metadata del ambiente data warehousing.

La metadata sirve, en un sentido, como el corazón del ambiente data warehousing. Crear definiciones de metadata completa y efectiva puede ser un proceso que consuma tiempo, pero lo mejor de las definiciones y si usted usa herramientas de gestión de software integrado, son los esfuerzos que darán como resultado el mantenimiento del data warehouse.

1.8. FLUJO DE DATOS

Existe un flujo de datos normal y predecible dentro del data warehouse. La Figura 10 muestra ese flujo.

Los datos ingresan al data warehouse desde el ambiente operacional. (Hay pocas excepciones a esta regla).

Al ingresar al data warehouse, la información va al nivel de detalle actual, tal como se muestra. Se queda allí y se usa hasta que ocurra uno de los tres eventos siguientes:

- Sea eliminado
- Sea resumido
- Sea archivado

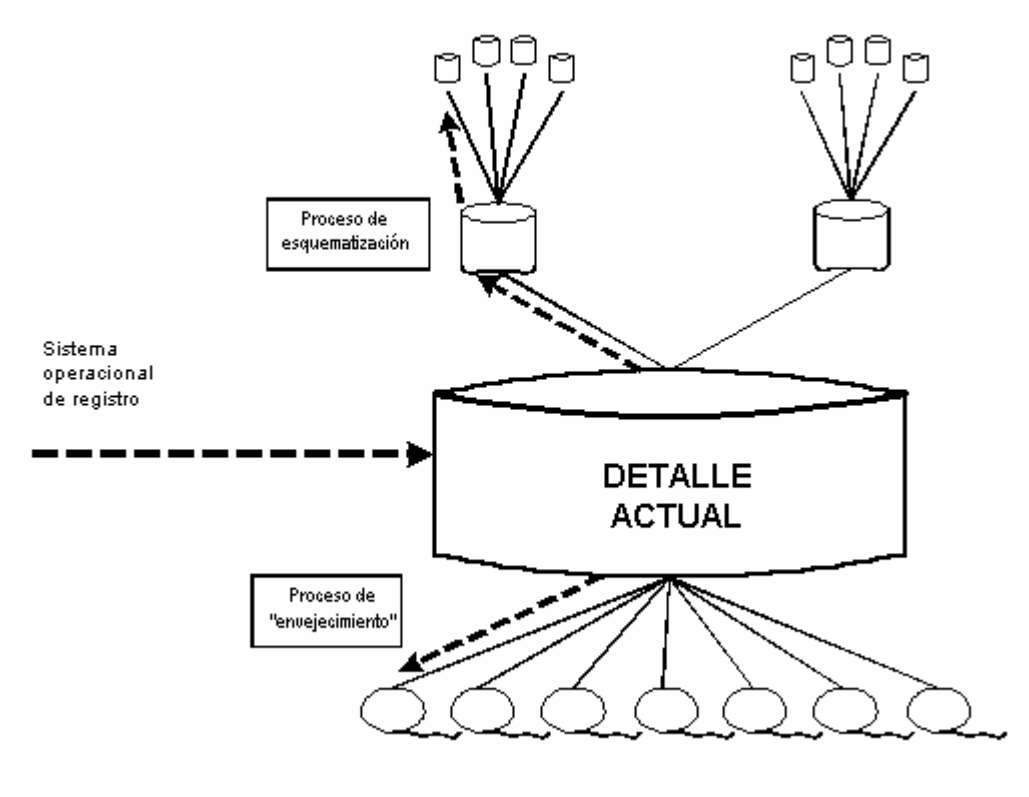

Figura 10. Flujo de datos en el data warehouse

Con el proceso de desactualización en un data warehouse se mueve el detalle de la data actual a data antigua, basado en el tiempo de los datos. El proceso de esquematización usa el detalle de los datos para calcular los datos en forma ligera y completamente resumidos.

Hay pocas excepciones al flujo mostrado. Sin embargo, en general, para la mayoría de datos encontrados en un data warehouse, el flujo de la información es como se ha explicado.

1.9. MEDIOS DE ALMACENAMIENTO PARA INFORMACIÓN ANTIGUA

El símbolo mostrado en la Figura 11 para medios de almacenamiento de información antigua es la cinta magnética, que puede usarse para almacenar este tipo de información. De hecho hay una amplia variedad de medios de almacenamiento que deben considerarse para almacenar datos más antiguos. En la figura se muestra algunos de esos medios.

Dependiendo del volumen de información, la frecuencia de acceso, el costo de los medios y el tipo de acceso, es probable que otros medios de almacenamiento sirvan a las necesidades del nivel de detalle más antiguo en el data warehouse.

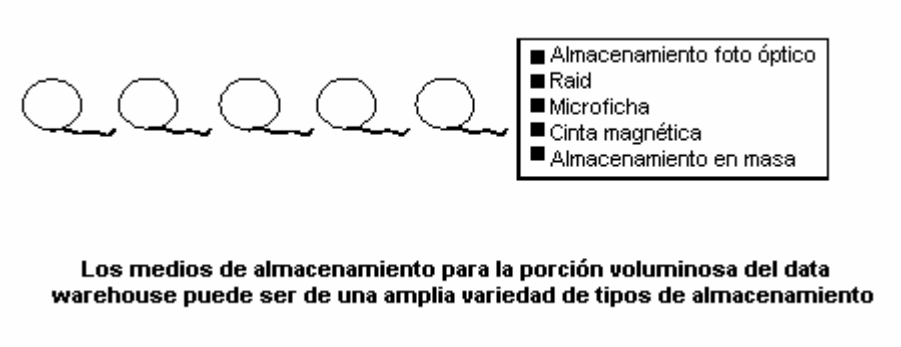

Figura 11. Medios de almacenamiento

1.10. USOS DEL DATA WAREHOUSE

Los datos operacionales y los datos del data warehouse son accesados por usuarios que usan los datos de maneras diferentes.

Uso de Base de Datos Operacional	Uso Data Warehouse
Muchos usuarios concurrentes	Pocos usuarios concurrentes
Consultas predefinidas y actualizables	Consultas complejas, frecuentemente no anticipadas
Cantidades pequeñas de datos detallados	Cantidades grandes de datos detallados
Requerimientos de respuesta inmediata	Requerimientos de respuesta no críticos

Tabla 3. Usos de un Data warehouse

Los usuarios de un data warehouse necesitan acceder a los datos complejos, frecuentemente desde fuentes múltiples y de formas no predecibles.

Los usuarios que accedan a los datos operacionales, comúnmente efectúan tareas predefinidas que, generalmente requieren acceso a una sola base de datos de una aplicación. Por el contrario, los usuarios que accedan al data warehouse, efectúan tareas que requieren acceso a un conjunto de datos desde fuentes múltiples y frecuentemente no son predecibles. Lo único que se conoce (si es modelada correctamente) es el conjunto inicial de datos que se han establecido en el depósito.

Por ejemplo, un especialista en el cuidado de la salud podría necesitar acceder a los datos actuales e históricos para analizar las tendencias de costos, usando un conjunto de consultas predefinidas. Por el contrario, un representante de ventas podría necesitar acceder a los datos de cliente y producto para evaluar la eficacia de una campaña de marketing, creando consultas base o ad-hoc para encontrar nuevamente necesidades definidas.

Sólo pocos usuarios acceden a los datos concurrentemente

En contraste a la producción de sistemas que pueden manejar cientos o miles de usuarios concurrentes, al data warehouse acceda un limitado conjunto de usuarios en cualquier tiempo determinado.

Los usuarios generan un procesamiento no predecible complejo

Los usuarios del data warehouse generan consultas complejas. A veces la respuesta a una consulta conduce a la formulación de otras preguntas más detalladas, en un proceso llamado drilling down. El data warehouse puede incluir niveles de resúmenes múltiples, derivado de un conjunto principal, único, de datos detallados, para soportar este tipo de uso.

En efecto, los usuarios frecuentemente comienzan buscando en los datos resumidos y como identifican áreas de interés, comienzan a acceder al conjunto de datos detallado. Los conjuntos de datos resumidos representan el "Qué" de una situación y los conjuntos de datos detallados permiten a los usuarios construir un cuadro sobre "Cómo" se ha derivado esa situación.

Las consultas de los usuarios accedan a cantidades grandes de datos

Debido a la necesidad de investigar tendencias y evaluar las relaciones entre muchas clases de datos, las consultas al data warehouse permiten acceder a volúmenes muy grandes tanto de data detallada como resumida. Debido a los requerimientos de datos históricos, los data warehouses evolucionan para llegar a un tamaño más grande que sus orígenes operacionales (de 10 a 100 veces más grande).

Las consultas de los usuarios no tienen tiempos de respuesta críticos

Las transacciones operacionales necesitan una respuesta inmediata porque un cliente puede estar esperando una respuesta. En el data warehouse, por el contrario, tiene un requerimiento de respuesta no-crítico porque el resultado frecuentemente se usa en un proceso de análisis y toma de decisiones. Aunque los tiempos de respuesta no son críticos, los usuarios esperan una respuesta dentro del mismo día en que es hecha la consulta.

Por lo general, los diferentes niveles de datos dentro del data warehouse reciben diferentes usos. A más alto nivel de esquematización, se tiene mayor uso de los datos.

En la Figura 12 se muestra que hay mayor uso de los datos completamente resumidos, a diferencia de la información antigua que apenas es usada.

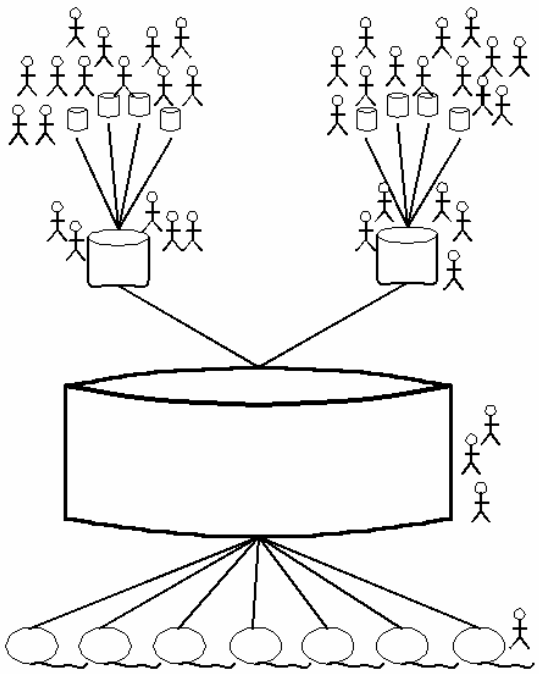

Figura 12. Uso de datos

Hay una buena razón para mover una organización al paradigma sugerido en la figura, la utilización del recurso. La data más resumida, permite capturar los datos en forma más rápida y eficiente. Si en una tarea se encuentra que se hace mucho procesamiento a niveles de detalle del data warehouse, entonces se consumirá muchos recursos de máquina. Es mejor hacer el procesamiento a niveles más altos de esquematización como sea posible.

Para muchas tareas, el analista de sistemas de soporte de decisiones usa la información a nivel de detalle en un pre data warehouse. La seguridad de la información de detalle se consigue de muchas maneras, aun cuando estén disponibles otros niveles de esquematización. Una de las

actividades del diseñador de datos es el de desconectar al usuario del sistema de soporte de decisiones del uso constante de datos a nivel de detalle más bajo.

El diseñador de datos tiene dos predisposiciones:

- Instalar un sistema chargeback, donde el usuario final pague por los recursos consumidos
- Señalar el mejor tiempo de respuesta que puede obtenerse cuando se trabaja con la data a un nivel alto de esquematización, a diferencia de un pobre tiempo de respuesta que resulta de trabajar con los datos a un nivel bajo de detalle.

1.11. CONSIDERACIONES PREVIAS PARA EL DESARROLLO DE UN DATA WAREHOUSE

Hay muchas maneras para desarrollar data warehouses como tantas organizaciones existen. Sin embargo, hay un número de dimensiones diferentes que necesitan ser consideradas:

- Alcance de un data warehouse
- Redundancia de datos
- Tipo de usuario final

La Figura 13 muestra un esquema bidimensional para analizar las opciones básicas. La dimensión horizontal indica el alcance del depósito y la vertical muestra la cantidad de datos redundantes que deben almacenarse y mantenerse.

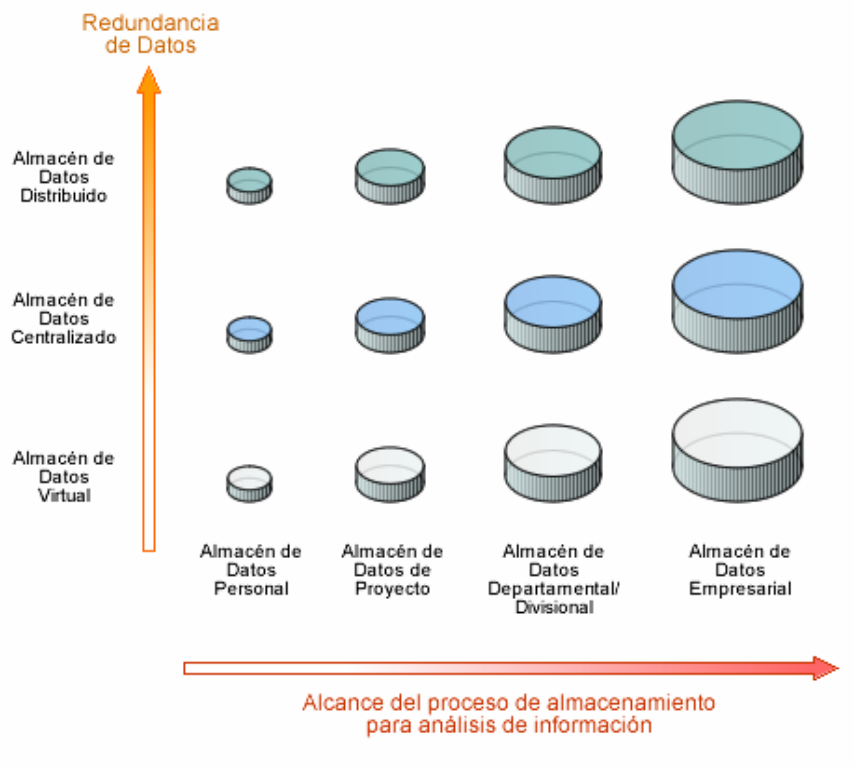

Figura 13. Alcance del data warehouse

1.11.1. ALCANCE DEL DATA WAREHOUSE

El alcance de un data warehouse puede ser tan amplio como toda la información estratégica de la empresa desde su inicio, o puede ser tan limitado como un data warehouse personal para un solo gerente durante un año.

En la práctica, en la amplitud del alcance, el mayor valor del data warehouse es para la empresa y lo más caro y consumidor de tiempo es crear y mantenerlo. Como consecuencia de ello, la mayoría de las organizaciones comienzan con data warehouses funcionales, departamentales o divisionales y luego los expanden como usuarios que proveen retroalimentación.

1.11.2. REDUNDANCIA DE DATOS

Hay tres niveles esenciales de redundancia de datos que las empresas deberían considerar en sus opciones de data warehouse:

- Data warehouses "virtual" o "Point to Point"
- Data warehouses "centrales"
- Data warehouses "distribuidos"

No se puede pensar en un único enfoque. Cada opción adapta un conjunto específico de requerimientos y una buena estrategia de almacenamiento de datos, lo constituye la inclusión de las tres opciones.

Data Warehouses "Virtual" o "Point to Point"

Una estrategia de data warehouses virtual, significa que los usuarios finales pueden acceder a bases de datos operacionales directamente, usando cualquier herramienta que posibilite "la red de acceso de datos".

Este enfoque provee flexibilidad así como también la cantidad mínima de datos redundantes que deben cargarse y mantenerse. Además, se pueden colocar las cargas de consulta no planificadas más grandes, sobre sistemas operacionales.

Como se verá, el almacenamiento virtual es, frecuentemente, una estrategia inicial, en organizaciones donde hay una amplia (pero en su mayor parte indefinida) necesidad de conseguir la data operacional, desde una clase relativamente grande de usuarios finales y donde la frecuencia probable de pedidos es baja.

Los depósitos virtuales de datos proveen un punto de partida para que las organizaciones determinen qué usuarios finales están buscando realmente.

Data Warehouses "Centrales"

El concepto de data warehouses centrales es el concepto inicial que se tiene del data warehouse. Es una única base de datos física, que contiene todos los datos para un área funcional específica, departamento, división o empresa.

Los data warehouses centrales se seleccionan por lo general donde hay una necesidad común de los datos informáticos y un número grande de usuarios finales ya conectados a una red o computadora central. Pueden contener datos para cualquier período específico de tiempo. Comúnmente, contienen datos de sistemas operacionales múltiples.

Los data warehouses centrales son reales. Los datos almacenados en el data warehouse son accesibles desde un lugar y deben cargarse y mantenerse sobre una base regular. Normalmente se construyen alrededor de RDBMs avanzados o, en alguna forma, de servidor de base de datos informático multidimensional.

Data Warehouses Distribuidos

Los data warehouses distribuidos son aquellos en los cuales ciertos componentes del depósito se distribuyen a través de un número de bases de datos físicas diferentes.

Cada vez más, las organizaciones grandes están tomando decisiones a niveles más inferiores de la organización y a la vez, llevando los datos que se necesitan para la toma de decisiones a la red de área local (Local Area Network - LAN) o computadora local que sirve al que toma decisiones.

Los data warehouses distribuidos comúnmente involucran la mayoría de los datos redundantes y como consecuencia de ello, se tienen procesos de actualización y carga más complejos.

1.11.3. TIPO DE USUARIO FINAL

De la misma forma que hay una gran cantidad de maneras para organizar un data warehouse, es importante notar que también hay una gama cada vez más amplia de usuarios finales.

En general, se puede considerar tres grandes categorías:

- Ejecutivos y gerentes
- "Power users" o "Buzo de Información" (analistas financieros y de negocios, ingenieros, etc.)
- Usuarios de soporte (de oficina, administrativos, etc.).

Cada una de estas categorías diferentes de usuario tienen su propio conjunto de requerimientos para los datos, acceso, flexibilidad y facilidad de uso.

1.12. ELEMENTOS CLAVES PARA EL DESARROLLO DE UN DATA WAREHOUSE

Los data warehouses exitosos comienzan cuando se escogen e integran satisfactoriamente tres elementos claves.

Un data warehouse está integrado por un servidor de hardware y los DBMS que conforman el depósito. Del lado del hardware, se debe combinar la configuración de plataformas de los servidores, mientras se decide cómo aprovechar los saltos casi constantes de la potencia del procesador. Del lado del software, la complejidad y el alto costo de los DBMS's fuerzan a tomar decisiones drásticas y balances comparativos inevitables, con respecto a la integración, requerimientos de soporte, desempeño, eficiencia y confiabilidad.

Si se escoge incorrectamente, el data warehouse se convierte en una gran empresa con problemas difíciles de trabajar en su entorno, costoso para arreglar y difícil de justificar.

Para conseguir que la implementación del depósito tenga un inicio exitoso, se necesita enfocar hacia tres bloques claves de construcción:

- Arquitectura total del depósito
- Arquitecturas del servidor
- Sistemas de Gestión de Base de Datos

1.12.1. ARQUITECTURA DEL DEPÓSITO

El desarrollo del data warehouse comienza con la estructura lógica y física de la base de datos del depósito más los servicios requeridos para operar y mantenerlo. Esta elección conduce a la selección de otros dos ítems fundamentales: el servidor de hardware y el DBMS.

La plataforma física puede centralizarse en una sola ubicación o distribuirse regional, nacional o internacionalmente. A continuación se dan las siguientes alternativas de arquitectura:

1.12.1.1. Arquitectura de depósito centralizada

Un plan para almacenar los datos de su compañía, que podría obtenerse desde fuentes múltiples internas y externas, es consolidar la base de datos en un data warehouse integrado. El enfoque consolidado proporciona eficiencia tanto en la potencia de procesamiento como en los costos de soporte. (Ver Figura 14).

En una arquitectura centralizada, una sola, el data warehouse integrado refleja todos los aspectos del negocio. Las bases de datos separadas son todas interrelacionadas y físicamente almacenadas en la misma plataforma.

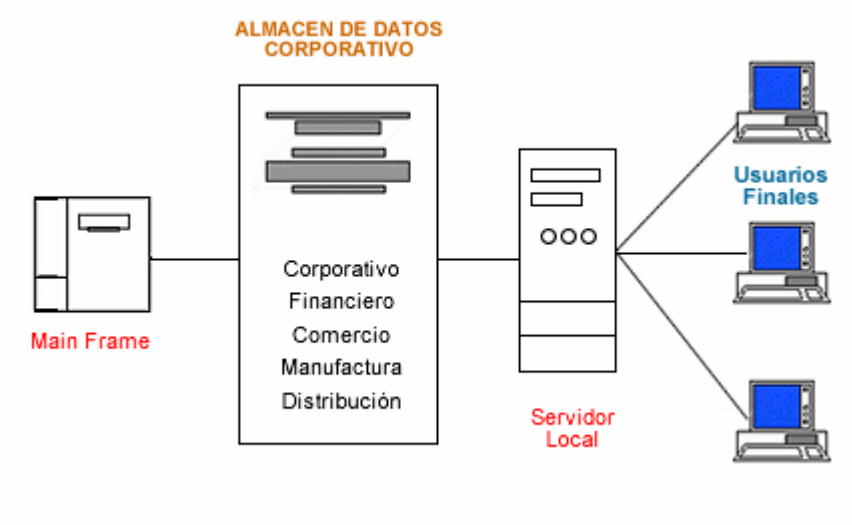

Figura 14. Arquitectura de depósito centralizada

1.12.1.2. Arquitectura de depósito global

La arquitectura global distribuye información por función, con datos financieros sobre un servidor en un sitio, los datos de comercialización en otro y los datos de fabricación en un tercer lugar. (Ver Figura 15).

La data es consolidada lógicamente pero se almacena por separado sin las bases de datos físicas relacionadas, en los mismos sitios físicos o en diferentes.

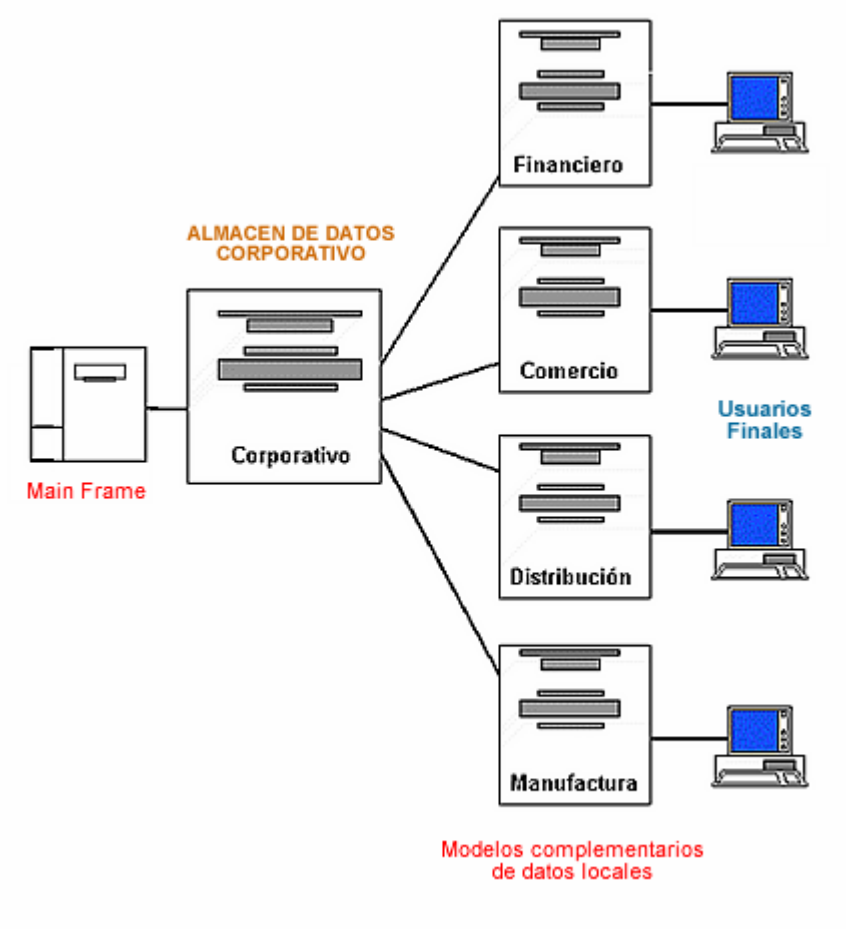

Figura 15. Arquitectura de depósito Global

1.12.1.3. Arquitectura de depósito por niveles

Una arquitectura por niveles almacena datos altamente resumidos sobre una estación de trabajo del usuario, con resúmenes más detallados en un segundo servidor y la información más detallada en un tercero.

La estación de trabajo del primer nivel maneja la mayoría de los pedidos para los datos, con pocos pedidos que pasan sucesivamente a los niveles 2 y 3 para la resolución.

Las computadoras en el primer nivel pueden optimizarse para usuarios de carga pesada y volumen bajo de datos, mientras que los servidores de los otros niveles son más adecuados para procesar los volúmenes pesados de datos, pero cargas más livianas de usuario. (Ver figura 16).

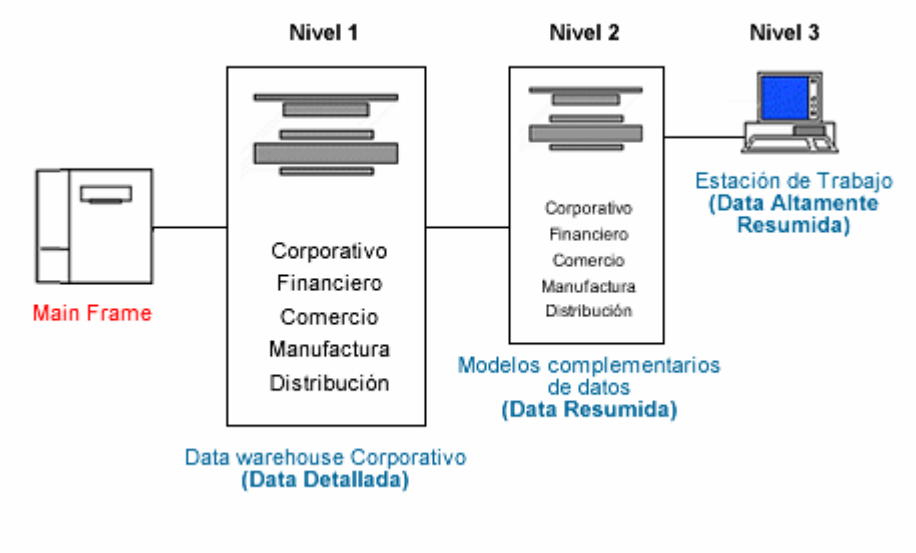

Figura 16. Arquitectura de depósito por niveles

1.12.2. ARQUITECTURA DEL SERVIDOR

Al decidir sobre una estructura de depósito distribuida o centralizada, también se necesita considerar los servidores que retendrán y entregarán los datos. El tamaño de su implementación (y las necesidades de su empresa para escalabilidad, disponibilidad y gestión de sistemas) influirá en la elección de la arquitectura del servidor.

1.12.2.1. Servidores de un solo procesador

Los servidores de un sólo procesador son los más fáciles de administrar, pero ofrecen limitada potencia de procesamiento y escalabilidad. Además, un servidor sólo presenta un único punto de falla, limitando la disponibilidad garantizada del depósito.

Se puede ampliar un solo servidor de redes mediante arquitecturas distribuidas que hacen uso de subproductos, tales como Ambientes de Computación Distribuida (Distributed Computing Environment - DCE) o Arquitectura Broker de Objeto Común (Common Objects Request Broker Architecture - CORBA), para distribuir el tráfico a través de servidores múltiples.

Estas arquitecturas aumentan también la disponibilidad, debido a que las operaciones pueden cambiarse al servidor de backup si un servidor falla, pero la gestión de sistemas es más compleja.

1.12.2.2. Multiprocesamiento simétrico

Las máquinas de multiprocesamiento simétrico (Symmetric MultiProcessing - SMP) aumentan mediante la adición de procesadores que comparten la memoria interna de los servidores y los dispositivos de almacenamiento de disco.

Se puede adquirir la mayoría de SMP en configuraciones mínimas (es decir, con dos procesadores) y levantar cuando es necesario, justificando el crecimiento con las necesidades de procesamiento. La escalabilidad de una máquina SMP alcanza su límite en el número máximo de procesadores soportados por los mecanismos de conexión (es decir, el backplane y bus compartido).

1.12.2.3. Procesamiento en paralelo masivo

Una máquina de procesamiento en paralelo masivo (Massively Parallel Processing - MPP), conecta un conjunto de procesadores por medio de un enlace de banda ancha y de alta velocidad. Cada nodo es un servidor, completo con su propio procesador (posiblemente SMP) y memoria interna. Para optimizar una arquitectura MPP, las aplicaciones deben ser "paralelizadas" es decir, diseñadas para operar por separado, en partes paralelas.

Esta arquitectura es ideal para la búsqueda de grandes bases de datos. Sin embargo, el DBMS que se selecciona debe ser uno que ofrezca una versión paralela. Y aún entonces, se requiere un diseño y afinamiento esenciales para obtener una óptima distribución de los datos y prevenir "hot spots" o "data skew" (donde una cantidad desproporcionada del procesamiento es cambiada a un nodo de procesamiento, debido a la partición de los datos bajo su control).

1.12.2.4. Acceso de memoria no uniforme

La dificultad de mover aplicaciones y los DBMS a agrupaciones o ambientes realmente paralelos ha conducido a nuevas y recientes arquitecturas, tales como el acceso de memoria no uniforme (Non Uniform Memory Access - NUMA).

NUMA crea una sola gran máquina SMP al conectar múltiples nodos SMP en un solo (aunque físicamente distribuida) banco de memoria y un ejemplo único de OS. NUMA facilita el enfoque SMP para obtener los beneficios de performance de las grandes máquinas MPP (con 32 o más procesadores), mientras se mantiene las ventajas de gestión y simplicidad de un ambiente SMP estándar.

Lo más importante de todo, es que existen DBMS y aplicaciones que pueden moverse desde un solo procesador o plataforma SMP a NUMA, sin modificaciones.

1.12.3. SISTEMAS DE GESTIÓN DE BASES DE DATOS

Los data warehouses (conjuntamente con los sistemas de soporte de decisión [Decision Support Systems - DSS] y las aplicaciones cliente/servidor), fueron los primeros éxitos para el DBMS relacional (Relational Data Base Management Systems - RDBMS).

Mientras la gran parte de los sistemas operacionales fueron resultados de aplicaciones basadas en antiguas estructuras de datos, los depósitos y sistemas de soporte de decisiones aprovecharon el RDBMS por su flexibilidad y capacidad para efectuar consultas con un único objetivo concreto.

Los RDBMS son muy flexibles cuando se usan con una estructura de datos normalizada. En una base de datos normalizada, las estructuras de datos son no redundantes y representan las entidades básicas y las relaciones descritas por los datos (por ejemplo productos, comercio y transacción de ventas). Pero un procesamiento analítico en línea (OLAP) típico de consultas que involucra varias estructuras, requiere varias operaciones de unión para colocar los datos juntos.

La performance de los RDBMS tradicionales es mejor para consultas basadas en claves ("Encuentre cuenta de cliente #2014") que para consultas basadas en el contenido ("Encuentre a todos los clientes con un ingreso sobre $ 10,000 que hayan comprado un automóvil en los últimos seis meses").

Para el soporte de depósitos a gran escala y para mejorar el interés hacia las aplicaciones OLAP, los proveedores han añadido nuevas características al RDBMS tradicional. Estas, también llamadas características super relacionales, incluyen el soporte para hardware de base de datos especializada, tales como la máquina de base de datos Teradata.

Los modelos super relacionales también soportan extensiones para almacenar formatos y operaciones relacionales (ofrecidas por proveedores como RedBrick) y diagramas de indexación especializados, tales como aquellos usados por Sybase IQ. Estas técnicas pueden mejorar el rendimiento para las recuperaciones basadas en el contenido, al pre juntar tablas usando índices o mediante el uso de listas de índice totalmente invertidos.

Muchas de las herramientas de acceso a los data warehouses explotan la naturaleza multidimensional del data warehouse. Por ejemplo, los analistas de marketing necesitan buscar en los volúmenes de ventas por producto, por mercado, por período de tiempo, por promociones y niveles anunciados y por combinaciones de estos diferentes aspectos.

La estructura de los datos en una base de datos relacional tradicional, facilita consultas y análisis a lo largo de dimensiones diferentes que han llegado a ser comunes. Estos esquemas podrían usar tablas múltiples e indicadores para simular una estructura multidimensional. Algunos productos DBMS, tales como Essbase y Gentium, implementan técnicas de almacenamiento y operadores que soportan estructuras de datos multidimensionales.

Mientras las bases de datos multidimensionales (MultiDimensional Databases - MDDBs) ayudan directamente a manipular los objetos de datos multidimensionales (por ejemplo, la rotación fácil de los datos para verlos entre dimensiones diferentes, o las operaciones de drill down que sucesivamente exponen los niveles de datos más detallados), se debe identificar estas dimensiones cuando se construya la estructura de la base de datos. Así, agregar una nueva dimensión o cambiar las vistas deseadas, puede ser engorroso y costoso. Algunos MDDBs requieren un recargue completo de la base de datos cuando ocurre una reestructuración.

Nuevas Dimensiones

Una limitación de un RDBMS y un MDDB, es la carencia de soporte para tipos de datos no tradicionales como imágenes, documentos y clips de video/ audio. Si usted necesita estos tipos de objetos en su data warehouse, busque un DBMS relacional-objeto (Ejemplo: Illustra de Informix).

Por su enfoque en los valores de datos codificados, la mayor parte de los sistemas de base de datos pueden acomodar estos tipos de datos, sólo con extensiones basadas en cierta referencias, tales como indicadores de archivos que contienen los objetos. Muchos RDBMS almacenan los datos complejos como objetos grandes binarios (Binary Large Objects - BLOBs). En este formato, los objetos no pueden ser indexados, clasificados, o buscados por el servidor.

Los DBMS relacional-objeto, de otro lado, almacenan los datos complejos como objetos nativos y pueden soportar las grandes estructuras de datos encontradas en un ambiente orientado a objetos.

Estos sistemas de base de datos naturalmente acomodan no sólo tipos de datos especiales sino también los métodos de procesamiento que son únicos para cada uno de ellos.

Pero una desventaja del enfoque relacional-objeto, es que la encapsulación de los datos dentro de los tipos especiales de datos (una serie de precios de stock a través del tiempo en cada registro de una tabla de stock, por ejemplo), requiere de operadores especializados para que hagan búsquedas simples previamente (por ejemplo, "Encontrar todas las existencias que han mostrado una disminución en el precio de Abril a Mayo 1996").

La selección del DBMS está también sujeta al servidor de hardware que se usa. Algunos RDBMS, como el DB2 Paralelo, Informix XPS y el Oracle Paralelo, ofrecen versiones que soportan operaciones paralelas. El software paralelo divide consultas, uniones a través de procesadores múltiples y corre estas operaciones simultáneamente para mejorar la performance.

Se requiere el paralelismo para el mejor desempeño en los servidores MPP grandes y SMP agrupados. No es aún una opción con MDDBS o DBMS relacional-objeto.

En la tabla 4 se debe resumir los pro y los contra de los diferentes tipos de DBMS para operaciones de data warehouse.

	SGBD			
Características	**Relacional**	**Super-Relacional**	**Multidimensional**	**Objeto-Relacional**
Estructuras Normalizadas				
Tipos de datos abstractos				
Paralelismo				
Estructuras Multidimensionales				
Drill-Down				
Rotación				
Operaciones dependientes de datos				
Entre otras…				

Tabla 4. Cuadro comparativo de SGBD

La tabla 5 contiene algunos ejemplos de cómo afectan estos criterios de decisión en la elección de una arquitectura de servidor/ data warehouse.

<table>
<tr><th colspan="3">PARA ESTOS AMBIENTES …</th><th colspan="3">ELIJA …</th></tr>
<tr><th>Requerimientos comerciales</th><th>Usuarios</th><th>Soporte de Sistemas</th><th>Arquitectura</th><th>Servidor</th><th>DBMS</th></tr>
<tr><td>Alcance: departamental
Usos: análisis de datos</td><td>Pequeña: ubicación única</td><td>Local mínimo
central promedio</td><td>Consolidado paquete</td><td>Procesador único o SMP</td><td>MDDB</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>Alcance: departamental</td><td rowspan="2">Grande: analistas en una sola ubicación y los usuarios informáticos dispersos</td><td>Local mínimo</td><td rowspan="2">Seccionado: Detalle en central. Resumen en local</td><td>Grupos de SMP para central;</td><td>RDBMS para central</td></tr>
<tr><td>Usos: análisis más informática</td><td>central promedio</td><td>SP o SMP para local</td><td>MDDB para local</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>Alcance: empresa
Usos: análisis más informática</td><td>Grande: geográficamente disperso</td><td>Central fuerte</td><td>Centralizado</td><td>Grupos de SMP</td><td>Objeto-relacional con soporte Web</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>Alcance: departamental
Usos: investigación</td><td>Pequeña: pocas ubicaciones</td><td>Central fuerte</td><td>Centralizado</td><td>MPP</td><td>RDBMS con soporte paralelo</td></tr>
</table>

Tabla 5. Matriz de Decisión

1.12.4. COMBINACIÓN DE LA ARQUITECTURA CON EL DBMS

Para seleccionar la combinación correcta de la arquitectura del servidor y el DBMS, primero es necesario comprender los requerimientos comerciales de su compañía, su población de usuarios y las habilidades del personal de soporte.

Las implementaciones de los data warehouses varían apreciablemente de acuerdo al área. Algunos son diseñados para soportar las necesidades de análisis específico para un solo departamento o área funcional de una organización, tales como finanzas, ventas o marketing. Las otras implementaciones reúnen datos a través de toda la empresa para soportar una variedad de grupos de usuarios y funciones. Por regla general, a mayor área del depósito, se requiere mayor potencia y funcionalidad del servidor y el DBMS.

Los modelos de uso de los data warehouses son también un factor. Las consultas y vistas de reportes preestructuradas frecuentemente satisfacen a los usuarios informáticos, mientras que hay menos demandas sobre el DBMS y la potencia de procesamiento del servidor. El análisis complejo, que es típico de los ambientes de decisión-soporte, requiere más poder y flexibilidad de todos los componentes del servidor. Las búsquedas masivas de grandes data warehouses favorecen el paralelismo en el DBMS y el servidor.

Los ambientes dinámicos, con sus requerimientos siempre cambiantes, se adaptan mejor a una arquitectura de datos simple, fácilmente cambiable (por ejemplo, una estructura relacional altamente normalizada), antes que una estructura intrincada que requiere una reconstrucción después de cada cambio (por ejemplo, una estructura multidimensional).

El valor de la data fresca requerida indica cuán importante es para el data warehouse renovar y cambiar los datos. Los grandes volúmenes de datos que se refrescan a intervalos frecuentes, favorecen una arquitectura físicamente centralizada para soportar una captura de datos eficiente y minimizar el tiempo de transporte de los datos.

Un perfil de usuario debería identificar quiénes son los usuarios de su data warehouse, dónde se ubican y cuántos necesita soportar. La información sobre cómo cada grupo espera usar los data warehouses, ayudará a analizar los diversos estilos de uso.

Conocer la ubicación física de sus usuarios ayudará a determinar cómo y a qué área necesita distribuir el data warehouse. Una arquitectura por niveles podría usar servidores en el lugar de las redes de área local. O puede necesitar un enfoque centralizado para soportar a los trabajadores que se movilizan y que trabajan en el depósito desde sus laptops.

El número total de usuarios y sus modelos de conexión determinan el tamaño de sus servidores de depósito. Los tamaños de memoria y los canales de I/O deben soportar el número previsto de usuarios concurrentes bajo condiciones normales, así como también en las horas punta de su organización.

Finalmente, se debe factorizar la sofisticación del personal de soporte. Los recursos de los sistemas de información (Information System - IS) que están disponibles dentro de su organización,

pueden limitar la complejidad o sofisticación de la arquitectura del servidor. Sin el personal especializado interno o consultores externos, es difícil de crear y mantener satisfactoriamente una arquitectura que requiere paralelismo en la plataforma del servidor (MPP o SMP agrupado, por ejemplo).

1.13. CONFIABILIDAD DE LOS DATOS

La data "sucia" es peligrosa. Las herramientas de limpieza especializadas y las formas de programar de los clientes proporcionan redes de seguridad.

No importa cómo esté diseñado un programa o cuán hábilmente se use. Si se alimenta mala información, se obtendrá resultados incorrectos o falsos. Desafortunadamente, los datos que se usan satisfactoriamente en las aplicaciones de línea comercial operacionales pueden ser basura en lo que concierne a la aplicación data warehousing.

Los datos "sucios" pueden presentarse al ingresar información en una entrada de datos (por ejemplo, "Sitsemas S. A." en lugar de "Sistemas S. A.") o de otras causas. Cualquiera que sea, la data sucia daña la credibilidad de la implementación del depósito completo. A continuación, en la Figura 17 se muestra un ejemplo de formato de ventas en el que se pueden presentar errores.

Afortunadamente, las herramientas de limpieza de datos pueden ser de gran ayuda. En algunos casos, puede crearse un programa de limpieza efectivo. En el caso de bases de datos grandes, imprecisas e inconsistentes, el uso de las herramientas comerciales puede ser casi obligatorio.

Decidir qué herramienta usar es importante y no solamente para la integridad de los datos. Si se equivoca, se podría malgastar semanas en recursos de programación o cientos de miles de dólares en costos de herramientas.

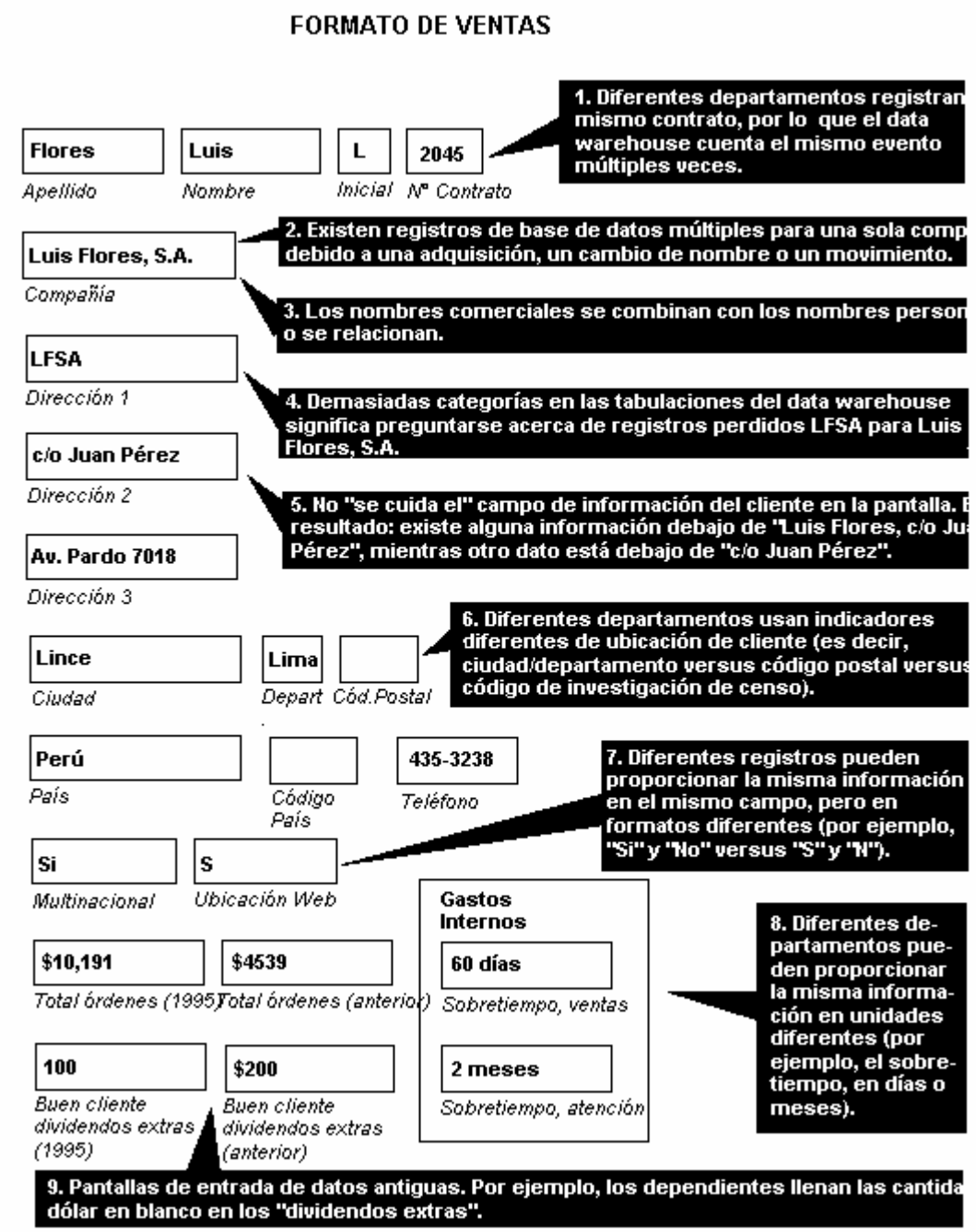

Figura 17. Ejemplo de un formato de información

1.13.1. LIMPIEZA DE LOS DATOS

La limpieza de una data "sucia" es un proceso multifacético y complejo. Los pasos a seguir son los siguientes:

1. Analizar sus datos corporativos para descubrir inexactitudes, anomalías y otros problemas.
2. Transformar los datos para asegurar que sean precisos y coherentes.
3. Asegurar la integridad referencial, que es la capacidad del data warehouse, para identificar correctamente al instante cada objeto del negocio, tales como un producto, un cliente o un empleado.
4. Validar los datos que usa la aplicación del data warehouse para realizar las consultas de prueba.
5. Producir la metadata, una descripción del tipo de datos, formato y el significado relacionado al negocio de cada campo.
6. Finalmente, viene el paso crucial de la documentación del proceso completo para que se pueda ampliar, modificar y arreglar los datos en el futuro con más facilidad.

En la práctica, se tendría que realizar múltiples pasos como parte de una operación única o cuando use una sola herramienta. En particular, limpiar la data y asegurar la integridad referencial son procesos interdependientes.

Las herramientas comerciales pueden ayudar en cada uno de estos pasos. Sin embargo, es posible escribir sus propios programas para hacer el mismo trabajo.

Los programas de limpieza de datos no proporcionan mucho razonamiento, por lo que las compañías necesitan tomar sus decisiones en forma manual, basados en información importante y reportes de auditoría de datos.

Cada vez que se carga un nuevo conjunto de datos, la limpieza de datos comúnmente constituye cerca del 25 por ciento de lo que puede ser un proceso de cuatro semanas.

A continuación, se darán algunos ejemplos de las experiencias de las empresas que han realizado limpieza de datos para un ambiente data warehousing.

Ejemplo 1:

CompuCom Systems, un gran integrador de sistemas basados en Dallas, implementó un registro de 12 millones, en un depósito de 10 Gb para el soporte de decisiones internas y de los clientes, según el orden y la condición y producir información por medio del Web.

CompuCom implementó algunas rutinas de mejoramiento de datos en lenguajes de cuarta generación (4GL), asociado con su base de datos Progress, la cual corre sobre un HP 9000. El incremento incluye desciframiento de valores de columnas en descripciones inglesas cortas o mnemotecnia. El código de limpieza de datos, tales como las conversiones de fecha y datos, están escritas en lenguaje C.

La ventaja de ésto es que CompuCom ahora posee estas rutinas y puede usarlas en otras aplicaciones.

Los usuarios ayudaron a definir los requerimientos de limpieza de datos, ya que son ellos los que mejor conocen los datos y pueden informar sobre qué tipo de datos sucios deben salir y cómo limpiarlos.

La compañía no usa una herramienta de limpieza comercial porque gran parte de sus datos está en la misma forma básica. Así, la compañía puede fácilmente usar de nuevo las rutinas escritas.

La desventaja principal ha sido la cantidad de tiempo de desarrollo (alrededor de una semana) que se necesitó para crear las rutinas. Aunque tienen cierta dificultad de tiempo para mantenerse al día con la demanda y han buscado paquetes de software [comercial], no han encontrado aún, en el mercado, algo que se ajuste mejor a sus requerimientos.

Ejemplo 2:

Ohio Casualty Insurance (Hamilton, OH) experimentó por dos años con la limpieza in-house, usando programas COBOL, antes de usar la herramienta comercial, Integrity Data Reengineering Tool de Vality Technology.

El data warehouse de Ohio Casualty combina registros asociados con alrededor de 1 millón de pólizas de seguro personales, incluyendo auto y pólizas de casa propia. Como una prueba, la compañía comenzó con 3,500 pólizas de sus empleados.

Sin embargo, es difícil tratar de programar para todas las situaciones en que se puede caer. Después de tomar un año en desarrollar programas genéricos de extraer/ transformar/cargar, se necesitó otro año, para programar en Cobol y editar el manual, para conseguir los datos de las pólizas correctos para el depósito.

La herramienta Vality Integrity Data Reengineering ayuda a atacar el primer conjunto de datos de los clientes - alrededor de 15, 000 pólizas en el centro comercial Denver de la compañía. Aunque el personal de Ohio Casualty todavía necesita investigar las anomalías que ha descubierto el producto Vality, no se ha requerido ninguna programación o redacción del manual de los datos. Los datos estuvieron listos para el depósito en alrededor de seis semanas.

Ejemplo 3:
Intel (Hillsboro) es un ejemplo de compañía que ha realizado exitosamente una limpieza de datos in-house, aunque con ciertos problemas. Inicialmente pretendió encargar su limpieza de datos a una agencia de servicios, para un depósito de aproximadamente 1 millón de registros tomados desde cinco sistemas operacionales.

La agencia de servicios prometió identificar las relaciones entre los diversos grupos dentro de las compañías clientes. Además, la agencia proveería información industrial para las organizaciones de clientes, tales como el número de empleados, las rentas y el crecimiento, las cuales serían valiosas para las ventas de Intel. Desafortunadamente, la agencia de servicio no hizo un buen trabajo de identificar las relaciones entre los clientes, lo que dio como resultado el hecho de que algunas personas estuvieron asociadas con compañías equivocadas.

Intel tomó la cinta de la agencia de servicio y luego corrió los datos con el paquete de análisis estadístico SAS, del Instituto SAS, para identificar y corregir los problemas con las relaciones con un tope de 10 agrupaciones (es decir, las primeras compañías en una relación jerárquica única).

La compañía luego usó las herramientas de base de datos Oracle para propiciar el análisis y la limpieza. Ya que la nueva data llegaba todo el tiempo, algunas de las rutinas de limpieza de Oracle fueron implementadas como procedimientos almacenados para que puedan correr automáticamente contra la nueva data.

Intel aún persiste en encargar las tareas de la limpieza de los datos. Sin embargo, la compañía planea mantener la limpieza in-house hasta que encuentre una agencia de servicio aceptable.

Ejemplo 4:

CrediCard (São Paulo, Brasil), un gran emisor de tarjetas de crédito en Sudamérica, consiguió herramientas de limpieza y mejora de datos como parte de la implementación de un data warehouse por Market Knowledge, una filial de Equifax.

El personal de comercialización de CrediCard usa aproximadamente 200 rutinas para efectuar operaciones de limpieza, tales como la eliminación de datos malos o sin uso, corrección de valores equivocados y estandarización de formatos diversos.

Además, ellos pueden mejorar los datos al realizar operaciones como corrección de cantidades monetarias por la inflación y la devaluación, creando un campo de edad virtual basado en la fecha de nacimiento de una persona y añadiendo datos de censos a los registros entrantes. Estas rutinas (por ejemplo, corrección de inflación) favorecen particularmente a los requerimientos brasileños.

Ellos además están diseñados para el uso del personal de comercialización no-técnico. Las rutinas de limpieza de los datos, las cuales son programadas como comandos SQL, empleó sólo alrededor de tres personas por semana para crearlas - una porción mínima de un proyecto de 2 años y medio.

Las herramientas para mejorar los datos, más automatizadas y más inteligentes, representan alrededor de $ 120,000 del total del proyecto de $ 840,000.

1.13.2. TIPOS DE LIMPIEZA DE DATOS

1.13.2.1. Limpieza de datos moderada

Si decide no programar funciones de limpieza de datos o contratar un consultor para hacer el trabajo, puede inhibirse también de la compra de una herramienta específica para esa tarea. El software de gestión del data warehouse puede ser suficiente para limpiar y validar según sus propósitos.

Muchos proyectos de data warehouse usan productos como Warehouse Manager de Prism Solutions o Passport de Carleton, para una gama de tareas de gestión de data warehouse, que incluyen:

- Extracción de los datos desde las bases de datos operacionales
- Preparación de los datos para cargarlos en una base de datos del depósito,
- Administración de la metadata.

Estos productos cuestan desde $ 75,000 a más de $ 200,000, dependiendo del tamaño y la complejidad del proyecto y pueden también limpiar, transformar y validar.

Ejemplo 5:
La Universidad Emory (Atlanta) hace la limpieza de toda la data para su depósito de 6 Gb con programas en Cobol generados por Prism Warehouse Manager. Además de tener problemas típicos, tales como formatos múltiples de fecha, la data con frecuencia contiene campos no inicializados que retienen valores arbitrarios. Dos miembros del personal utilizan como 4 horas de un día de trabajo en las tareas de limpieza de datos.

Emory ha considerado usar herramientas de limpieza de datos especializados, pero la escuela está eliminando la data sucia hasta ahora, lo suficientemente bien, que no ve el valor adicional en otros productos comerciales para justificar la compra.

Sin embargo, tienen una buena oportunidad de que las herramientas mencionadas anteriormente de Prism y Carleton no limpien todo lo que se necesite. Ellos pueden encontrar anomalías

comunes que pueden manejarse mediante simples tablas de búsqueda de información (por ejemplo, reconocer que Avenida y Av. representan la misma información), pero podrían no salir exitosos con irregularidades más importantes e impredecibles, porque estas herramientas no están diseñadas para hacer tipos de limpieza de gran intensidad.

Si los datos que requieren limpieza consisten predominantemente de nombres (incluyendo nombres de compañía) y direcciones, las compañías tales como Harte-Hanks Communications e Innovative Systems proveen no solamente herramientas de software, sino que actualizan periódicamente los archivos de datos para ayudar a combinar las variantes de los nombres de las compañías, detectar códigos postales que no corresponden a las direcciones proporcionadas y encontrar anomalías similares.

Estas herramientas pueden ser apropiadas en otros campos (aparte de nombres y direcciones) que sean conocidos para ser corregidos (por ejemplo, cantidades de dólar devaluados que han sido validados por las cuentas) o contengan información independiente que no será usada como una llave o índice (por ejemplo, las anotaciones de contacto de los vendedores).

Las soluciones orientadas al nombre y la dirección pueden costar en cualquier parte desde $ 30,000 a más de $ 200,000, dependiendo del tamaño del data warehouse en cuestión. Además se necesita, una herramienta de extraer/ transformar/cargar (Extract, Transform, Load - ETL), tales como el Warehouse Manager o Passport.

Lamentablemente, en el país no existen empresas que se especialicen en estas actividades. Sólo corporaciones internacionales como las de Arthur Andersen han efectuado limpieza de datos en nuestro medio en bancos privados y muy pocos organismos públicos.

1.13.2.2. Limpieza de datos intensa

Para trabajos de limpieza intensos, se deben considerar herramientas que se han desarrollado para esas tareas. Existen dos grandes competidores: Enterprise/Integrator de Apertus Technologies y la herramienta Integrity Data Reengineering de Vality.

Enfoque Top-Down

La empresa Enterprise/Integrator toma un enfoque top-down, en la que usted propone las reglas para limpiar los datos. Esta es una estrategia directa, donde usted impone sus conocimientos sobre su negocio en los datos.

Por ejemplo:

- ¿Desea usted tratar una serie de concesiones de Martha's Fried Chicken como un cliente único con direcciones múltiples?
- Para los propósitos del data warehouse, ¿tiene sentido sustituir una dirección central única para las diferentes direcciones de las concesiones?
- O, ¿le gustaría tratar las ubicaciones de las concesiones como clientes completamente diferentes?

Esta decisión determina cómo se agrega o consolida estos registros y si se trata las diferentes direcciones de Martha's Fried Chicken como excepciones.

La empresa Enterprise/Integrator ofrece no solamente limpieza de datos, sino también extracción, transformación, carga de datos, repetición, sincronización y administración de la metadata. Es bastante caro (de $130,000 a $250,000), pero se puede ahorrar dinero si elimina la necesidad de otras herramientas de gestión de data warehouse.

La desventaja principal del enfoque top-down de Enterprise/Integrator es que usted tiene que conocer, o ser capaz de deducir las reglas del negocio y de la limpieza de datos.

Apertus provee ejemplos para trabajar con muchas estructuras comerciales y excepciones comunes. Aún así, crear reglas es consumo de tiempo y esté seguro de encontrar algunas excepciones no esperadas. Estos pueden manejarse manualmente mediante un sistema de excepto - manipulación, pero es un proceso que consume tiempo.

Enfoque Bottom-Up

La herramienta Integrity Data Reengineering de Vality tiene un enfoque bottom-up. Analiza los datos caracter por caracter y automáticamente emergen los modelos y las reglas del negocio. Integrity proporciona un diseño de la data para ayudar a normalizar, condicionar y consolidar los datos. Este enfoque tiende a dejar pocas excepciones para manejarse manualmente y el proceso tiende a consumir menos tiempo.

Al igual que Enterprise/Integrator, Integrity puede tomar en cuenta las relaciones comerciales que no son obvias a partir de los datos, tales como fusiones y adquisiciones que han tenido lugar desde que fueron creados los datos. Pero con cualquier herramienta, estas reglas deben imponerse con un modelo top-down.

Integrity incide exclusivamente sobre la limpieza de los datos, comenzando desde los archivos básicos. No extrae los datos desde bases de datos operacionales, carga los datos en la base de datos del depósito, duplica y sincroniza los datos o administra la metadata.

Por ello, además de costar $ 250,000, Integrity podría requerir también una herramienta como Warehouse Manager o Passport. Sin embargo, pueden ser suficientes los utilitarios disponibles con la base de datos para una simple extracción/carga.

1.14. IMPACTOS DE UN DATA WAREHOUSE

El éxito de DW no está en su construcción, sino en usarlo para mejorar procesos empresariales, operaciones y decisiones. Posicionar un DW para que sea usado efectivamente, requiere entender los impactos de implementación en los siguientes ámbitos:

- Impactos Humanos
- Impactos Empresariales
- Impactos Técnicos

1.14.1. IMPACTOS HUMANOS

Efectos sobre la gente de la empresa:

Construcción del DW: Construir un DW requiere la participación activa de quienes usarán el DW. A diferencia del desarrollo de aplicaciones, donde los requerimientos de la empresa logran ser relativamente bien definidos producto de la estabilidad de las reglas de negocio a través del tiempo, construir un DW depende de la realidad de la empresa como de las condiciones que en ese momento existan, las cuales determinan qué debe contener el DW. La gente de negocios debe participar activamente durante el desarrollo del DW, desde una perspectiva de construcción y creación.

Accesando el DW: El DW intenta proveer los datos que posibilitan a los usuarios accesar su propia información cuando ellos la necesitan. Esta aproximación para entregar información tiene varias implicancias:

- La gente de la empresa puede necesitar aprender nuevas destrezas.
- Análisis extensos y demoras de programación para obtener información será eliminada. Como la información estará lista para ser accesada, las expectativas probablemente aumentarán.

- Nuevas oportunidades pueden existir en la comunidad empresarial para los especialistas de información.
- La gran cantidad de reportes en papel serán reducidas o eliminadas.
- La madurez del DW dependerá del uso activo y retroalimentación de sus usuarios.

Usando aplicaciones DSS/EIS: Usuarios de aplicaciones DSS y EIS necesitarán menos experiencia para construir su propia información y desarrollar nuevas destrezas.

1.14.2. IMPACTOS EMPRESARIALES.

Corresponden a los procesos empresariales y decisiones empresariales. Se deben considerar los beneficios empresariales potenciales de los siguientes impactos:

- Los Procesos de Toma de Decisiones pueden ser mejorados mediante la disponibilidad de información. Decisiones empresariales se hacen más rápidas por gente más informada.
- Los procesos empresariales pueden ser optimizados. El tiempo perdido esperando por información que finalmente es incorrecta o no encontrada, es eliminada.
- Conexiones y dependencias entre procesos empresariales se vuelven más claros y entendibles. Secuencias de procesos empresariales pueden ser optimizados para ganar eficiencia y reducir costos.
- Procesos y datos de los sistemas operacionales, así como los datos en el DW, son usados y examinados. Cuando los datos son organizados y estructurados para tener significado empresarial, la gente aprende mucho de los sistemas de información. Pueden quedar expuestos posibles defectos en aplicaciones actuales, siendo posible entonces mejorar la calidad de nuevas aplicaciones. Comunicación e Impactos Organizacionales.

Apenas el DW comienza a ser fuente primaria de información empresarial consistente, los siguientes impactos pueden comenzar a presentarse:

- La gente tiene mayor confianza en las decisiones empresariales que se toman. Ambos, quienes toman las decisiones como los afectados conocen que está basada en buena información.
- Las organizaciones empresariales y la gente de la cual ella se compone queda determinada por el acceso a la información. De esta manera, la gente queda mejor habilitada para entender su propio rol y responsabilidades como también los efectos de sus contribuciones; a la vez, desarrollan un mejor entendimiento y apreciación con las contribuciones de otros.
- La información compartida conduce a un lenguaje común, conocimiento común, y mejoramiento de la comunicación en la empresa. Se mejora la confianza y cooperación entre distintos sectores de la empresa , viéndose reducida la sectorización de funciones.
- Visibilidad, accesibilidad, y conocimiento de los datos producen mayor confianza en los sistemas operacionales.

1.14.3. IMPACTOS TÉCNICOS

Considerando las etapas de construcción, soporte del DW y soporte de sistemas operacionales, se tienen los siguientes impactos técnicos:

Nuevas destrezas de desarrollo: Cuando se construye el DW, el impacto más grande sobre la gente técnica está dada por la curva de aprendizaje, muchas destrezas nuevas se deben aprender, incluyendo:

- Conceptos y estructura DW.
- El DW introduce muchas tecnologías nuevas (ETT, Carga, Acceso de Datos, Catálogo de Metadatos, Implementación de DSS/EIS), y cambia la manera que nosotros usamos la tecnología existente. Nuevas responsabilidades de soporte, nuevas demandas de recursos y nuevas expectativas, son los efectos de estos cambios.
- Destrezas de diseño y análisis donde los requerimientos empresariales no son posibles de definir de una forma estable a través del tiempo.
- Técnicas de desarrollo incremental y evolutivo.

- Trabajo en equipo cooperativo con gente de negocios como participantes activos en el desarrollo del proyecto.

Nuevas responsabilidades de operación: Cambios sobre los sistemas y datos operacionales deben ser examinados más cuidadosamente para determinar el impacto que estos cambios tienen sobre ellos, y sobre el DW.

1.15. COSTOS DE UN DATA WAREHOUSE

1.15.1. COSTOS DE CONSTRUCCIÓN

Los costos de construir un DW son similares para cualquier proyecto de tecnología de información. Estos pueden ser clasificados en tres categorías:

RRHH: La gente necesita contar con un enfoque fuerte sobre el conocimiento del área de la empresa y de los procesos empresariales. Además es muy importante considerar las cualidades de la gente, ya que el desarrollo del DW requiere participación de la gente de negocios como de los especialistas tecnológicos; estos dos grupos de gente deben trabajar juntos, compartiendo su conocimiento y destrezas en un espíritu de equipo de trabajo, para enfrentar los desafíos de desarrollo del DW.

Tiempo: Se debe establecer el tiempo no tan solo para la construcción y entrega de resultados del DW, sino también para la planeación del proyecto y la definición de la arquitectura. La planeación y la arquitectura, establecen un marco de referencia y un conjunto de estándares que son críticos para la eficacia del DW.

Tecnología: Muchas tecnologías nuevas son introducidas por el DW. El costo de la nueva tecnología puede ser tan sólo la inversión inicial del proyecto.

1.15.2. COSTOS DE OPERACIÓN

Una vez que está construido y entregado un DW debe ser soportado para que tenga valor empresarial. Son justamente estas actividades de soporte, la fuente de continuos costos operacionales para un DW. Se pueden distinguir tres tipos de costos de operación:

Evolutivos: Ajustes continuos del DW a través del tiempo, como cambios de expectativas y, cambios producto del aprendizaje del RRHH del proyecto mediante su experiencia usando el DW.

Crecimiento: Incrementos en el tiempo en volúmenes de datos, del número de usuarios del DW, lo cual conllevará a un incremento de los recursos necesarios como a la demanda de monitoreo, administración y sintonización del DW (evitando así, un incremento en los tiempos de respuesta y de recuperación de datos, principalmente).

Cambios: El DW requiere soportar cambios que ocurren tanto en el origen de datos que éste usa, como en las necesidades de la información que éste soporta.

Los dos primeros tipos de costos de operación, son básicos en la manutención de cualquier sistema de información, por lo cual no nos resultan ajenos; sin embargo, se debe tener especial cuidado con los costos de operación por cambios, ya que ellos consideran el impacto producto de la relación del OLTP y del Ambiente Empresarial, con el DW.

Resulta esencial para llevar a cabo un proyecto DW, tener claridad en la forma que éste se ve afectado por medio de cambios a nivel de OLTP como del Ambiente Empresarial; por ello entonces, a continuación se analiza más en detalle este tipo de costos de operación.

1.15.3. CAMBIOS Y EL DW.

Cuando se implementa un DW, el impacto de cambios es compuesto. Dos orígenes primarios de cambios existen:

Cambios en el ambiente empresarial: Un cambio en el ambiente empresarial puede cambiar las necesidades de información de los usuarios. Así, el contenido del DW se puede ver afectado y las aplicaciones DSS y EIS pueden requerir cambios.

Cambios en la tecnología: Un cambio en la tecnología puede afectar la manera que los datos operacionales son almacenados, lo cual implicaría un ajuste en los procesos de Extracción, Transporte y Carga para adaptar las variaciones presentadas.

Un cambio de cualquiera de ellos impacta los sistemas operacionales. Un cambio en el ambiente operacional puede cambiar el formato, estructura o significado de los datos operacionales usados como origen para el DW. De esta forma serían impactados los procesos de Extracción, Transformación y Carga de datos.

1.16. VALOR DEL DATA WAREHOUSE

El valor de un DW queda descrito en tres dimensiones:

1. **Mejorar la Entrega de Información:** información completa, correcta, consistente, oportuna y accesible. Información que la gente necesita, en el tiempo que la necesita y en el formato que la necesita.
2. **Mejorar el Proceso de Toma de Decisiones:** con un mayor soporte de información se obtienen decisiones más rápidas; así también, la gente de negocios adquiere mayor confianza en sus propias decisiones y las del resto, y logra un mayor entendimiento de los impactos de sus decisiones.

3. **Impacto Positivo sobre los Procesos Empresariales:** Cuando a la gente se le da acceso a una mejor calidad de información, la empresa puede lograr por sí sola:

   - Eliminar los retardos de los procesos empresariales que resultan de información incorrecta, inconsistente y/o no existente.
   - Integrar y optimizar procesos empresariales a través del uso compartido e integrado de las fuentes de información.
   - Eliminar la producción y el procesamiento de datos que no son usados ni necesarios, producto de aplicaciones mal diseñados o ya no utilizados.

1.17. COSTOS V/S VALOR DEL DATA WAREHOUSE

En todo proyecto es importante e inevitable realizar un análisis desde la perspectiva Costo/Valor.

A grandes rasgos, los costos asociados a un proyecto DW incluyen el costo de construcción y, la manutención y operación una vez que está construido. En cuanto al valor, éste considera, el valor de mejorar la entrega de información, el valor de mejorar el proceso de toma de decisiones y el valor agregado para los procesos empresariales.

1.18. SOFTWARE EN UN DATA WAREHOUSE

La información estratégica sobre clientes importantes o un exitoso lanzamiento de producto, se almacena en gigabytes de datos de marketing o índice de transacciones de venta. Esa información debe ser extraída de alguna forma para la toma de decisiones.

En este caso se necesita software especializado que permita capturar los datos relevantes en forma rápida y pueda verse a través de diferentes dimensiones de los datos. El software no debería limitarse únicamente al acceso a los datos, si no también, al análisis significativo de los datos. En efecto, transformar los datos de la información cruda o no procesada, en información útil para la empresa.

Los software's o herramientas de negocios inteligentes se colocan sobre la plataforma data warehousing y proveen este servicio. Debido a que son el punto principal de contacto entre la aplicación del depósito y la gente que lo usa, estas herramientas pueden constituir la diferencia entre el éxito o fracaso de un depósito.

Las herramientas de negocio inteligentes se han convertido en los sucesores de los sistemas de soporte de decisión, pero tienen un alcance más amplio. No solamente ayudan en las decisiones de soporte sino, en muchos casos, estas herramientas soportan muchas funciones operacionales y de misión-crítica de la compañía. Sin embargo, estos productos no son infalibles ya que sólo se consigue el máximo provecho del data warehouse, si elige las herramientas adecuadas a las necesidades de cada usuario final.

Los software usados en un data warehouse se clasifican principalmente en:

- Herramientas de Consulta y Reporte
- Herramientas de Base de Datos Multidimensionales/ Olap (On Line Analytical Processing)
- Sistemas de Información Ejecutivos
- Herramientas Data Minino
- Sistemas de Gestión de Bases de Datos

1.18.1. HERRAMIENTAS DE CONSULTA Y REPORTE

Existe una gran cantidad de poderosas herramientas de consulta y reporte en el mercado (Ver Tabla 6). Algunos proveedores ofrecen productos que permiten tener más control sobre qué procesamiento de consulta es hecho en el cliente y qué procesamiento en el servidor.

Las más simples de estas herramientas son productos de reporte y consultas básicas. Ellos proporcionan desde pantallas gráficas a generadores SQL (o más preciso, generadores de acceso-llamada a base de datos).

Más que aprender SQL o escribir un programa para acceder a la información de una base de datos, las herramientas de consulta al igual que la mayoría de herramientas visuales, le permiten apuntar y dar un click a los menús y botones para especificar los elementos de datos, condiciones, criterios de agrupación y otros atributos de una solicitud de información.

La herramienta de consulta genera entonces un llamado a una base de datos, extrae los datos pertinentes, efectúa cálculos adicionales, manipula los datos si es necesario y presenta los resultados en un formato claro.

Se puede almacenar las consultas y los pedidos de reporte para trabajos subsiguientes, como está o con modificaciones. El procesamiento estadístico se limita comúnmente a promedios, sumas, desviaciones estándar y otras funciones de análisis básicas. Aunque las capacidades varían de un producto a otro, las herramientas de consulta y reporte son más apropiadas cuando se necesita responder a la pregunta ¿"Qué sucedió"? (Ejemplo: ¿"Cómo comparar las ventas de los productos X,Y y Z del mes pasado con las ventas del presente mes y las ventas del mismo mes del año pasado?").

Para hacer consultas más accesibles a usuarios no-técnicos, los productos tales como Crystal Reports de Seagate, Impromptu de Cognos, Reportsmith de Borland, Intelligent Query de IQ Software, Esperant de Software AG y GQL de Andyne, ofrecen interfases gráficas para seleccionar, arrastrar y pegar.

Lo más avanzado de estos productos lo orientará hasta las consultas que tienen sintaxis mala o que devuelven resultados imprevistos. El acceso a los datos han mejorado también con las nuevas versiones de estos productos y los vendedores ya instalan drivers estándares tales como ODBC y 32-bit nativo, hasta fuentes de datos comerciales.

En general, los administradores de data warehouses que usen estos tipos de productos, deben estar dispuestos a ocupar su tiempo para resolver las tareas de estructuración, como administrar bibliotecas y directorios, instalar software de conectividad, establecer nombres similares en Inglés y precalcular "campos de datos virtuales".

Una vez que se han creado las pantallas SQL, puede necesitar desarrollar un conjunto de consultas y reportes estándares, aunque algunos productos ofrecen librerías de plantillas prediseñadas y reportes predefinidos que se pueden modificar rápidamente.

PRODUCTO	EMPRESA DISTRIBUIDORA
Access	Microsoft
Access+	Sonetics
Actuate Reporting System	Actuate Software Corporation
AMIS Information Server	Hoskyns Group plc
Application System	IBM
Approach	Lotus Corporation
ARPEGGIO	Wall Data Inc.
APTuser	International Software Group
AS/Access for Microsoft Access	Martin Spencer & Associates
ASK Joe	Information Management Services
aXcess/400	Glenbrook Software
BrioQuery	Brio Technology
Business Objects	Business Objects, Inc.
Clear: Access	Sterling Software
Crystal Reports, Crystal Info	Seagate Software
d.b. Express	Computer Concepts Corp.
Databoard, Dataread	SLP Infoware
DataDirect Explorer	Intersolv
DataSite	NetScheme Solutions, Inc.
DB Publisher	Xense Technology Inc.
DbPower	Db-Tech Inc.
Decision Analyzer	Decisión Technology
DECquery, DECdecision	Touch Technologies, Inc.
Discoverer, Discoverer/2000	Oracle Corporation
DS Server, DS Modeler	Interweave
EasyReporter	Speedware Corporation
Eclipse Query/Report	Cornut Informatique
ELF	ELF Software

PRODUCTO	EMPRESA DISTRIBUIDORA
English Wizard	English Wizard
EnQuiry	Progress Software
Esperant	Speedware
FOCUS Six	Information Builders, Inc.
4S-Report	Four Seasons Software, Inc
Freequery	Dimension Software Systems
Front & Center for Reporting, Nomad	Thomson Software Products
GQL	Andyne
HarborLight	Harbor Software
HP Information Access	Hewlett-Packard
if…	Leep Technology, Inc.
Impress, SqlBuddy	Objective Technologies, Inc.
Impromptu	Cognos Corporation
InfoAssistant	Asymetrix
InfoMaker	Powersoft Corporation
InfoQuery	Platinum Technology, Inc.
InfoReports	Platinum Technology, Inc.
InformEnt Warehouse Desktop	Fiserv
Internet DataSpot	DTL Data Technologies Ltd.
inSight	Williams & Partner
Interactive Query	New Generation software
IQ/Objects, IQ/SmartServer	IQ Software Corporation
Iridon Panorama	The Great Elk Company Limited
Kinetix	Hilco Technologies
LANSA/Client	LANSA USA
MARKIS/400	AS Software
Nirvana	Synergy Technologies
OR-REPORTER II	Output Reporting, Inc.
Oracle Reports, Browser	Oracle Corporation
Paradox	Borland
Platinum Report Facility	Platinum Technology, Inc
ProBit	System Builder
Productivity Series Reports	michaels, ross & cole
QBE Vision	Sysdeco
QMF	IBM
QueryObject	Cross/Z International, Inc.
Quest	Centura Software Corporation
R&R Report Writer	Concentric Data Systems
Report Writer	Raima
Reportoire	Synergistic Systems, Inc.
Reports	Nine to Five software Co.
ReporTool	Zen Software
ReportSmith	Borland
Rocket Shuttle	Rocket Software, Inc.
Safari ReportWriter	Interactive Software Systems
Sagent Data Mart Solution	Sagent Technology, Inc.
SAS System	SAS Institute
Second Wind	Anju Technologies
Select!	Attachmate
SEQUEL	Advanced Systems Concepts

PRODUCTO	EMPRESA DISTRIBUIDORA
Snow Report Writer	Snow International Corporation
Spectrum Writer	Pacific Systems Group
SQLPRO Agent	Beacon Ware, Inc.
SQR Workbench	MITI
Star Tracker	Leep Technology, Inc.
Strategy	ShowCase Corporation
The Reporter	Sea Change Systems, Inc
Unique XTRA	Unique AS
URSA InfoSuite	Decision Support Inc.
ViewPoint	Informix
ViewPoint	Soliton Associates
Viper	Brann Software
VisPro/Reports	Hock Ware
Visual Cyberquery	Cyberscience Corporation
Visual Dbase	Borland
Visual Express	Computer Associates International
Visual FoxPro	Microsoft Corporation
Visual Net	CNet Svenska AB
Visualizer Query, Charts	IBM
Voyant	Brossco Systems
WebBiz	Cybercom Partners
WebSeQueL	InfoSpace Inc.
WinQL	Data Access Corporation
Xentis	GrayMatter Software Corporation

Tabla 6. Herramientas de consulta y reporte

1.18.2. HERRAMIENTAS DE BASE DE DATOS MULTIDIMENSIONALES / OLAP

Los generadores de reporte tienen sus limitaciones cuando los usuarios finales necesitan más que una sola, una vista estática de los datos, que no sean sujeto de otras manipulaciones. Para estos usuarios, las herramientas del procesamiento analítico en línea (OLAP - On Line Analytical Processing), proveen capacidades "Slide y Dice" que contestaría "¿qué sucedió?" al analizar por qué los resultados están como están.

Las primeras soluciones OLAP estuvieron basadas en bases de datos multidimensionales (MDDBS). Un cubo estructural (dos veces un hipercubo o un arreglo multidimensional) almacenaba los datos para que se puedan manipular intuitivamente y claramente ver las asociaciones a través de dimensiones múltiples. Los productos pioneros tal como Essbase de Arbor Software soportan directamente las diferentes vistas y las manipulaciones dimensionales requeridas por OLAP.

Limitaciones del enfoque de bases de datos multidimensionales:

1. Las nuevas estructuras de almacenamiento de datos requieren bases de datos propietarias. No hay realmente estándares disponibles para acceder a los datos multidimensionales.

   Los proveedores como Arbor, vieron ésto como una oportunidad para crear de facto normas para editar MDDB APIs, propiciando herramientas terceristas y estableciendo asociaciones estratégicas.

   Muchas de estas herramientas de consulta y de soluciones data-mining soportan directamente Essbase, Oracle Express y otros formatos MDDB comunes. El Commander OLAP, herramienta cliente/servidor de Comshare, se sitúa sobre la parte superior de un data warehouse multidimensional Essbase y soporta el acceso dinámico y la manipulación de los datos.

2. La segunda limitación de un MDDB concierne al desarrollo de una estructura de datos. Las compañías generalmente almacenan los datos de la empresa en bases de datos relacionales, lo que significa que alguien tiene que extraer, transformar y cargar estos datos en el hipercubo.

   Este proceso puede ser complejo y consumidor de tiempo pero, nuevamente, los proveedores están investigando la forma de solucionarlos. Las herramientas de extracción de datos y otras automatizan el proceso, trazando campos relacionales en la estructura multidimensional y desarrollando el MDDB sobre la marcha.

   Algunos proveedores ofrecen ahora la técnica OLAP relacional (Relational On Line Analytical Processing - ROLAP), que explora y opera en el data warehouse directamente usando llamadas SQL estándares. Las herramientas de pantallas permiten retener los pedidos multidimensionales, pero el motor ROLAP transforma las consultas en rutinas SQL. Entonces se recibe los resultados tabulados como una hoja de cálculos multidimensional o en alguna otra forma que soporte rotación, drilling down y reducción.

Así como la extracción de los datos, el desarrollo y evolución de la estructura MDDB puede cambiarse. Los administradores ROLAP deben afrontar algunas veces las tareas (agobiantes) de desarrollar las rutinas SQL para agregar e indexar los datos ROLAP, así como, asegurar la traducción correcta de los pedidos multidimensionales en la ventana de comandos SQL.

Los defensores de ROLAP argumentan que se usan estándares abiertos (SQL) y que se esquematiza (nivel de detalle) los datos para hacerlos más fácilmente accesibles. Por otra parte, argumentan que una estructura multidimensional nativa logra mejor performance y flexibilidad, una vez que se desarrolla el almacén de los datos.

Lo bueno es que estas tecnologías evolucionan rápidamente y/o pueden proveer una pronta solución OLAP. Algunos productos ejemplos son PowerPlay de Cognos, Business Objects con el software del mismo nombre, Brio Query de Brio Technology y una serie de DSS Agent/DSS Server de MicroStrategy.

Los retos administrativos y de desarrollo de OLAP, a diferencia de las encontradas con las herramientas de consulta y reporte, son generalmente más complejos. Definiendo el OLAP y el software de acceso a los datos, se requiere un claro entendimiento de los modelos de datos de la corporación y las funciones analíticas requeridas por ejecutivos, gerentes y otros analistas de datos.

El desarrollo de productos comerciales pueden aminorar los problemas, pero OLAP es raramente una solución clave. La arquitectura debe permitir el soporte a su fuente de datos y requerimientos. Pero una vez que se ha establecido un sistema OLAP, el soporte al usuario final será mínimo.

Los usuarios de estos productos deben decidir sobre si los datos del procesamiento analítico en línea, deberían almacenarse en bases de datos multidimensionales especialmente diseñadas o en bases de datos relacionales. Esto depende de las necesidades de la organización. En la Tabla 7, se indica si un producto almacena datos en bases de datos relacionales o en una base de datos multidimensional (MDDB).

PRODUCTO	EMPRESA DISTRIBUIDORA	TIPO
Acuity ES	Acuity Management Systems Ltd.	MDDB
Acumate ES	Kenan Systems Corporation	MDDB
Advance For Windows	Lighten, Inc.	MDDB
AMIS OLAP Server	Hoskyns Group plc	MDDB
BrioQuery	Brio Technology	MDDB
Business Objects	Business Objects, Inc.	Relacional
Commander OLAP, Decision, Prism	Comshare Inc.	MDDB
Control	KCI Computing	Relacional
CrossTarget	Dimensional Insight	MDDB
Cube-It	FICS Group	MDDB
Dataman	SLP Infoware	MDDB
DataTracker	Silvon Software, Inc.	Relacional
DecisionSuite	Information Advantage, Inc.	Relacional
Delta Solutions	MIS AG	MDDB
Demon for Windows	Data Command Limited	MDDB
DSS Agent	MicroStrategy	Relacional
DynamicCube.OCX	Data Dynamics, Ltd.	Relacional
EKS/Empower	Metapraxis, Inc.	MDDB
Essbase Analysis Server	Arbor Software Corporation	MDDB
Essbase/400	ShowCase Corporation	MDDB
Express Server, Objects	Oracle	MDDB
Fiscal	Lingo Computer Design, Inc.	Relacional
Fusion	Information Builders, Inc.	MDDB
FYI Planner	Think Systems	MDDB
Gentia	Planning Sciences	MDDB
Helm	Codeworks	MDDB
Holos	Holistic Systems	MDDB
Hyperion OLAP	Hyperion Software	MDDB
InfoBeacon	Platinum technology, Inc.	Relacional
Informer	Reportech	MDDB/Relacional
Intelligent Decision Server	IBM	Relacional
IQ/Vision	IQ Software Corporation	Relacional
Khalix	Longview Solutions, Inc.	Relacional
Lightship	Pilot Software, Inc.	MDDB
Matryx	Stone, Timber, River	MDDB
MDDB Server	SAS	Relacional
Media	Speedware Corporation	MDDB
Metacube	Informix	Relacional
MIKSolution	MIK	MDDB
MIT/400	SAMAC, Inc	MDDB
MSM	Micronetics Design Corporation	MDDB
Muse	OCCAM Research Corp.	MDDB
OLAP Office	Graphitti Software GmbH	MDDB
OpenOLAP	Inphase Software Limited	Relacional
Pablo	Andyne	MDDB/Relacional
ParaScope	DataVista	Relacional
PowerPlay	Cognos Corporation	MDDB/Relacional
StarTrieve	SelectStar	Relacional
The Ant Colony	Geppetto's Workshop LLC	Relacional
TM/1	Applix	MDDB

PRODUCTO	EMPRESA DISTRIBUIDORA	TIPO
Toto	Ambit Research Ltd.	MDDB
Track for OLAP	Track Business Solutions	MDDB
Visualizer Plans for OS/2	IBM	MDDB

Tabla 7. Herramientas de base de datos multidimensional/OLAP

1.18.3. SISTEMAS DE INFORMACIÓN EJECUTIVOS

Las herramientas de sistemas de información ejecutivos (Executive Information Systems - EIS), proporcionan medios sumamente fáciles de usar para consulta y análisis de la información confiable. Generalmente se diseñan para el usuario que necesita conseguir los datos rápidamente, pero quiere utilizar el menor tiempo posible para comprender el uso de la herramienta.

También, permiten a los desarrolladores de sistemas colocar el contexto del negocio alrededor de información diversa. Un uso típico de un EIS es facilitar al usuario la recuperación y análisis de la métricas, de performance de la organización.

El precio de esta facilidad de uso es que por lo general existen algunas limitaciones sobre las capacidades analíticas disponibles con el sistema de información ejecutivo. Además, muchas de las herramientas de consulta/reporte y OLAP/multidimensional, pueden usarse para desarrollar sistemas de información ejecutivos.

El concepto de sistema de información ejecutivo es simple: los ejecutivos no tienen mucho tiempo, ni la habilidad en muchos casos, para efectuar el análisis de grandes volúmenes de datos. El EIS presenta vistas de los datos simplificados, altamente consolidados y mayormente estáticas.

Categorías de Ambientes EIS:

1. El libro electrónico es una versión en línea, electrónica, contraparte del papel que muchos ejecutivos usan en reuniones con el personal. Las diapositivas electrónicas presentan una

visión concreta de una iniciativa organizacional o quizás los datos para dar a conocer la situación actual de un proyecto importante.

2. El centro de comando es básicamente una colección de puertos en un amplio conjunto de reportes, el newsgroup recupera desde Internet y otros materiales que proveen conocimientos en la organización.

Los reportes del centro de comando pueden ser accesados diariamente o con más frecuencia, si la información cambia constantemente o sólo cuando se garantiza las excepciones. Algunos productos generan alarmas cuando ocurren las excepciones especificadas.

Cuando sea apropiado, cada diapositiva del libro electrónico o pantalla del centro de comando, debería permitir al ejecutivo recibir información adicional si lo desea (y si está disponible). A diferencia del modelo OLAP, donde el incremento de niveles de información se dan a conocer tal como el analista manipula los datos, un ejecutivo espera una descripción global. No deberían escudriñar para obtener respuestas.

Por ello, cuando los ejecutivos piden más información desde las diapositivas del libro electrónico o de las pantallas del centro de comandos, la presentación debería ser cuidadosamente elaborada para presentar principalmente información adicional amplificada. El ejecutivo debe ser capaz de pasar cada punto para "más información", sin perder alguna información crítica.

Los ejecutivos pueden administrar su propio libro electrónico y centro de comandos o los administradores pueden mantener y modificar el EIS de acuerdo a las especificaciones del ejecutivo. Los sistemas de información ejecutivos, generalmente tienen una programación que variará en complejidad de un producto a otro. Los pioneros en el mercado de EIS incluyen Comshare, creadores del Commander EIS y Pilot Software, desarrolladores del Pilot Command Center.

En la Tabla 8, se incluye una relación de productos y empresas que brindan herramientas de Sistemas de Información Ejecutivos.

PRODUCTO	EMPRESA DISTRIBUIDORA	TIPO
Acuity/ES	Acuity Management Systems Limited	1
Applixware	Applix	1
BusinessMetrics	Valstar Systems Ltd.	1
BOARD	Pragma Inform	1
COINS	Russell Consulting Limited	1
ColumbusEIS	Jitcons YO	1
Commander EIS	Comshare Inc.	1
Corporate Management/ Financial Executive Information System	Strategic Information Associates, Inc.	1
CorVu	CorVu Pty Ltd.	1
Decision Suite	Softkit	1
Discovery EIS	Atlantic Information Systems Ltd.	1
EIS	Inphase Software Limited	1
Electronic Balanced Scorecard	ASI Financial Services	1
Enterprise Periscope	Everyware Development Corp.	1
Eureka	European Management Systems	1
ExecuSense	TLG Corporation	1
FOCUS EIS	Information Builders, Inc.	1
Forest & Trees	Platinum Technologies, Inc.	1
iMonitor	BayStone Software	1
InfoManager	Ferguson Information Systems	1
Iridon Almanac	The Great Elk Company Limited	1
inSight	Arcplan Information Services	2
LEADER	Sterling Strategic Solutions	1
MagnaFORUM	Forum Systems, Inc.	1
Merit	GIST, s.r.o.	1
Open EIS Pak	Microsoft	1
Panorama Business Views	Panorama Business Views Inc.	1
Perspectives	Syntell	1
Qbit	Zenia Software, Inc.	1
Reveal	CSD Software Inc.	1
SAS System	SAS Institute	1
Show Business EIS	Show Business Software	1
Tiler EIS++	Avoca Systems Limited	1
Track	Track Business Solutions	1
Traffic Control EIS	Research & Planning, Inc.	3
VentoMap, VentoSales	Vento Software Inc.	1
Virtual Headquarters Management System	vHQ LLC	1
Visual EIS	Synergistic Software	1
Visual Publisher	KMA Associates International, Inc	1
VITAL	Braintec Corporation	1
Wingz	Investment Intelligence Systems Group	1
Wired for OLAP	AppSource Corporation	1
Xecutive Pulse EIS	Megatrend Systems, Ltd.	1

Tabla 8. Sistemas de Información Ejecutivos

Tipo 1. Proporciona un sistema de información ejecutivo con capacidades analíticas.
Tipo 2. Proporciona un sistema de información ejecutivo con capacidades analíticas para usuarios SAP R/3.
Tipo 3. Proporciona un sistema de información ejecutivo con capacidades analíticas para usuarios SAP R/2 y R/3.

1.18.4. HERRAMIENTAS DATA MINING

Data mining es una categoría de herramientas de análisis open-end. En lugar de hacer preguntas, se toma estas herramientas y se pregunta algo "interesante", una tendencia o una agrupación peculiar, por ejemplo. El proceso de data mining extrae los conocimientos guardados o información predictiva desde el data warehouse sin requerir pedidos o preguntas específicas.

Las herramientas Mining usan algunas de las técnicas de computación más avanzadas para generar modelos y asociaciones. Técnicas como:

- Redes neurales
- Detección de desviación
- Modelamiento predictivo
- Programación genética

Mining es un dato-conducido, no una aplicación-conducida.

El Intelligent Miner de IBM para AIX soporta sofisticadas técnicas mining, así como las funciones de preparación de los datos para extraer información desde bases de datos Oracle o Sybase y cargarlos en DB2 para mining. Con su opción Data Mine para el motor Red Brick Warehouse 5.0, Red Brick integra la funcionalidad de un data mining y la arquitectura de almacenamiento.

Otros ejemplos de herramientas data mining comerciales incluyen Darwin de Thinking Machines, herramientas de visualización de datos en MDDB de SAS Institute, SGI MineSet y Focus 6 Serie de Visualización y Análisis de Information Builders.

1.18.5. SISTEMAS DE GESTIÓN DE BASES DE DATOS

Estos software proporcionan procesamiento en paralelo y/o algo fuera de los aspectos ordinarios, que puedan ser especialmente interesantes para la gente de desarrollo de data warehouse y de sistemas de soporte de decisiones.

En la Tabla 9 se incluye una relación de Bases de Datos usados para Data Warehouse.

PRODUCTO	EMPRESA DISTRIBUIDORA
Adabas D	Software AG
Advanced Pick	Pick Systems
DB2	IBM
Fast-Count DBMS	MegaPlex Software
HOPS	HOPS International
Microsoft SQL Server	Microsoft
Model 204	Computer Corporation of America
NonStop SQL	Tandem
Nucleus Server	Sand Technology Systems
OnLine Dynamic Server, Extended Parallel Server	Informix
OpenIngres	Computer Associates
Oracle Server	Oracle
Rdb	Oracle
Red Brick Warehouse	Red Brick Systems
SAS System	SAS
Sybase IQ	Sybase
Sybase SQL Server, SQL Server MPP	Sybase
SymfoWARE	Fujitsu
Teradata DBS	NCR
THOR	Hitachi
Time Machine	Data Management Technologies, Inc.
Titanium	Micro Data Base Systems, Inc.
Unidata	Unidata, Inc.
UniVerse	VMARK
Vision	Innovative Systems Techniques, Inc.
WX9000	White Cross Systems Inc.
XDB Server	XDB Systems, Inc.

Tabla 9. Bases de datos usadas para Data warehouse

1.18.6. ELECCIÓN DE HERRAMIENTAS

Hay algunas reglas obvias a seguir cuando se eligen herramientas de análisis. Las herramientas se combinan según las necesidades de los usuarios finales, capacidad técnica empresarial y la fuente de datos existente.

1. Si se elige un proveedor de depósito que además ofrece herramientas integradas, probablemente se ahorrará un tiempo de desarrollo significativo al elegir un conjunto de herramientas compatibles.
   De otro modo, seleccione un conjunto de herramientas que soporte su fuente de datos original. Sin ese soporte, se debería optar por una solución OLAP relacional debido a que provee una arquitectura abierta.

2. Después que se ha seleccionado un conjunto de herramientas compatible con su fuente de datos, determine cuánto análisis necesita realmente.

   - Si usted simplemente necesita saber "cuánto" o "cuántos", será suficiente una herramienta básica de consultas y reportes.
   - Si usted requiere un análisis más avanzado que explique la causa y los efectos de las ocurrencias y las tendencias, busque una solución OLAP.
   - Las herramientas data mining sofisticadas requieren expertos en técnicas de análisis de datos y se necesitan para pronósticos avanzados, clasificación y creación del modelo.

3. Como con cualquier tecnología, para el mejor desempeño de su compañía, se puede optar por una solución única o un conjunto de soluciones. Su personal debe comprender los requerimientos de tecnología, desarrollar soluciones que reúnan esos requerimientos y mantener y mejorar efectivamente los sistemas.

Los software's de negocio inteligentes son sólo herramientas. Todavía se necesita gerentes y ejecutivos que capten los conocimientos derivados y tomen decisiones intuitivamente. En otras palabras, estos software's requieren todavía inteligencia comercial propia.

En la Tabla 10 se definen los parámetros a tener en cuenta para la elección de las herramientas adecuadas.

Tipo de Herramienta	Pregunta básica	Modelo de Salida	Usuario típico
Consulta y Reporte	¿Qué sucedió?	Reportes de ventas mensuales; histórico de inventario	Necesita data histórica puede tener aptitud técnica limitada
Procesamiento analítico en línea (OLAP)	¿Qué sucedió y por qué?	Ventas mensuales vs. Cambios de precio de los competidores	Necesita ir de una visión estática de los datos a "slicing and dicing" técnicamente astuto
Sistema de Información Ejecutiva (SIE)	¿Qué necesito conocer ahora?	Libros electrónicos; Centros de comandos	Necesita información resumida o de alto nivel puede no ser técnicamente astuto
Data mining	¿Qué es interesante? ¿Qué podría pasar?	Modelos predictivos	Necesita extraer la relación y tendencias de la data ininteligible técnicamente astuto.

Tabla 10. Elección adecuada de herramientas

2. DATAMINING

2.1. FUNDAMENTOS DE DATAMINING

Las técnicas de Datamining son el resultado de un largo proceso de investigación y desarrollo de productos. Esta evolución comenzó cuando los datos de negocios fueron almacenados por primera vez en computadoras, y continúo con mejoras en el acceso a los datos, y más recientemente con tecnologías generadas para permitir a los usuarios navegar a través de los datos en tiempo real.

La Minería de Datos toma este proceso de evolución más allá del acceso y navegación retrospectiva de los datos, hacia la entrega de información prospectiva y preactiva. Minería de Datos está listo para su aplicación en la comunidad de negocios porque está soportado por tres tecnologías que están suficiente maduras:

- Recolección masiva de datos
- Potentes computadoras con multiprocesadores
- Técnicas de Minería de Datos

Las bases de datos comerciales están creciendo a un ritmo sin precedentes. Un reciente estudio sobre los proyectos de Data Warehouse encontró que el 19% de los que contestaron están por encima del nivel de los 50 Gigabytes, mientras que el 59% espera alcanzarlo en un periodo de tres meses. Esta investigación fue realizada en el año 1997, esto nos lleva a pensar en la magnitud de las bases de datos de hoy. En algunas industrias, tales como las de ventas al por menor, estos números pueden ser aún mayores. MCI Telecomunications Corp. cuenta con una base de datos de 3 terabytes + 1 terabyte de índices y cabecera. La necesidad paralela de motores computacionales mejorados puede ahora alcanzarse de forma menos costosa y más efectiva con tecnología de computadores con multiprocesamiento paralelo. Las técnicas de Minería de Datos han existido por lo menos desde hace 10 años, pero solo han sido implementadas recientemente

como herramientas maduras, confiables, entendibles que consistentemente son más eficientes que los métodos estadísticos clásicos.

En la transformación de los datos de negocios a información de negocios, cada nuevo paso se basa en el anterior. Por ejemplo, el acceso a datos dinámicos es crítico para las aplicaciones de navegación de datos (drill trough applications), y la habilidad para almacenar grandes bases de datos es crítica para la Minería de Datos.

Los componentes esenciales de la tecnología de Minería de Datos han estado bajo desarrollo por décadas, en áreas de investigación como estadísticas, inteligencia artificial y aprendizaje de máquinas. Hoy, la madurez de estas técnicas, junto con los motores de bases de datos relacionales de alta performance, hicieron que estas tecnologías fueran prácticas para los entornos de Data Warehouse actuales.

2.2. ALCANCE DE DATAMINING

El nombre de Minería de Datos deriva de las similitudes entre buscar valiosa información en grandes bases de datos y minar una montaña para encontrar una veta de metales valiosos. Ambos procesos requieren examinar una inmensa cantidad de material, o investigar inteligentemente hasta encontrar exactamente donde residen los valores. Dadas bases de datos de suficiente tamaño y calidad, la tecnología de Minería de Datos puede generar nuevas oportunidades de negocios al proveer estas capacidades:

- *Predicción automatizada de tendencias y comportamientos*. La Minería de Datos automatiza el proceso de encontrar información predecible en grandes bases de datos. Preguntas que tradicionalmente requerían un intenso análisis manual, ahora pueden ser contestadas directa y rápidamente desde los datos. Un típico ejemplo de problema predecible es el marketing apuntado a objetivos (trgeted marketing). Por ejemplo, problemas que incluyen pronósticos de problemas financieros e identificar segmentos de población que probablemente respondan similarmente a eventos dados.

- *Descubrimiento automatizado de modelos previamente desconocidos*. Las herramientas de Minería de Datos barren las bases de datos e identifican modelos en un solo paso. Otros problemas incluyen detectar transacciones fraudulentas por ejemplo de tarjetas de crédito e identificar datos anormales que pueden representar errores de tipeado en la carga de datos.

2.3. DEFINICIONES DE DATAMINING

2.3.1. DEFINICIONES DE DATA MINING SEGÚN DIFERENTES AUTORES

El término Minería de Datos o Datamining se emplea a menudo para designar el conjunto de herramientas que permiten al usuario acceder a los datos de la empresa, especialmente los datos históricos, descubriendo modelos implícitos en ellos y analizarlos.

La extracción de información menos estudiada y predecible de grandes colecciones de datos. Una poderosa nueva tecnología con gran potencial para ayudar a las compañías a concentrarse en la información más importante de sus bases de información (Data Warehouse)…

Las herramientas de Minería de Datos predicen futuras tendencias y comportamientos, permitiendo en los negocios tomar decisiones preactivas y conducidas por un conocimiento obtenido a partir de su información de negocio…

La Minería de Datos es un proceso que, a través del descubrimiento y cuantificación de relaciones predictivas en los datos, permite transformar la información disponible en conocimiento útil. Constituye una de las vías principales de explotación del Data Warehouse.

La Minería de Datos se refiere al proceso de extraer conocimiento de bases de datos. Su objetivo es descubrir situaciones anómalas y/o interesantes, tendencias, patrones y secuencias en los datos…

El servicio de Minería de Datos ofrece una amplia gama de técnicas de exploración de información automatizada que permite al analista de la organización descubrir perfiles escondidos en los datos…

La Minería de Datos es una herramienta que ayuda a descubrir patrones y relaciones que puedan pasar desapercibidos en el análisis del negocio…

"Minería de Datos es el proceso de descubrir nuevas correlaciones significantes, modelos y tendencias examinando a través de grandes cantidades de datos guardados en almacenes y usa tecnologías de reconocimiento de modelo así como técnicas estadísticas y matemáticas".

Esta última definición no se limita a la comprobación de una simple hipótesis. Las metodologías incluyen reconocimiento del modelo junto con los métodos estadísticos y matemáticos ampliamente definidos. La gran cantidad de datos y tecnologías (reconocimiento del modelo) extienden la Minería de Datos más allá de las fronteras tradicionales del análisis estadístico.

2.3.2. DEFINICIONES AMPLIAS Y REDUCIDAS

Como en muchos campos que están surgiendo, existen varias definiciones para la Minería de Datos. Como por ejemplo, definiciones que contrastan, amplias y reducidas. La Definición amplia incluye métodos estadísticos tradicionales lo que implica la veracidad de la siguiente frase: "Somos todos mineros de datos".

La Definición reducida da énfasis a Métodos Automatizados y Heurísticos. Estas definiciones restringen a la búsqueda automatizada y al descubrimiento de métodos realizados en grandes bases de datos. También, el uso de métodos heurísticos de Minería de Datos, es un factor distinguible ya que proporcionan respuestas que no pueden prácticamente ser obtenidas.

Finalmente de la misma manera que la minería obtiene el metal, que necesariamente debe tener un mercado; el producto que se genera del procesamiento de la información mediante el análisis

de la Minería de Datos es el conocimiento. Este es hoy en día la principal arma de competitividad de las organizaciones.

2.4. BENEFICIOS CLAVE DEL USO DE MINERÍA DE DATOS

1. Contribuye a la toma de decisiones tácticas y estratégicas proporcionando un sentido automatizado para identificar información clave desde volúmenes de datos generados por proceso tradicionales de E-Business.

2. Permite a los usuarios dar prioridad a decisiones y acciones mostrando factores que tienen una mayor relevancia en un objetivo.

3. Proporciona poderes de decisión a los usuarios del negocio que mejor entienden el problema y el entrono y es capaz de medir las acciones y los resultados de la mejor forma.

4. Modelos descriptivos: En un contexto de objetivos definidos en los negocios permite a la Minería de Datos a organizaciones, sin tener en cuenta la industria o el tamaño, obtener soluciones que impactan en los resultados finales de la cuenta de resultados.

5. Modelos predictivos: Permiten que relaciones no descubiertas e identificadas a través del proceso de Minería de Datos sean expresadas como reglas de negocio o modelos predictivos.

2.5. ARQUITECTURA DE LA MINERÍA DE DATOS

Para integrar la tecnología de Minería de Datos se necesita una cierta arquitectura. En el nivel inferior tenemos las comunicaciones y el soporte del sistema. Luego tenemos el middleware, seguido de la gestión de datos y el Data Warehouse. Luego las diversas tecnologías de Minería de Datos y finalmente los sistemas de visualización y descubrimiento de conocimiento.

Aspectos Funcionales

Los datos son críticos para la Minería de Datos, el primer requisito para que puedan ser analizados es que encuentren estructurados en bases de datos gestionados eficientemente para que puedan ser analizados. Estos sistemas pueden ser bases de datos relacionales, de entidad relación, orientadas a objetos, lógicas o sistemas de almacenamiento de datos (data warehousing systems).

Muchos expertos están de acuerdo en que el 80% de un proceso de Minería de Datos implica preparación de datos.

El tipo de modelo de datos utilizado en la base de datos tiene un impacto en la Minería de Datos. La mayoría de los datos están guardados en bases de datos relacionales y consecuentemente las herramientas de Minería de Datos se han desarrollado para ser aplicadas a este tipo de datos.

Actualmente cada vez más datos están siendo guardados en bases de datos no relacionales como orientados a objetos, objeto-relacionales y multimedia para los que las herramientas de Minería de Datos no están todavía propiamente desarrolladas. Una de las formas de hacer Minería de Datos de una base de datos orientada a objetos pasaría por la extracción de las relaciones entre objetos y su posterior almacenamiento en una base de datos relacional que luego sería analizada.

La mayoría de los datos que se necesitan para Minería de Datos se encuentran en bases de datos heterogéneas que necesitan ser integradas para su posterior análisis. En este sentido el Data Warehouse es una de las tecnologías claves de gestión de datos para llevar a cabo la Minería de Datos.

Esencialmente, un Data Warehouse organiza los datos efectivamente para que puedan ser analizados por la Minería de Datos. Mientras que el Data Warehouse formatea los datos y los organiza para apoyar a las funciones de gestión, la Minería de Datos intenta extraer información útil además de predecir tendencias en los datos. No obstante un Data Warehouse estructura los datos de forma que facilita la Minería de Datos aunque no necesariamente tener un Data Warehouse significa hacer Minería de Datos. Se puede tener también una herramienta de integración que lleve a cabo funciones de Data Warehouse y Minería de Datos.

2.6. DESCUBRIMIENTO DE CONOCIMIENTO EN BASES DE DATOS (KDD)

Según fallad [fallad 96u], el descubrimiento de conocimiento en bases de datos (KDD) se refiere al proceso de hallazgo de conocimiento en datos y a la aplicación de las técnicas de Minería de Datos. También menciona que la inteligencia artificial y los investigadores de Técnicas de Aprendizaje tienden a usar KDD, mientras que estadísticos, analistas y las personas de sistemas de información hablan de Minería de Datos.

La Minería de Datos se enmarca en el proceso completo de extracción de información conocido como KDD, que se encarga además de la preparación de los datos. No se debe olvidar que de la simple aplicación de técnicas de Minería de Datos solo se obtienen patrones que sirven si se les encuentra significado.

KDD se ha definido como la extracción no trivial de información potencialmente útil a partir de un gran volumen de datos en el cual la información está implícita (aunque no se conoce previamente).

Se trata de interpretar grandes cantidades de datos y encontrar relaciones o patrones. Para conseguirlo harán falta técnicas de aprendizaje [Machina Learning], estadísticas y bases de datos.

Bajo sus convenciones, el proceso de descubrimiento del conocimiento toma los resultados tal como vienen de los datos (proceso de extraer tendencias o modelos de los datos) cuidadosamente y con precisión los transforma en información útil y entendible. Esta información no es típicamente

recuperable por las técnicas normales pero es descubierta a través del uso de técnicas de Inteligencia Artificial.

Los datos recogen un conjunto de hechos (una base de datos) y los patrones son expresiones que describen un subconjunto de los datos (un modelo aplicable a ese subconjunto). KDD involucra un proceso iterativo e interactivo de búsqueda de modelos, patrones o parámetros al igual que la Minería de Datos. Los patrones descubiertos han de ser válidos, novedosos para el sistema (para el usuario siempre que sea posible) y potencialmente útiles.

Se han de definir medidas cuantitativas para los patrones obtenidos (precisión, utilidad, beneficio obtenido). Se debe establecer alguna medida de interés que considere la validez, utilidad y simplicidad de los patrones obtenidos mediante alguna de las técnicas de Minería de Datos. El objetivo final de todo esto es incorporar el conocimiento obtenido en algún sistema real, tomar decisiones a partir de los resultados alcanzados o, simplemente, registrar la información conseguida y suministrársela a quien esté interesado.

En muchos lugares se han preocupado de recopilar gran cantidad de información de todo tipo. Es fácil digitalizar información, ya no es excesivamente caro almacenarla y, en principio, los datos recogidos creemos que pueden llegar a sernos útiles.

Ha llegado un momento en el que disponemos de tanta información que nos vemos incapaces de sacarle provecho. Los datos tal cual se almacenan no suelen proporcionar beneficios directos. Su valor real reside en la información que podamos extraer de ellos: información que nos ayude a tomar decisiones o a mejorar nuestra comprensión de los fenómenos que nos rodean.

Hasta ahora, los mayores éxitos en Minería de Datos se pueden atribuir directa o indirectamente a avances en bases de datos (un campo en el que los ordenadores superan a los humanos). No obstante, muchos problemas de representación del conocimiento y de reducción de la complejidad de la búsqueda necesaria (usando conocimiento a priori) están aún por resolver. Ahí reside el interés que ha despertado el tema entre investigadores de todo el mundo.

Las técnicas de Minería de Datos, según otros autores, pertenecen a una etapa dentro del proceso completo de KDD e intentan obtener patrones o modelos a partir de los datos recopilados. Decidir

si los modelos obtenidos son útiles o no suele requerir una valoración subjetiva por parte del usuario.

Hoy en día, la cantidad de datos que ha sido recabada en las bases de datos, no está lejos de exceder nuestra habilidad para reducir y analizar los datos sin el uso de técnicas de análisis automatizados. Muchas bases de datos comerciales transaccionales y científicas crecen en una importante proporción. Un solo sistema, SCICAT es una aplicación del estudio astronómico, se espera que exceda tres terabytes de datos en su terminación. El descubrimiento de conocimiento en las bases de datos es el campo que está evolucionando para proporcionar soluciones al análisis automatizado.

2.7. MÉTODOS APLICADOS DE MINERÍA DE DATOS

Pueden emplearse diferentes criterios para clasificar los sistemas de minería de datos:

- *Dependiendo del objetivo para el que se realiza el aprendizaje*, pueden distinguirse sistemas para: clasificación (clasificar datos en clases predefinidas), regresión (función que convierte datos en valores de una función de predicción), agrupamiento de conceptos (búsqueda de conjuntos en los que agrupar los datos), compactación (búsqueda de descripciones más compactas de los datos), modelado de dependencias (dependencias entre las variables de los datos), detección de desviaciones (búsqueda de desviaciones importantes de los datos respecto de valores anteriores o medios), etc.

- *Dependiendo de la tendencia con que se aborde el problema*, se pueden distinguir tres grandes líneas de investigación o paradigmas: sistemas *conexionistas* (redes neuronales), sistemas *evolucionistas* (algoritmos genéticos) y sistemas *simbólicos*.

- *Dependiendo del lenguaje utilizado para representar del conocimiento*, se pueden distinguir: representaciones basadas en la lógica de proposiciones, representaciones basadas en lógica

de predicados de primer orden, representaciones estructuradas, representaciones a través de ejemplos y representaciones no simbólicas como las redes neuronales.

A continuación describiremos con más detalle los diferentes métodos de representación del conocimiento que se emplean en la minería de datos.

2.7.1. REPRESENTACIÓN DEL CONOCIMIENTO

2.7.1.1. Representaciones basadas en la lógica de proposiciones extendida

Los tradicionales sistemas de aprendizaje han utilizado con gran asiduidad, para representar el conocimiento, una extensión de la lógica de proposiciones, denominada lógica "0+" o representación objeto-atributo-valor. Dentro de la misma, pueden englobarse métodos de representación equivalentes como los árboles de decisión, las reglas de producción y las listas de decisión:

2.7.1.1.1 Árboles de decisión

Los *árboles de decisión* son una forma de representación sencilla, muy usada entre los sistemas de aprendizaje supervisado, para clasificar ejemplos en un número finito de clases. Se basan en la partición del conjunto de ejemplos según ciertas condiciones que se aplican a los valores de los atributos. Su potencia descriptiva viene limitada por las condiciones o reglas con las que se divide el conjunto de entrenamiento; por ejemplo, estas reglas pueden ser simplemente relaciones de igualdad entre un atributo y un valor, o relaciones de comparación ("mayor que", etc.), etc.

Los sistemas basados en árboles de decisión forman una familia llamada TDIDT *(Top-Down Induction of Decision Tree*s), cuyo representante más conocido es ID3.

ID3 (Interactive Dichotomizer) se basa en la reducción de la entropía media para seleccionar el atributo que genera cada partición (cada nodo del árbol), seleccionando aquél con el que la reducción es máxima. Los nodos del árbol están etiquetados con nombres de atributos, las ramas con los posibles valores del atributo, y las hojas con las diferentes clases. Existen versiones *secuénciales* de ID3, como ID5R.

C4.5, es una variante de ID3, que permite clasificar ejemplos con atributos que toman valores continuos.

El aprendizaje de árboles de decisión es uno de los más sencillos y fáciles de implementar y a su vez de los más poderosos. Un árbol de decisión toma de entrada un objeto o situación descrita por un conjunto de atributos y regresa una decisión ``verdadero/falso''. En general pueden tener un rango más amplio que simples funciones Booleanas, pero por simplicidad, consideremos primero sólo estas.

Cada nodo interno corresponde a una prueba en el valor de uno de los atributos y las ramas están etiquetadas con los posibles valores de la prueba. Cada hoja especifica el valor de la clase.

Expresividad

Los árboles de decisión están limitados a hablar de un solo objeto, osea, son escencialmente proposicionales, siendo cada prueba de atributo una proposición.
Por lo mismo no podemos usar los árboles de decisión para expresar pruebas sobre dos o más objetos diferentes, e.g.

$$\exists r_2 Cercano(r_2, r) \wedge \Pr ecio(r_2, p_2) \wedge \Pr ecio(r, p) \wedge MasBarato(p_{2,} p)$$

Claro que podríamos añadir un atributo Booleano que se llame: *RestMásBaratoCerca*, pero es intratable para todas las combinaciones de atributos.

Por otro lado, los árboles de decisión son completamente expresivos dentro de la clase de lenguajes proposicionales. O sea que cualquier función Booleana puede ser descrita por un árbol de decisión.

Trivialmente, podemos tomar cada fila como un camino en la construcción de un árbol. Sin embargo, la tabla es exponencial en el número de atributos.

Para muchas funciones, los árboles son relativamente pequeños. Sin embargo, para otras funciones puede requerir un árbol exponencialmente grande. Por ejemplo, la función *paridad* (es

decir, regresa 1 si la suma de 1's es par) o la función de *mayoría* (regresa 1 si más de la mitad de la entrada es un 1).

Para *n* atributos, hay 2^n filas. Podemos considerar la salida como una función definida por 2^n bits. Con esto hay 2^{2^n} posibles funciones diferentes para *n* atributos (para 6 atributos, hay **2x10^{19}**).

Por lo mismo, tenemos que usar algún algoritmo ingenioso para encontrar una hipótesis consistente en un espacio de búsqueda tan grande.

Inducción de árboles de decisión a partir de ejemplos

Un ejemplo es descrito por los valores de los atributos y el valor del predicado meta. El valor del predicado meta se le llama la clasificación del ejemplo.

Si el predicado es verdadero, entonces el ejemplo es positivo, sino el ejemplo es negativo.

En caso de existir más clases, los ejemplos de una sola clase son positivos y el resto de los ejemplos son considerados negativos.

Cuando se tiene un conjunto de ejemplos (datos), normalmente se divide aleatoriamente en dos subconjuntos. Uno de entrenamiento (con el cual se construye la hipótesis) y otro de prueba (con el que se prueba la hipótesis encontrada).

Más formalmente:

1. Junte una gran cantidad de ejemplos
2. Divídalos en dos conjuntos disjuntos: entrenamiento y prueba
3. Usa el algoritmo de aprendizaje para generar una hipótesis *H*
4. Mida el porcentaje de clasificación correcta de *H* en el conjunto de prueba
5. Repite los pasos 1 - 4 para diferentes tamaños de conjuntos de entrenamiento y diferentes conjuntos seleccionados aleatoriamente.

Encontrar un árbol puede ser trivial (por ejemplo, construir un camino para cada ejemplo). Sin embargo, no es bueno para predecir casos no vistos. El problema es que sólo memoriza lo visto, por lo que no extrae ningún patrón de los ejemplos (por lo que no podemos esperar que extrapole).

El extraer un patrón significa el poder describir una gran cantidad de ejemplos en forma concisa. Esto también sigue un principio general en los algoritmos de inducción llamada: *Ockham's razor* (muchas veces escrito como Occam): dar preferencia a hipótesis más simples que sean consistentes con todas las observaciones.

Encontrar el árbol más pequeño es intratable, pero se pueden usar heurísticas para encontrar árboles pequeños.

Una buena idea es probar primero el atributo más ``importante'' (el que diferencia mejor los ejemplos).

Después que el primer atributo particiona los ejemplos, cada subconjunto es un nuevo problema de aprendizaje a su vez, con menos ejemplos y un atributo menos. Este proceso recursivo tiene 4 posibles resultados (ver Tabla 11):

1. Si existen ejemplos positivos y negativos, escoge el mejor atributo para particionarlos.
2. Si todos los atributos restantes son positivos (o negativos), termina y regresa True (o False)
3. No quedan ejemplos (no ha sido observado un ejemplo con esa combinación de atributos). Regresa un *default* en base a la clasificación mayoritaria de su nodo padre
4. No hay más atributos, pero seguimos con ejemplos positivos y negativos (i.e., existen ejemplos con la misma descripción, pero diferente clasificación). Posiblemente por ruido y/o falta de atributos y/o dominio no determinístico. Posible solución: tomar la clasificación mayoritaria

El árbol resultante no necesariamente es el ``correcto''. Para eso lo probamos con el conjunto de prueba.

Algoritmo de construcción de árboles de decisión.
función *Arbol-decisón* (ejemplos, atributos, default) regresa un árbol de decisión entradas: ejemplos: conjunto de ejemplos

atributos: conjunto de atributos
default: valor de default para el predicado meta

If ejemplos = vacio then regresa *default*
else if todos los *ejemplos* tienen la misma clasificación
then regresa la clasificación
else if *atributos* = vacio
then regresa VALOR-MAYORITARIO(ejemplos)
Else
Mejor ← ESCOGE-ATRIBUTO(atributos, ejemplos)
Árbol ← nuevo árbol de decisión con Mejor raíz para cada valor v_i de Mejor d_o
ejemplos$_i$ ← {ejemplos con Mejor = v_i}
Subárbol← ÁRBOL-DECISIÓN(ejemplos$_i$, atributos-mejor, VALOR-MAYORITARIO(ejemplos))
Añade una rama a Árbol con etiqueta v_i y subárbol Subárbol
end
return Árbol

Tabla 11. Algoritmo de construcción de árboles de decisión.

Tabla de ejemplos para decidir si jugar o no golf				
Ambiente	**Temp**	**Humedad**	**Viento**	**Clase**
Soleado	alta	alta	no	N
Soleado	alta	alta	si	N
Nublado	alta	alta	no	P
Lluvia	media	alta	no	P
Lluvia	baja	normal	no	N
Lluvia	baja	normal	si	N
Nublado	baja	normal	si	P
Soleado	media	alta	no	N
Soleado	baja	normal	no	P
Lluvia	media	normal	no	P

Soleado	media	normal	si	P
Nublado	media	alta	si	P
Nublado	alta	normal	no	P
Lluvia	media	alta	si	N

Tabla 12. Tabla de ejemplos para decidir si jugar o no golf.

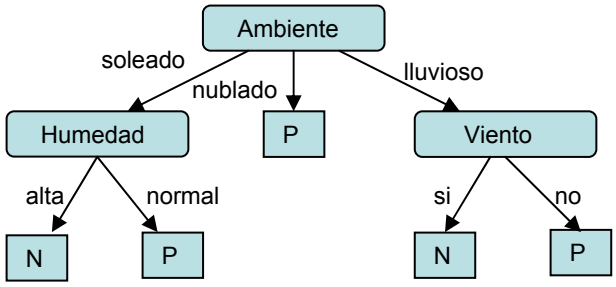

Figura 18. Árbol de decisión para jugar Golf.

Aplicaciones:

Es la técnica que posiblemente se ha usado más en aplicaciones reales. Tres ejemplos:

- GASOIL (1986): Diseño de sistemas de separación de hidrocarburos en plataformas petroleras de BP, 2,800 reglas, 1 año-hombre de tiempo de desarrollo, 0.1 de mantenimiento, mejor que expertos y ahorro de millones de dolares.
- BMT (1990): Congfiguración de equipo de protección de incendios en edificios, > 30,000 reglas, 9 años hombre de desarrollo y 2 de mantenimiento (comparado con: MYCIN: 400 reglas, 100 años-hombre de desarrollo o R1/XCON: 8,000 reglas, 180 años-hombre de desarrollo y 30 de mantenimiento).
- Aprendiendo a volar (1992): En lugar de construir un modelo preciso de la dinámica del sistema, se aprendió un mapeo adecuado entre el estado actual y la decisión de control correcta para volar un Cessna en un simulador de vuelo. Los datos se obtuvieron de 3 pilotos experimentados haciendo un plan de vuelo asignado 30 veces. Cada acción del piloto creaba un ejemplo. Se usaron 90,000 ejemplos descritos por 20 atributos. Se uso C4.5 que genero un árbol y se convirtió a C. Se insertó en el simulador y logro volar. Los

resultados fueron sorprendentes en el sentido de que aparte de aprender a volar a veces tomaba decisiones mejores que las de sus ``maestros''

Cómo le hace?

La medida utilizada en ESCOGE-ATRIBUTO debe de tener su valor máximo cuando el atributo sea perfecto (i.e., discrimine perfectamente ejemplos positivos y negativos) y mínimo cuando el atributo no sea relevante.

Una posibilidad es basar la medida en la cantidad de información que da el atributo (basado en la teoría de Shanon y Weaver '49).

La cantidad de información mide la impureza en una colección arbitraria de ejemplos.

La cantidad de información recibida respecto a la ocurrencia de un evento es inversamente proporcional a la probabilidad de ocurrencia de dicho evento.

La información se mide en bits (un bit de información es suficiente para responder Verdadero/Falso a una pregunta cuya respuesta no se sabe).

Si se tienen v_i posibles respuestas con probabilidades $P(v_i)$, el contenido de información es:

$$I(P(v_1),.....,P(v_n)) = -\sum_{i=1}^{n} P(v_i)\log_2 P(v_i)$$

Nos representa el contenido promedio de información para los diferentes eventos (ver Figura19).

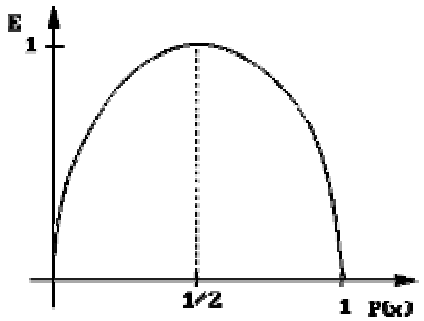

Figure 19. Función de Entropía.

En el caso de los árboles de decisión queremos estimar las probabilidades de las respuestas. Esto se hace por la proporción de ejemplos positivos y negativos.

Si se tienen *p* ejemplos positivos y *n* ejemplos negativos, entonces:

$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Un solo atributo normalmente no nos proporciona toda esta información, pero podemos estimar cuanta, viendo cuanta información necesitamos después de utilizar ese atributo,

Cada atributo *A*, divide a los ejemplos del conjunto de entrenamiento en subconjuntos

$E_1, E_2, \ldots, E_n$ de acuerdo a los *v* valores del atributo.

Cada subconjunto E_i tiene p_i ejemplos positivos y n_i ejemplos negativos, por lo que para cada rama necesitamos:

$I\left(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}\right)$ Cantidad de información para responder a una pregunta.

Un ejemplo aleatorio tiene el valor *i*-ésimo del atributo *A* con probabilidad: $\frac{p_i+n_i}{p+n}$

Por lo que en promedio, después de probar el atributo *A*, necesitamos:

$$E(A) = \sum_{i=1}^{n} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

La cantidad de información que ganamos al seleccionar un atributo está dada por:

$$Ganancia(A) = I(\frac{p}{p+n}, \frac{n}{p+n}) - E(A)$$

La ganancia de *A* me dice el número de bits que ahorramos para responder a la pregunta de la clase de un ejemplo, dado que conocemos el valor del atributo *A*.

Dicho de otra forma, mide que tan bien un atributo separa a los ejemplos de entrenamiento de acuerdo a la clase meta.

La función de evaluación escoge el atributo de mayor ganancia.

Por ejemplo, si calculamos la ganancia para el atributo para los datos de la tabla (asumimos que 0 $log_2(0) = 0$):

$$I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.941$$

Para *Ambiente*:
Soleado: p_1=2,n_1=3,$I(p_1,n_1)$=0.971
Nublado: p_2=4,n_2=0,$I(p_2,n_2)$=0
Lluvia: p_3=3,n_3=2,$I(p_3,n_2)$=0.971

$$Entropía(Ambiente) = \frac{5}{14} I(p_1, n_1) + \frac{4}{14} I(p_2, n_2) + \frac{5}{14} I(p_3, n_3) = 0.694$$

Para *Humedad*:
Alta: p_1=3,n_1=4,$I(p_1,n_1)$=0.985

Normal: p_2=6,n_2=1,$I(p_2,n_2)$=0.592
Entropía (Humedad) = 0.798

Para *Viento*:
No: p_1=6,n_1=2,$I(p_1,n_1)$=0.811
Si: p_2=3,n_2=3,$I(p_2,n_2)$=1.0
Entropía (Viento) = 0.892

Para *Temperatura*, Entropía (Temperatura) = 0.9111

Las ganancias son entonces:
Ganancia (Ambiente) = 0.246 (MAX)
Ganancia (Humedad) = 0.151
Ganancia (Viento) = 0.048
Ganancia (Temperatura) = 0.029

Por lo que ID3 escoge el atributo *Ambiente* como nodo raíz y procede a realizar el mismo proceso con los ejemplos de cada rama.

Para *Ambiente* tenemos tres subconjuntos: soleado (2+,3-), nublado (4+,0-), lluvioso (3+,2-). Para nublado, no tenemos que hacer nada, mas que asignarle la clase *P*.

Por ejemplo, para soleado haríamos el mismo proceso:

Ganancia (Humedad) = 0.97 - [(3/5)0 + (2/5)0] = 0.97 (MAX)
Ganancia (Temperatura) = 0.97 - [(2/5)0 + (2/5)1 + (1/5)0] = 0.570
Ganancia (Viento) = 0.97 - [(2/5)1 + (3/5)0.918] = 0.019

Uso del Árbol de Decisión

Con el árbol construido, podemos preguntar si esta bien jugar el sábado en la mañana con ambiente soleado, temperatura alta, humedad alta y con viento, a lo cual el árbol me responde que no.

ID3 sigue una estrategia *hill-climbing*, sin *backtracking*, incrementando en cada paso la complejidad del árbol. Utiliza todos los ejemplos, con los cuales extrae estadísticas y que lo hace más robusto que un algoritmo incremental y por otro lado lo hace fácilmente extensible para manejar ruido. Tiende a preferir construir árboles pequeños con atributos con ganancia de información alta cerca de la raíz.

Criterio de Selección:

El criterio de selección basado en contenido de información tiende a favorecer atributos que tienen más valores.

Por ejemplo, si un atributo tiene valores aleatorios o es un identificador único de cada ejemplo (su clasificación sería perfecta y su información después al seleccionarlo sería 0 (ganancia máxima). Con esto el algoritmo básico construye un árbol de un solo nivel o *decision stump*.

Posible solución: árbol binario, dividiendo los posibles valores de los atributos en dos. Desventaja: árboles difíciles de entender + computacionalmente caro (2^nsubconjuntos para n valores).

Otra solución: Para compensar esto se definió una razón de ganancia de información. Esto es dividir la ganancia de información entre la información de la división (la cantidad de información en los ejemplos que se dividió).

La información de la división (*split information*) se define como:

$$SI(A) = -\sum_{i=1}^{n} \frac{p_i + n_i}{p+n} \log_2 \frac{p_i + n_i}{p+n}$$

Esto es, la entropía de los datos con respecto a los valores del atributo (versus entropía con respecto a la clase).

E.g., si un atributo binario divide el conjunto de ejemplos en dos subconjuntos de igual tamaño, el contenido de información de su división es 1. Mientras que un atributo que divide los ejemplos en 14 subconjuntos de tamaño 1, sería: 14 (1/14 log_2 (1/14)) = log_2(1/14).

Sin embargo, no siempre funciona ya que puede sobrecompensar. Una práctica común es usar el atributo de la razón de ganancia de información máxima dado que su ganancia de información es al menos tan grande como el promedio de ganancia de información del resto de los atributos.

Ruido y ``Overfitting''

Algunas de las ventajas de ID3 es que es útil en dominios con un alto grado de no homogeneidad (diferentes relaciones entre atributos en diferentes regiones del espacio de problemas) y alta dimensionalidad (muchos atributos).

En general, podemos hablar de que a pesar de que falte información relevante, se pueda construir un árbol con los atributos irrelevantes.

Con muchas posibles hipótesis se tiene que tener cuidado en no encontrar ``regularidades con poco sentido'' a partir de los datos. A este problema se le llama *overfitting* y afecta a todos los tipos de aprendizaje (i.e., no sólo a los árboles de decisión).

Definición: dado un espacio de hipótesis *H*, una hipótesis $h \in H$ se dice que *sobreajusta* los datos de entrenamiento si existe otra hipótesis $h' \in H$, tal que *h* tiene errores más pequeños que *h*' en los ejemplos de entrenamiento, pero *h*' tiene errores más pequeños que *h* en toda la distribución de ejemplos.

Uno de los problemas a los que se enfrentan los sistemas de aprendizaje, y que provocan el sobreajuste, es cuando los ejemplos de entrenamiento contienen ruido:

- valores de atributos erróneos, subjetivos
- clasificación equivocada
- valores desconocidos

Con ruido, se pueden tener dos ejemplos con los mismos valores de atributos, pero clase diferente. En presencia de ruido, el algoritmo básico (ID3) tiende a construir árboles de decisión que son más grandes de lo necesario, y no clasifican adecuadamente.

En el caso de árboles de decisión se tiene que decidir:

- cómo trabajar con atributos inadecuados
- cuándo al añadir atributos extra no mejora la predicción del árbol de decisión

En general, podemos hablar de dos métodos utilizados para manejar ruido (basados en la condición de terminación):

- *pruning* (o *pre-pruning*): cambiar el criterio de paro del árbol de decisión para ``podar'' ramas.
- *post-pruning*: ``podar'' ramas una vez construido el árbol.

2.7.1.1.2 Reglas de producción

Una desventaja de los árboles de decisión es que tienden a ser demasiado grandes en aplicaciones reales y, por tanto, se hacen difíciles de interpretar desde el punto de vista humano. Por ello, se han realizado diversos intentos para convertir los árboles de decisión en otras formas de representación, como las *reglas de producció*n. Aquí consideramos reglas de producción del tipo *si-entonce*s, basadas en lógica de proposiciones. El consecuente es una clase, y el antecedente es una expresión proposicional, que puede estar en forma normal conjuntiva (CNF) o ser simplemente un término.

En 1987 se propone una técnica para construir reglas de decisión, basadas en lógica de proposiciones, a partir de árboles de decisión. El problema es que incluso las reglas de producción así obtenidas pueden resultar demasiado complejas para un experto humano.

Algunos sistemas como PRISM, generan directamente reglas algo más sencillas, sin tener que construir el árbol previamente, mediante un algoritmo parecido a ID3 y con una precisión similar.

La familia AQ la forman sistemas (AQ11, AQ15, etc.) que generan *descripciones estructurale*s, por diferenciarlas de las *descripciones de atributos* de los sistemas anteriormente mencionados. Estas descripciones son también reglas de producción, aunque con mayor capacidad descriptiva, pues su antecedente es una fórmula lógica. La notación utilizada en estas reglas se denomina VL1 *(Variable-valued Logia system* 1), y permite utilizar selectores (igualdad entre un atributo y un valor o conjunto de valores), complejos (conjunciones de selectores) y disyunciones de complejos para construir las reglas de producción.

Los sistemas basados en reglas son los más comúnmente utilizados. Su simplicidad y similitud con el razonamiento humano, han contribuido para su popularidad en diferentes dominios. Las reglas son un importante paradigma de representación del conocimiento.

Las reglas representan el conocimiento utilizando un formato ***SI-ENTONCES*** (***IF-THEN***), es decir tienen 2 partes:

- La parte ***SI*** (***IF***), es el antecedente, premisa, condición o situación; y
- La parte ***ENTONCES*** (***THEN***), es el consecuente, conclusión, acción o respuesta.

Las reglas pueden ser utilizadas para expresar un amplio rango de asociaciones, por ejemplo:

- **SI** está manejando un vehículo **Y** se aproxima una ambulancia, **ENTONCES** baje la velocidad **Y** hágase a un lado para permitir el paso de la ambulancia.
- **SI** su temperatura corporal es de 39 ºC, **ENTONCES** tiene fiebre.
- **SI** el drenaje del lavabo está tapado **Y** la llave de agua está abierta, **ENTONCES** se puede inundar el piso.

Inferencia Basada en Reglas

Una declaración de que algo es verdadero o es un hecho conocido, es una *afirmación* (*fact*). El conjunto de afirmaciones se conoce a menudo con el nombre de *memoria de trabajo o base de afirmaciones*. De igual forma, al conjunto de reglas se lo denomina *base de reglas*.

Un sistema basado en reglas utiliza el *modus ponens* para manipular las afirmaciones y las reglas durante el proceso de inferencia. Mediante técnicas de búsqueda y procesos de unificación, los sistemas basados en reglas automatizan sus métodos de razonamiento y proporcionan una progresión lógica desde los datos iniciales, hasta las conclusiones deseadas. Esta progresión hace que se vayan conociendo nuevos hechos o descubriendo nuevas afirmaciones, a medida que va guiando hacia la solución del problema.

En consecuencia, el proceso de solución de un problema en los sistemas basados en reglas va realizando una serie de inferencias que crean un sendero entre la definición del problema y su solución. Las inferencias están concatenadas y se las realiza en forma progresiva, por lo que se lo que se dice que el proceso de solución origina una *cadena de inferencias*.

Los sistemas basados en reglas difieren de la representación basada en lógica en las siguientes características principales:

- Son en general no-monotónicos, es decir hechos o afirmaciones derivadas, pueden ser retractados, en el momento en que dejen de ser verdaderos.
- Pueden aceptar incertidumbre en el proceso de razonamiento.

El Proceso de Razonamiento

El proceso de razonamiento en un sistema basado en reglas es una progresión desde un conjunto inicial de afirmaciones y reglas hacia una solución, respuesta o conclusión. Como se llega a obtener el resultado, sin embargo, puede variar significativamente:

- Se puede partir considerando todos los datos conocidos y luego ir progresivamente avanzando hacia la solución. Este proceso se lo denomina *guiado por los datos* o de **encadenamiento progresivo** (*forward chainning***)**.
- Se puede seleccionar una posible solución y tratar de probar su validez buscando evidencia que la apoye. Este proceso se denomina *guiado por el objetivo* o de **encadenamiento regresivo** (*backward chainning*).

Razonamiento Progresivo

En el caso del razonamiento progresivo, se empieza a partir de un conjunto de datos colectados a través de observación y se evoluciona hacia una conclusión. Se chequea cada una de las reglas para ver si los datos observados satisfacen las premisas de alguna de las reglas. Si una regla es satisfecha, es ejecutada derivando nuevos hechos que pueden ser utilizados por otras reglas para derivar hechos adicionales. Este proceso de chequear reglas para ver si pueden ser satisfechas se denomina *interpretación de reglas*.

La interpretación de reglas es realizada por una máquina de inferencia en un sistema basado en conocimiento. La interpretación de reglas, o inferencia, en el razonamiento progresivo involucra la repetición de los pasos que se indican en la siguiente figura.

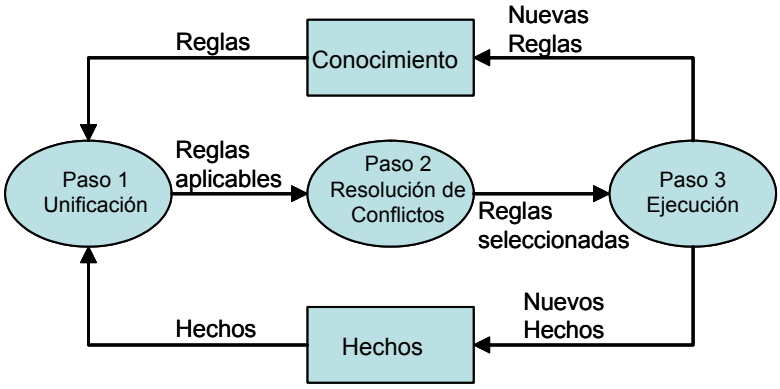

Figura 20. Proceso de Razonamiento Progresivo

1. *Unificación (Matching).-* En este paso, en las reglas en la base de conocimientos se prueban los hechos conocidos al momento para ver cuáles son las que resulten satisfechas. Para decir que una regla ha sido satisfecha, se requiere que todas las premisas o antecedentes de la regla resuelvan a verdadero.

2. *Resolución de Conflictos.-* Es posible que en la fase de unificación resulten satisfechas varias reglas. La resolución de conflictos involucra la selección de la regla que tenga la más alta prioridad de entre el conjunto de reglas que han sido satisfechas.

3. *Ejecución.-* El último paso en la interpretación de reglas es la ejecución de la regla. La ejecución puede dar lugar a uno o dos resultados posibles: nuevo hecho (o hechos) pueden ser derivados y añadidos a la base de hechos, o una nueva regla (o reglas) pueden ser añadidas al conjunto de reglas (base de conocimiento) que el sistema considera para ejecución.

En esta forma, la ejecución de las reglas procede de una manera progresiva (hacia adelante) hacia los objetivos finales.

Un conjunto de aplicaciones adecuadas al razonamiento progresivo incluye supervisión y diagnóstico en sistemas de control de procesos en tiempo real, donde los datos están continuamente siendo adquiridos, modificados y actualizados. Estas aplicaciones tienen 2 importantes características:

1. Necesidad de respuesta rápida a los cambios en los datos de entrada.
2. Existencia de pocas relaciones predeterminadas entre los datos de entrada y las conclusiones derivadas.

Otro conjunto de aplicaciones adecuadas para el razonamiento progresivo está formado por: diseño, planeamiento y calendarización, donde ocurre la síntesis de nuevos hechos basados en las conclusiones de las reglas. En estas aplicaciones hay potencialmente muchas soluciones que pueden ser derivadas de los datos de entrada. Debido a que estas soluciones no pueden ser

enumeradas, las reglas expresan conocimiento como patrones generales y las conexiones precisas entre estas reglas no pueden ser predeterminadas.

Razonamiento Regresivo

El mecanismo de inferencia, o intérprete de reglas para el razonamiento regresivo, difiere significativamente del mecanismo de razonamiento progresivo. Si bien es cierto, ambos procesos involucran el examen y aplicación de reglas, el razonamiento regresivo empieza con la conclusión deseada y decide si los hechos que existen pueden dar lugar a la obtención de un valor para esta conclusión. El razonamiento regresivo sigue un proceso muy similar a la búsqueda primero en profundidad.

El sistema empieza con un conjunto de hechos conocidos que típicamente está vacío. Se proporciona una lista ordenada de objetivos (o conclusiones), para las cuales el sistema trata de derivar valores. El proceso de razonamiento regresivo utiliza esta lista de objetivos para coordinar su búsqueda a través de las reglas de la base de conocimientos. Esta búsqueda consiste de los siguientes pasos:

1. Conformar una pila inicialmente compuesta por todos los objetivos prioritarios definidos en el sistema.
2. Considerar el primer objetivo de la pila. Determinar todas las reglas capaces de satisfacer este objetivo, es decir aquellas que mencionen al objetivo en su conclusión.
3. Para cada una de estas reglas examinar en turno sus antecedentes:
    a. Si todos los antecedentes de la regla son satisfechos (esto es, cada parámetro de la premisa tiene su valor especificado dentro de la base de datos), entonces ejecutar esta regla para derivar sus conclusiones. Debido a que se ha asignado un valor al objetivo actual, removerlo de la pila y retornar al paso (2).
    b. Si alguna premisa de la regla no puede ser satisfecha, buscar reglas que permitan derivar el valor especificado para el parámetro utilizado en esta premisa.
    c. Si en el paso (b) no se puede encontrar una regla para derivar el valor especificado para el parámetro actual, entonces preguntar al usuario por dicho valor y añadirlo a la base de datos. Si este valor satisface la premisa actual entonces continuar con la

siguiente premisa de la regla. Si la premisa no es satisfecha, considerar la siguiente regla.

Si todas las reglas que pueden satisfacer el objetivo actual se han probado y todas no han podido derivar un valor, entonces este objetivo quedará indeterminado. Removerlo de la pila y retornar al paso (2). Si la pila está vacía parar y anunciar que se ha terminado el proceso.

El razonamiento regresivo es mucho más adecuado para aplicaciones que tienen mucho mayor número de entradas, que de soluciones posibles. La habilidad de la lógica regresiva para trazar desde las pocas conclusiones hacia las múltiples entradas la hace más eficiente que el encadenamiento progresivo.

Una excelente aplicación para el razonamiento regresivo es el diagnóstico, donde el usuario dialoga directamente con el sistema basado en conocimiento y proporciona los datos a través del teclado. Problemas de clasificación también son adecuados para ser resuelto mediante el razonamiento regresivo.

Desventajas de las Reglas de Producción

Algunos problemas existen en los sistemas basados en reglas. Estos problemas caen dentro de una de las siguientes categorías: encadenamiento infinito; incorporación de conocimiento nuevo contradictorio, y; modificación de reglas existentes.

Desventajas adicionales pueden ser: ineficiencia (necesidad de modularizar o de introducir metarreglas), opacidad (dificultad de establecer relaciones), adaptación al dominio (rápido crecimiento del número de reglas).

El conocimiento acerca de las reglas de producción se denomina *Metarregla*. Las metarreglas facilitan y aceleran la búsqueda de soluciones.

Ventajas de las Reglas de Producción

A pesar de las desventajas anotadas, los sistemas basados en reglas han permanecido como los esquemas más comúnmente utilizados para la representación del conocimiento. Como ventajas

significativas se pueden mencionar las siguientes: modularidad, uniformidad y naturalidad para expresar el conocimiento.

2.7.1.1.3 Listas de decisión

Las listas de decisión son otra forma de representación basada en lógica de proposiciones. Es una generalización de los árboles de decisión y de representaciones conjuntivas (CNF) y disyuntivas (DNF). Una lista de decisión es una lista de pares de la forma:

$$(d\ 1\ ,\ C\ 1\ ),\ (d\ 2\ ,\ C\ 2\ ),...,\ (d\ n\ ,\ C\ n\ )$$

donde cada d i es una descripción elemental, cada C i es una clase, y la última descripción C n es el valor *verdadero*. La clase de un objeto será C j cuando d j sea la primera descripción que lo satisface. Por tanto, se puede pensar en una lista de decisión como en una regla de la forma "si d 1 entonces C 1 , sino si d 2 ..., sino si d n entonces C n ".

Se usan por ejemplo en el sistema CN2 que es una modificación del sistema AQ, que pretende mejorar el aprendizaje a partir de ejemplos con ruido (al evitar la dependencia de ejemplos específicos e incorporar una poda del espacio de búsqueda).

Las descripciones elementales de los pares que forman la lista de decisión tienen la misma forma que los *complejos* de AQ.

2.7.1.2. Representaciones basadas en la lógica de predicados de primer orden

Aunque las representaciones basadas en lógica de proposiciones han sido usadas con éxito en muchos sistemas de aprendizaje en computadores, tienen algunas importantes limitaciones, superadas por la lógica de predicados de primer orden, que restringen su campo de aplicación:

- *Potencia expresiva*: Las representaciones basadas en lógica de proposiciones limitan la forma de los patrones que pueden ser representados, ya que, en general, no pueden expresar relaciones. Así, por ejemplo, no se pueden representar (al menos de un modo sencillo) patrones en los que se cumpla una relación de igualdad entre dos atributos (sólo se permitiría expresar la igualdad entre un atributo y un valor constante).

- *Conocimiento de base*: Es difícil incorporar conocimiento de base en el proceso de aprendizaje. Una forma sencilla de conocimiento de base la constituyen restricciones impuestas a las descripciones generadas por el sistema, aunque esto puede resultar demasiado restrictivo.

- *Restricciones en el vocabulario*: Las descripciones de los sistemas actuales vienen limitadas por un vocabulario fijo de atributos proposicionales. Podría ser muy útil tener la posibilidad de mejorar la representación mediante la invención de nuevos predicados.

Todas estas limitaciones pueden superarse con una representación del conocimiento más potente: la lógica de predicados de primer orden. Cada vez hay más sistemas de aprendizaje en sistemas que utilizan de algún modo la lógica de primer orden, surgiendo así una nuevo área de interés llamado *programación lógica inductiva* (en inglés *Inductive Logic Programming* o *I*LP). El objetivo de la programación lógica inductiva es construir un programa lógico (en lógica de predicados de primer orden) que, junto al conocimiento que se tenga del dominio, tenga como consecuencia lógica el conjunto de entrenamiento del sistema.

2.7.1.3. Representaciones estructuradas

Las representaciones estructuradas de conocimiento tienen una gran potencia expresiva (aunque, en teoría, no mayor que la de la lógica de predicados de primer orden) y permiten una fácil interpretación del mismo. Entre las representaciones estructuradas se pueden incluir las *redes semánticas* y los *marco*s. En cualquier caso, el conocimiento expresado mediante una de estas

representaciones estructuradas puede ser traducido fácilmente a lógica de predicados de primer orden.

2.7.1.3.1 Redes semánticas

El término de *red semántica* surge a partir del modelo de memoria semántica de Quillian (1968), con el que pretendía representar el significado de los vocablos del idioma inglés. En una red semántica la representación consta de un conjunto de nodos (conceptos), unidos entre sí por diferentes clases de enlaces asociativos (relaciones). A su vez, las relaciones entre un concepto y su clase, denominadas *relaciones de subtipo* (ej.instancia_de, es_un), a veces se representan en una red separada.

La principal ventaja de las redes semánticas es que toda la información relativa a un objeto concreto se obtiene fácilmente a partir del mismo, siguiendo los arcos que parten de él.

Para aplicar la minería de datos con redes semánticas, se representa cada ejemplo como una red semántica, y las operaciones que se realizan consisten en manipular los grafos, para encontrar los patrones (subgrafos) que cumplen todos los ejemplos de la misma clase.

Es práctica común el denominar *red semántica* a todo formalismo con forma de red usado para representar conocimiento. Sin embargo, es más preciso considerar como tales sólo las que se dedican a la representación del lenguaje natural, y denominar, en general, *redes asociativas* a todas ellas.

Una red asociativa es una red (conjunto de nodos unidos entre sí por enlaces) en la que los nodos representan conceptos y los enlaces representan relaciones (de pertenencia, inclusión, causalidad, etc.) entre los conceptos. Dentro de las redes asociativas se incluyen: las *redes semánticas* (destinadas a representar o comprender el lenguaje natural), las *redes de clasificación* (representan conceptos mediante una jerarquía de clases y propiedades) y las *redes causales* (representan relaciones de influencia, generalmente de causalidad, entre variables).

Las redes de clasificación pueden considerarse redes semánticas sólo cuando representan conocimiento taxonómico de conceptos, pero no cuando se utilizan dentro de un sistema experto basado en marcos para definir clases e instancias.

Las redes causales representan un modelo en el que los nodos corresponden a variables y los enlaces a relaciones de influencia, generalmente de causalidad. Los modelos causales se orientan, sobre todo, a problemas de diagnóstico. Las redes bayesianas pueden considerarse como redes causales a las que se ha añadido una distribución de probabilidad sobre sus variables.

2.7.1.3.2 Marcos

El concepto de marco fue introducido por Minsky (1975) como método de representación del conocimiento y de razonamiento, intentando superar las limitaciones de la lógica a la hora de abordar problemas como la visión artificial, la comprensión del lenguaje o el razonamiento de sentido común.

Un marco es una estructura de datos, formada por un nombre y un conjunto de campos (o ranuras, del inglés *slot*s), que se rellenan con valores para cada ejemplo concreto. Las ranuras pueden llenarse con valores de atributos o con referencias a otros marcos para expresar las *relaciones de subtip*o, como la relación *es_u*n, en cuyo caso se heredan los atributos del marco de nivel superior.

Basándose en el concepto de marco, se desarrolló el lenguaje de programación KRL *(Knowledge Representation Languag*e), para representar el conocimiento de forma estructurada. La ingeniería del software también heredó de la inteligencia artificial el concepto de marco para construir la orientación a objetos (puede observarse el gran parecido entre los objetos de la programación orientada a objetos y los marcos).

Entre los sistemas de aprendizaje que utilizan marcos para representar el conocimiento adquirido, podemos mencionar el sistema EURISKO.

El conocimiento expresado mediante marcos puede traducirse fácilmente a lógica de predicados de primer orden, aunque perdiendo la ventaja de ser estructurado.

2.7.1.4. Representaciones basadas en ejemplos

Los sistemas de aprendizaje basado en ejemplos *(Instance-Based Learning algorithm*s) representan el conocimiento mediante ejemplos representativos, basándose en "similitudes" entre los datos. El aprendizaje consiste en la selección de los ejemplos que mejor representan a los conceptos existentes en la base de datos (se trata de aprendizaje supervisado); estos ejemplos representativos serán los únicos que se almacenen, reduciendo así considerablemente el espacio necesario. El principal problema de estos sistemas es que se necesita una función de "similitud", a veces difícil de definir, para clasificar los nuevos ejemplos según sea su parecido con los ejemplos prototipo.

Los algoritmos de aprendizaje basado en ejemplos surgieron a partir de los clasificadores por vecindad *(nearest-neighbor classifie*r), y han adquirido importancia más recientemente con los sistemas de razonamiento basado en casos *(case-based reasonin*g), para diagnóstico y resolución de problemas. Además, pueden utilizarse como paso previo a otros sistemas de aprendizaje a partir de ejemplos, para entrenarlos con conjuntos de ejemplos más pequeños y representativos.

2.7.1.5. Redes neuronales

Las redes neuronales, incluidas dentro de los modelos *conexionista*s, son sistemas formados por un conjunto de sencillos elementos de computación llamados *neuronas artificiale*s. Estas neuronas están interconectadas a través de unas conexiones con unos pesos asociados, que representan el conocimiento en la red.

Cada neurona calcula la suma de sus entradas, ponderadas por los pesos de las conexiones, le resta un valor umbral y le aplica una función no lineal (por ej. una sigmoide); el resultado sirve de entrada a las neuronas de la capa siguiente (en redes como el *perceptrón multicapa*).

Uno de los algoritmos más usado para entrenar redes neuronales es el *back-propagatio*n, que utiliza un método iterativo para propagar los términos de error (diferencia entre valores obtenidos y valores deseados), necesarios para modificar los pesos de las conexiones interneuronales. El *back-propagation* puede considerarse como un método de regresión no lineal, en el que aplica un descenso de gradiente en el espacio de parámetros (pesos), para encontrar mínimos locales en la función de error.

Las redes neuronales han sido utilizadas con éxito en diferentes tipos de problemas:

- Auto-asociación: la red genera una representación interna de los ejemplos aportados, y responde con el más aproximado a su "memoria". Ejemplo: máquina de Boltzman.
- Clasificación de patrones: la red es capaz de clasificar cada entrada en un conjunto predefinido de clases. Ej.: back-propagation.
- Detección de regularidades: la red se adapta a los ejemplos de entrada, tomando de ellos varias características para clasificarlos; en este caso, el conjunto de clases no está definido de antemano, por lo que el aprendizaje es no supervisado. Ej.: red MAXNET, ART1, mapas de Kohonen, red de Oja, etc.

Las tasas de error de las redes neuronales son equivalentes a las de las reglas generadas por los métodos de aprendizaje simbólicos, aunque son algo más robustas cuando los datos son ruidosos.

Las principales desventajas para usar redes neuronales en la minería de datos son:

- el aprendizaje es bastante más lento que en un sistema de aprendizaje simbólico;
- el conocimiento obtenido por las mismas no es representable en forma de reglas inteligibles, sino que lo forma el conjunto de pesos de las conexiones interneuronales;
- además, es difícil incorporar conocimiento de base o interacción del usuario en el proceso de aprendizaje de una red neuronal.

2.7.2. APRENDIZAJE

2.7.2.1. Enfoques del aprendizaje: conductista y cognoscitivo

Entre la gran variedad de sistemas de aprendizaje desarrollados en ingeniería del conocimiento, se pueden distinguir dos claras tendencias desde un punto de vista psicológico: el enfoque *conductista* y el enfoque *cognoscitiv*o. Esto marca diferentes actitudes de los sistemas ante el proceso de aprendizaje, así como su empleo en diferentes aplicaciones y el uso de diferentes lenguajes para representar el conocimiento.

2.7.2.1.1 Sistemas conductistas

Según la Psicología conductista, el *aprendizaje* es la capacidad de experimentar cambios adaptativos para mejorar el rendimiento. Siguiendo este enfoque, un sistema de aprendizaje será como una caja negra (de la que no interesa su estructura interna) capaz de adecuar su *comportamiento* para que el rendimiento de sus respuestas ante los datos de entrada aumente durante el proceso de aprendizaje.

Los sistemas de aprendizaje conductistas hacen mayor énfasis en modelos de comportamiento que en la representación interna del conocimiento, que muchas veces es opaca e ininteligible. Los lenguajes de descripción suelen ser diferentes para los objetos y para el conocimiento: los objetos se describen por vectores de características, mientras que para el conocimiento se emplean parámetros o tablas. En cualquier caso, esta representación del conocimiento hace difícil su traducción a reglas que expliquen de forma racional el comportamiento del sistema.

Las aplicaciones de los sistemas de aprendizaje conductista se extienden por varios campos relacionados con la IA: autómatas de aprendizaje, control adaptativo, reconocimiento de formas y clasificación. En general, presentan buena inmunidad al ruido.

Entre los sistemas conductistas de aprendizaje, hay que destacar los sistemas *conexionistas* y los sistemas *evolucionista*s, que realizan inducción de conocimiento.

2.7.2.1.2 Sistemas cognoscitivos

Según el enfoque cognoscitivo de la Psicología, el *aprendizaje* consiste en la construcción y modificación de la *representación del conocimient*o. Durante el proceso de aprendizaje se produce un incremento del conocimiento, que no supone un simple cambio cuantitativo, sino también cualitativo; es decir, no hay un mero aumento del volumen de conocimiento almacenado, sino, sobre todo, una *reorganización* del mismo.

La "calidad" del aprendizaje vendrá dada no sólo por el aumento de precisión del conocimiento almacenado (en una base de conocimiento), sino también por la utilidad del mismo para los objetivos del usuario y por el nivel de abstracción empleado. Por tanto, la representación del conocimiento jugará un papel principal en los sistemas que sigan este enfoque.

Los lenguajes de descripción usados por estos sistemas suelen coincidir para representar a los objetos y al conocimiento. Están basados, normalmente, en la lógica (de proposiciones o de predicados) o en representaciones estructuradas (como los marcos).

Las aplicaciones de los sistemas de aprendizaje cognoscitivo dependen del tipo de aprendizaje que realicen, siendo los más importantes los que utilizan deducción (sistemas EBL, basados en explicaciones), analogía (sistemas expertos basados en casos) o inducción (adquisición y formación de conceptos). Los sistemas inductivos en los que el conocimiento se expresa mediante reglas sencillas se suelen denominar *simbólic*o, y tienen gran importancia para construir bases de conocimiento en sistemas expertos y para extraer conocimiento de grandes bases de datos.

2.7.2.2. Tipos de aprendizaje

En cualquier proceso de aprendizaje, el aprendiz aplica el conocimiento poseído a la información que le llega, procedente de un maestro o del entorno, para obtener nuevo conocimiento, que es almacenado para poder ser usado posteriormente.

Dependiendo del esfuerzo requerido por el aprendiz (o número de inferencias que necesita sobre la información que tiene disponible) han sido identificadas varias estrategias, aunque, en la práctica,

muchos procesos de aprendizaje aplican de forma simultánea varias de ellas. Una clasificación ya "clásica" de los diferentes tipos de aprendizaje, en orden creciente de esfuerzo inferencial por parte del aprendiz, es:

2.7.2.2.1 Aprendizaje por implantación directa 1 *(rote learnin*g)

Es un caso extremo, en el que el aprendiz no ha de realizar ningún tipo de inferencia sobre la información suministrada, sino que la acepta directamente. Esta estrategia incluye aprendizaje por programación y por memorización.

2.7.2.2.2 Aprendizaje por instrucción

El sistema de aprendizaje adquiere el nuevo conocimiento a través de la información proporcionada por un maestro, pero no la copia directamente en memoria, sino que selecciona los datos más relevantes y/o los transforma a una forma de representación más apropiada.

2.7.2.2.3 Aprendizaje por deducción

Partiendo del conocimiento suministrado y/o poseído, se deduce el nuevo conocimiento, es decir, se transforma el conocimiento existente mediante una función preservadora de la verdad.

2.7.2.2.4 Aprendizaje por analogía

Se adquiere un nuevo concepto mediante la modificación de la definición ya conocida de un concepto similar. El aprendizaje por analogía puede ser entendido como una combinación de la inducción y la deducción, ya que mediante la inferencia inductiva se determinan características comunes a los dos conceptos comparados, unificando la misma definición para ambos; entonces se aplica la deducción para obtener las características esperadas para el nuevo concepto. Este tipo de aprendizaje es especialmente importante en la resolución de problemas.

2.7.2.2.5 Aprendizaje por inducción

El sistema de aprendizaje aplica la inducción a los hechos u observaciones suministrados, para obtener nuevo conocimiento. La inferencia inductiva no preserva la verdad del conocimiento, sólo su falsedad; es decir, si partimos de hechos falsos, el conocimiento adquirido por inducción será falso, pero si los hechos son verdaderos, el conocimiento inducido será válido con cierta probabilidad (y no con certeza absoluta, como ocurre con la deducción). Hay dos tipos de aprendizaje inductivo:

- *Aprendizaje con ejemplos*: el nuevo conocimiento es inducido mediante la generalización a partir de una serie de ejemplos y contraejemplos. Este método también se conoce como *adquisición de concepto*s.
- *Aprendizaje por observación* y descubrimiento: el sistema de aprendizaje analiza una serie de entidades y determina que algunas tienen características comunes, por lo que pueden ser agrupadas formando un concepto. Puesto que los conceptos no son conocidos de antemano, este método se llama también aprendizaje *no supervisado* o *formación de concepto*s.

BIBLIOGRAFIA

CRISPDM. “CRoss Industry Standard Process for Data Mining”. http://www.crisp-dm.org. 2003.

Colín-Flores. "Reingeniería de procesos de negocios”. Management Today 1995

Departamento de Ciencias Matemáticas e Informática. “http://dmi.uib.es/~bbuades/riesgos/”. Universidad de les Illes Balears. 1999.

E. Hall. “Managing Risk : Methods for Software Systems Development2. Adison Wesley. 1998.

Fernando Carpani. Tésis de Maestría "CMDM: Un Modelo Conceptual para la Especificación de Bases Multidimensionales". Agosto 2000.

Fernando Martín Sánchez; Nieves Ibarrola De Andrés; Guillermo López Campos. “Minería, Visualización y Descubrimiento de Conocimiento en Bases de Datos”. Unidad de Bioinformática – BIOTIC. 1997.

Froilan Solis Almonacid. “¿Qué es Data warehousing?”. www.monografias.com

H. Gill, P. Rao. “Data warehousing la integración de información para la mejor toma de decisiones”, Prentice-Hall, 1996.

James A. Senn. “Análisis y Diseño de Sistemas de Información”. Segunda Edición. McGraw-Hill, 1998.

José Hernández Orallo. “Análisis y Extracción de Conocimiento en Sistemas de Información: Datawarehouse y Datamining”. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Julio 2003.

Jean Michel Franco. “El Data Warehouse y el Data Mining”. Ed. Gestión 2000–1997.

J. Han, Y. Cai, y N. Cercone. “Knowledge Discovery in Databases: An Attribute-Oriented Approach”, VLDB Conference, Vancouver, Canada, 1992.

J. Han, Y. Cai y N.Cercone. "Data-Driven Discovery of Quantitative Rules in Relational Databases". En IEEE Transactions on knowledge and data engineering, vol. 5(1), Febrero 1993.

Matilde Celma Giménez. “Bases de Datos. http://www.dsic.upv.es/~mcelma”. Dpto. Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. 2003

Olivia R. y Liu Sheng. "Data Warehouse Design: Auxiliary Examples and Concepts". MIS Department. University of Arizona. 2002.

Ralph Kimball. “The Data Warehouse Toolkit”. Wiley Computer Publishing, John Wiley & Sons Inc., New York, Chichester, Brisbane, Toronto, Singapore, 1997.

Ralph Kimball, Laura Reeves, Margy Ross, Warren Thornthwaite: “The Data Warehouse Lifecycle Toolkit”. Wiley Computer Publishing, John Wiley & Sons Inc., New York, Chichester, Brisbane, Toronto, Singapore, Weinheim, 1998

U. M. Fayyad, G. Piatetsky-Shapiro y P. Smyth. “From Data Mining to Knowledge Discovery”. Eds., AAAI Press, Menlo Park, California, 1996.

U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth y R. Uthurusamy. “Advances in Knowledge Discovery and Data Mining”. The AAAI Press, Menlo Park, California, 1996.

U. M. Fayyad. “Data Mining and Knowledge Discovery: Making Sense Out of Data”. En IEEE Expert, vol. 11(5), Oct. 1996.

William H. Inmon. “Building the Data Warehouse”. 2000.

W. J. Frawley, G. Piatetski-Shapiro y C. J. Matheus. "Knowledge Discovery in Databases: An Overview". Knowledge Discovery in Databases,G. Piatetsky-hapiro y W. Frawley, AAAI-MIT Press, Menlo Park, California, 1991.

William H. Inmon. “Creating the Data Warehouse Data Model from the Corporate Data Model”. 2000.

William H. Inmon. “Using the Generic Data Model”. 2000.