

13.1 | Introducción a BB DD Orientadas a Objetos

En la actualidad el mundo de las bases de datos se ha diversificado notoriamente, ya no gobiernan a sus anchas los sistemas de bases de datos relacionales, el enfoque orientado a objetos para las bases de datos (o BDOO) si bien tiene ya varias décadas de haberse promulgado y colocado en el mercado como una serie de productos mezcla de híbridos objeto-relacional, no ha sido por mucho de las variantes que más interés han despertado. Existe toda una gama de productos con enfoques puros no relacionales (noSQL), que han ganado terreno en el ciberespacio, aplicaciones de IoT entre otras áreas en donde se busca simpleza y versiones peso pluma (lightweight) para entregar un desempeño con volúmenes.

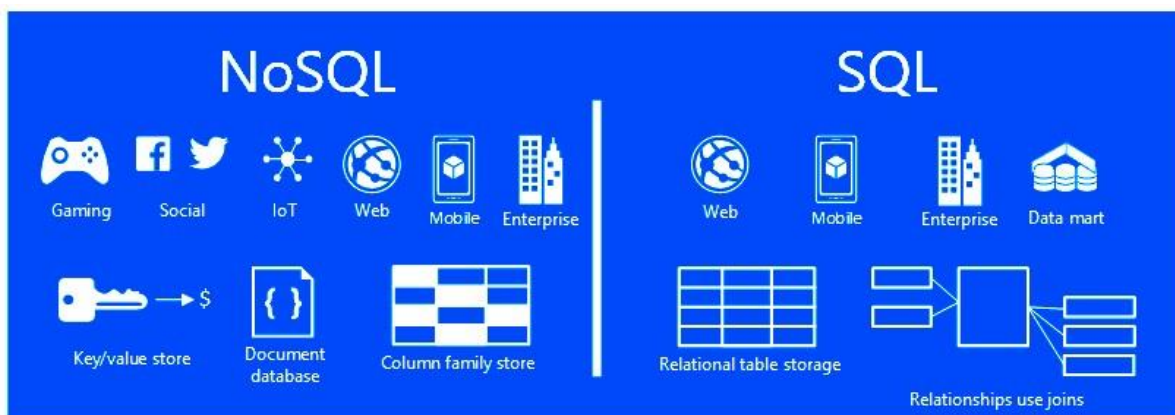
13.2 | Bases de datos NoSQL

El enfoque conocido como sistemas de bases de datos NoSQL deja bien clara su premisa: "No SQL gracias", y entonces ¿Qué son estos sistemas? algunos son una variante simplificada de OO que utilizan una notación de objetos estándar (JSON/BSON) otros son simples estructuras de asociación, empero más o menos todos guardan cierta similitud: la forma de estructurar los datos de manera mas simple y la forma de acceder a estos mediante llamadas a funciones y operaciones consistentes con el CRUD necesario en toda base de datos. Existe toda una gama de productos NoSQL en el mercado, algunos mas otros menos conocidos:



13.3 | Tabla comparativa NoSQL versus SQL

A continuación damos evidencia comparativa de los productos NoSQL versus las bases de datos relacionales tradicionales, desde el punto de vista de las aplicaciones que aprovechan ambos productos y la forma estructural de cada uno:









Diseño de Base de Datos En


la década pasada se generó mucho interés sobre los SBDOO.

- Estos sistemas tienen su origen en los lenguajes de POO.
- Los Sistemas de BD (SBD) han mejorado los procesos de desarrollo de aplicación en grandes ambientes de datos intensivos, proporcionando una sencilla y uniforme vista de datos expresada en términos de estructuras independientes.
- Su alto nivel de características lingüísticas y sus facilidades para compartir información de una manera controlada, hace posible que se puedan crear aplicaciones integradas más fácilmente.
- La integridad de los datos está controlada por el SBD.
- Sino que en su lugar deben tener la posibilidad de manejar objetos y operaciones sobre esos objetos, que se asemejan mucho más a sus contrapartes en la realidad.
- En otras palabras, la idea fundamental es elevar el nivel de abstracción.
- Creen que las técnicas de objetos son el enfoque a escoger para las áreas de aplicaciones “complejas”.

Actualmente, los sistemas de software han puesto un ambiente típico, incluyen herramientas, editores de esquema, verificadores de diseño y programas de circuito disponibles, y todo integrado.

- La disponibilidad de alta ejecución en las estaciones de trabajo gráfica, se han incrementado tanto en la expansión como en la complejidad de las aplicaciones de los datos intensivos.
- Algunos ejemplos son:
 -  Diseño y Manufactura Asistido por Computadora (CAD/CAM).
 -  Ingeniería de Software Asistida por Computadora (CASE).
 -  Sistemas de Información de Oficinas (OIS).
 -  Manufactura Interada por Computadora (CIM).
 -  Sistema de Información Geográfica (GIS).
 -  Ciencia y Medicina.
- Como cualquier BD programable, una BDOO da un ambiente para el desarrollo de aplicaciones con un depósito persistente listo para su explotación.

PRINCIPIOS DE ORIENTACIÓN A OBJETOS

- En una BDOO, cualquier cosa es un objeto y se manipula como tal.
- Un Objeto es una instancia que responde a mensajes activando un método.
 -  La estructura interna no es visible para los usuarios de ese objeto.

- Los sistemas OO proporcionan el concepto de Identificador del Objeto (OID) para identificar a los objetos.
- Los identificadores de objetos son únicos.
- Cada objeto tiene un solo identificador y no hay dos objetos que tengan el mismo identificador.
- Generalmente el identificador lo genera el sistema de manera automática.
- A diferencia de los BD relacionales donde un valor de dato único, se genera de manera externa al sistema.
- Si se desplazan los datos a un sistema de BD diferente, hay que traducirlos.
- Además, los identificadores generados por el sistema pueden resultar redundantes, si las entidades que se modelan ya disponen de identificadores unívocos externos al sistema.
- Por lo general, los SMBDR se apoyan en claves definidas y controladas por el usuario (claves de usuario), para efectos.

PRINCIPIOS DE ORIENTACIÓN A OBJETOS

Los objetos soportan una serie de características:






- Se agrupan en tipos denominados clases.
- Contienen datos internos que definen su estado actual.
- Soportan ocultación de datos.
- Pueden heredar propiedades de otros objetos.
- Pueden comunicarse con otros objetos enviando o pasando mensajes.
- Tienen métodos que definen su comportamiento. Las clases son una colección de objetos con propiedades similares, compartimiento común, relaciones comunes a otras clases.
- **La instancia** es un objeto con propiedades definidas en su descripción de la clase.
- El mensaje es una clase que debe tener un método correspondiente. Un mensaje puede ser enviado a un objeto a ejecutar una acción.
- El **método** es una lista de instrucciones detalladas que definen cómo responde un objeto a un mensaje en particular.
- La **superclase** es la clase que deriva a otra clase.
- La **subclase** es la clase derivada de una superclase.

La liga expresa compatibilidad de relaciones entre las clases.

- Los objetos heredan las características de su clase y de todas las clases de nivel superior a la que pertenecen.
- Estos principios y técnicas hacen que las BDOO estén adecuadas a aplicaciones que implican tipos de datos complejos.
- Tipos de datos complejos, tales como documentos compuestos o de diseño asistidos por computadora que combinan texto, gráficos y hojas de cálculo.
- La BD proporciona un modo natural de representar las jerarquías que aparecen en los datos complejos.
- La jerarquía de clases permite al BD seguir la pista del tipo de cada objeto en el documento.
- Finalmente, el mecanismo de mensajes ofrece soporte natural para una interfaz de usuario gráfica.

BASES DE DATOS ORIENTADAS A OBJETOS (BDOO)

- El campo de las BDOO se ha introducido como una nueva área de investigación.
- Los campos de Lenguajes de Programación, Inteligencia Artificial e Ingeniería de Software han contribuido con el uso de la tecnología OO en el área de las BD.
- El desafío del área de BD es integrarlos en un diseño de sistema simple que mantenga el equipo deseado para cada campo.
- El resultado es conservar las características centrales de las BD modernas, incluyendo:

-  Persistencia
-  Control de Concurrencia
-  Recuperación
-  Consistencia
-  Lenguaje de Consulta

- Desde el punto de vista del desarrollador las ventajas están dadas en ganancias de productividad, dado su modelado más simple que facilita el desarrollo de aplicaciones.
- El modelado de aplicaciones es mucho más sencillo gracias al concepto de objetos complejos y el modelo obtenido es fácilmente comprensible para el usuario.
- Este modelo puede ser validado directamente por el cliente de la aplicación.
- El enfoque OO favorece la reutilización.
- Porque gracias al encapsulamiento y a la herencia, es fácil especializar un componente existente para responder a las necesidades particulares de la aplicación.

- Desde el punto de vista del usuario final, los SMBDOO, proporcionan mayor aporte en la calidad en términos de ergonomía, fiabilidad, evolución y desempeño.
- El Modelo Orientado a Objetos (MOO) se basa en encapsular código y datos en una única unidad, llamada objeto.
- La interfaz entre un objeto y el resto del sistema se define mediante un conjunto de mensajes.
- El término mensaje en un contexto orientado a objetos, no implica el uso de un mensaje físico en una red de computadoras.
- Si no que se refiere al paso de solicitudes entre objetos sin tener en cuenta detalles específicos de implementación.
- La capacidad de modificar la definición de un objeto sin afectar al resto del sistema, está considerada como una de las mayores ventajas del modelo de POO.

JERARQUÍA DE CLASES

En una BD existen objetos que responden a los mismos mensajes, utilizan los mismos métodos y tienen variables del mismo nombre y tipo.

- Sería inútil definir cada uno de estos objetos por separado por lo tanto se agrupan los objetos similares para que formen una clase, a cada uno de estos objetos se le llama instancia de su clase.
 - Todos los objetos de su clase comparten una definición común, aunque difieran en los valores asignados a las variables.
 - Así que básicamente las BDOO tienen la finalidad de agrupar aquellos elementos que sean semejantes en las entidades para formar una clase, dejando por separado aquellas que no lo son.
- Por ejemplo: Tomemos como referencia la entidad **Alumno** y la entidad **Maestro**.

Donde los atributos considerados para cada uno, son en Alumno:



Nombre

Dirección Teléfono

Especialidad



Semestre



Sección



Maestro:



Nombre



Dirección



Teléfono



Número catedrático



Plaza

- Los atributos de Nombre, Dirección y Teléfono se repiten en la entidad Alumno y Maestro, así que podemos agrupar estos elementos para formar la clase Persona, con dichos campos.
- Quedando por separado en Alumno:
 - Especialidad
 - Semestre
 - Sección
- Y en Maestro:
 - Número Catedrático
 - Plaza



Una BDOO debe tener por lo mismo los siguientes requerimientos:

- Debe proporcionar funcionalidad en la BD.
- Incluir todas las características esenciales.
- Debe soportar la Identificación de Objetos (OID).
- Debe proporcionar Encapsulamiento. Esta encapsulación puede ser la base por la cual todos los objetos abstractos son definidos.
- Debe soportar objetos con estado complejo. El estado de un objeto puede referirse a otros objetos.

HERENCIA

Las clases en un Sistema OO se representan en forma jerárquica.

- Así que las propiedades o características de un elemento, las contendrán (o heredarán) los elementos que de ésta se deriven.
- Decimos por lo tanto que las entidades derivadas son subclases de la clase padre o Superclase.
- Se pueden crear muchas agrupaciones (clases) para simplificar un modelo.

CONSULTAS ORIENTADAS A OBJETOS

Los lenguajes de POO, requieren que toda la interacción con objetos se realice mediante el envío de mensajes.

- El término mensaje en un entorno OO no implica el uso de mensajes físicos en la red.
- Por el contrario hace referencia al intercambio de solicitudes entre los objetos.
- Se utiliza a veces la expresión invocar a un método para denotar el hecho de enviar un mensaje a un objeto y la ejecución del método correspondiente.
 - Dado que la única interfaz externa presentada por cada objeto es el conjunto de mensajes a los que responde, resulta posible modificar las definiciones de los métodos y de las variables sin afectar al resto del sistema.
- La posibilidad de modificar la definición de un objeto sin afectar al resto del sistema se considera una de las mayores ventajas del paradigma de la POO.
- Consideremos un ejemplo con una entidad Alumno, y deseamos que la consulta realice la búsqueda de todos los alumnos que cursan la materia de Base de Datos II.
- Para realizar esta consulta, se tendría que enviar un mensaje a cada instancia Alumno dentro del sistema.
 - Generalmente en una BD hay muchos objetos similares.
- Por similar se entiende que responden a los mismos mensajes, utilizan los mismos métodos y tienen variables del mismo nombre y del mismo tipo.
- Así un lenguaje de consultas para un sistema de BDOO, debe incluir tanto el modelo de pasar el mensaje de objeto a objeto, como el modelo de pasar el mensaje de conjunto en conjunto.

Ejemplo del lenguaje para Manipulación de objetos C++ de ODMG


```

int crear_titular_cuenta(String nombre, String direccion{
    d_Database bd_banco_obj;
    d_Database * bd_banco = &bd_banco_obj;
    bd_banco->open("BD-Banco");
    d_Transacction Trans;
    Trans.begin();

    d_Ref<Cuenta> cuenta = new(bd_banco, "Cuenta")Cuenta;
    d_Ref<Cliente> clien = new(bd_banco, "Cliente")Cliente;
    clien->nombre = nombre;
    clien->dirección = dirección;
    clien->cuentas.insert_element(cuenta);
    cuenta->titulares.insert_element(clien);

    .....
    Trans.commit();
    Bd_banco->close();
}

```

COMPLEJIDAD DE MODIFICACIÓN

En las BDOO pueden existir los siguientes cambios:

- Adición de una nueva clase: Para realizar este proceso, la nueva clase debe colocarse en la jerarquía de clase o subclase, cuidando las variables o métodos de herencia correspondientes.
- Eliminación de una clase: Se requiere la realización de varias operaciones, se debe de cuidar los elementos que se han heredado de esa clase a otras y reestructurar la jerarquía.
- En sí la estructuración de modelos OO simplifica la estructura, evitando elementos o variables repetidas en diversas entidades.

COMPLEJIDAD DE MODIFICACIÓN

- Sin embargo el precio de esto es dedicarle un minucioso cuidado a las relaciones entre las clases cuando el modelo es complejo.

- La dificultad del manejo de objetos radica en la complejidad de las modificaciones y eliminaciones de clases.
- Ya que de tener variables que heredan de otros objetos, se tiene que realizar una reestructuración que involucra una serie de pasos complejos.

ARQUITECTURA

Una manera de desarrollar un sistema distribuido es la de Cliente/Servidor.

- La mayoría de los actuales SMBDOO son sistemas cliente/servidor.
- Donde el cliente solicita “objetos” del servidor.
- Y el servidor los obtiene de la B | D y los regresa al cliente solicitante.
- Estos sistemas son llamados Servidor de Objetos.

ARQUITECTURA DE UNA BDOO

En la siguiente se muestran algunos de los principales productos de BDOO y sus vendedores.

<i>PRODUCTO</i>	<i>PROVEEDOR</i>
<i>Gemstone</i>	<i>Servio Corporation, Alameda, CA</i>
<i>Itasca</i>	<i>Itasca Systems, Inc., Minneapolis, MN</i>
<i>Objectivity</i>	<i>Objectivity, Menlo Park, Ca</i>
<i>Object Store</i>	<i>Object Design, Inc., Burlington, MA</i>
<i>Ontos</i>	<i>Ontos Inc., Bellerica, MA</i>
<i>Versant</i>	<i>Versant Object Technology, Menlo Park, CA</i>

Seis Productos de BDOO y sus Proveedores.

Desarrollo con Bases de Datos OO

- Las BDOO se desarrollan al describir, en primer lugar, los tipos de objetos importantes del dominio.
- Estos tipos de objetos determinan las clases que conformarán la definición de la BDOO.
- Por ejemplo: Una BD diseñada para almacenar la geometría de ciertas partes mecánicas incluiría clases como CILINDRO, ESFERA y CUBO.
- El comportamiento de CILINDRO podría incluir información relativa a sus dimensiones, volumen, área, etc.:

```
• Clase de CILINDRO{  
    ALTURA          FLOTANTE 0;  
    RADIO            FLOTANTE 0;  
    VOLUMEN          FLOTANTE 0;  
    AREA             FLOTANTE 0;  
};
```

- Se puede llegar a definiciones similares para el cubo y la esfera.
- En la definición anterior, ALTURA, RADIO y ÁREA representan los mensajes que se pueden enviar a un objeto CILINDRO.
- La Implantación se lleva a cabo en el mismo lenguaje, escribiendo funciones correspondientes a las solicitudes OO:

CILINDRO::ALTURA () {RETORNA CILINDRO_ALTURA;}

CILINDRO::VOLUMEN () {RETORNA PI * RADIO () * ALTURA () ;}

- En este caso, la Altura se almacena como un elemento de los datos, mientras que Volumen se calcula mediante la fórmula apropiada.
- Observe que la implantación interna de Volumen utiliza solicitudes para obtener Altura y Radio.
- Sin embargo, el aspecto más importante es la sencillez y uniformidad que experimentan los usuarios de CILINDRO.
- Sólo necesitan conocer la forma de enviar una solicitud y las solicitudes disponibles.

TRES ENFOQUES DE CONSTRUCCIÓN DE BDOO

- Las BDOO se pueden construir mediante alguno de los tres enfoques siguientes:

El Primero: Se puede utilizar el código actual altamente complejo de los sistemas de administración de las BD, de modo que una BDOO se implante más rápido sin tener que iniciar de cero.

- Las Técnicas OO se pueden utilizar como medios para el diseño sencillo de sistemas complejos.
- Los sistemas se construyen a partir de componentes ya probados con un formato definido para las solicitudes de las operaciones del componente.

El Segundo: Considera a la BDOO como una extensión de la tecnología del BD por Relación.

- De este modo, las herramientas, técnicas, y vasta experiencia de la tecnología por relación se utilizan para construir un nuevo SMBD.
- Se pueden añadir apuntadores a las tablas de relación para ligarlas con objetos binarios de gran tamaño.
- La BD también debe proporcionar a las aplicaciones clientes, un acceso aleatorio y por partes a grandes objetos, con el fin de que sólo sea necesario recuperar a través de la red la parte solicitada de los datos.



El Tercero: Reflexiona sobre la arquitectura de los SBD y produce una nueva arquitectura optimizada, que cumple las necesidades de la tecnología

OO.

- Las compañías como Versant, Objectivity, Itasca, etc., utilizan este enfoque y afirman que la tecnología de relación es un subconjunto de una capacidad más general.
- Además, mencionan que las BDOO son aproximadamente dos veces más rápidas que las bases de datos por relación, para almacenar y recuperar la información compleja.
- Por lo tanto, son esenciales en aplicaciones como CAD y permitirían que un depósito CASE fuera una facilidad de tiempo real en vez de una facilidad por lotes.
- La Arquitectura de Versant está orientada al soporte Cliente/Servidor con acercamiento a la computación distribuida.
- Cualquier petición del Cliente, el Servidor la procesa. Los servidores pueden cooperar en una BD distribuida de Versant.
- Las BD pueden estar levantadas como un sistema **m-Cliente/n-Servidor**.

IMPACTO DE LA ORIENTACIÓN A OBJETOS EN LA INGENIERÍA DEL SOFTWARE

- En las BDOO, la organización "Grupo de Administración de Datos Objeto (ODMG)" representa el 100% de las BDOO industriales.
- Y ha establecido un estándar de BD equivalente a SQL:

 ODL: Lenguaje de Definición de Datos Objetos, y
 OQL: Lenguaje de Consulta a Objetos.

- Respecto a las relacionales, todas (Oracle, Informix, etc.) están añadiendo en mayor o menor grado algunos aspectos de la OO.
- ANSI (Instituto Nacional Americano de Estándar), por su parte, está definiendo un SQL-3 que incorpora muchos aspectos de la OO.
- El futuro del SQL-3 es sin embargo incierto, ya que ODMG ha ofrecido a ANSI su estándar para que sirva de base para un nuevo SQL.

- Con lo que sólo habría un único estándar de BD.

RENDIMIENTO

Un SMBDOO debe satisfacer dos criterios:



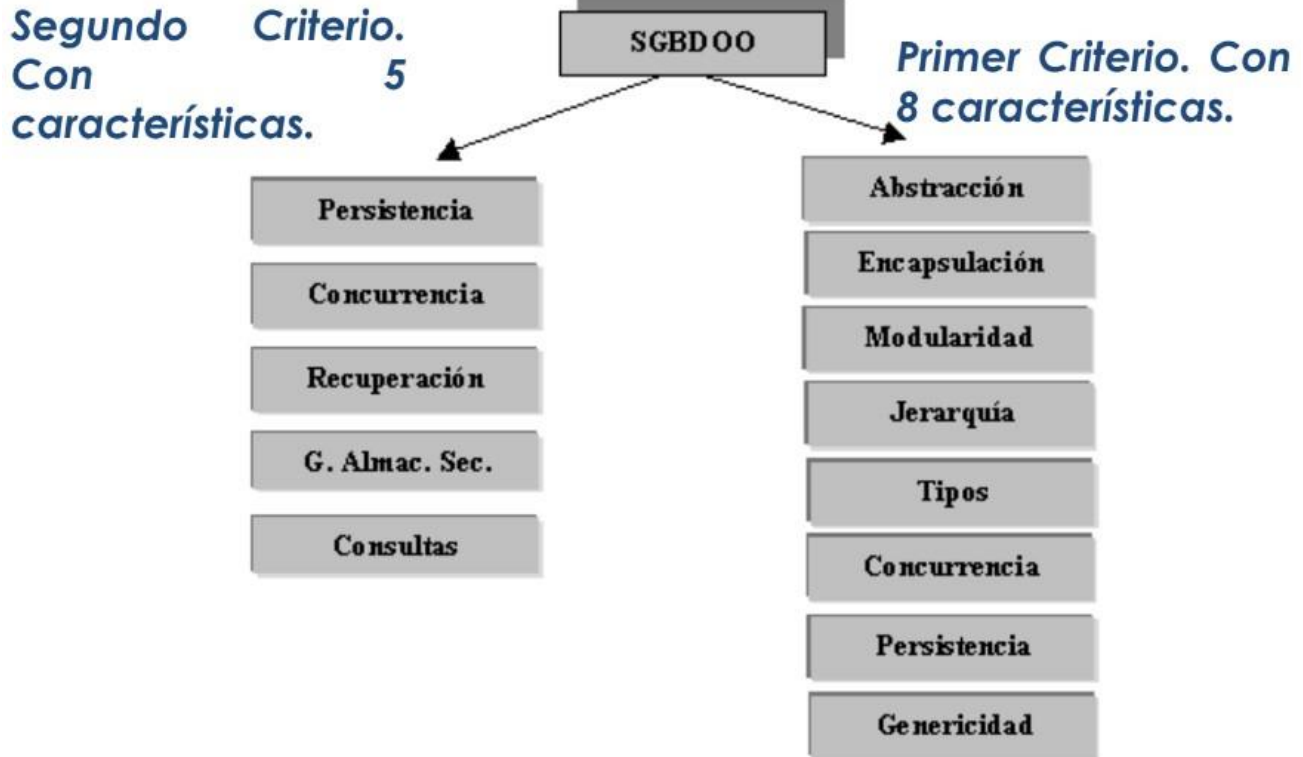
Ser un Sistema OO.



Y ser un Sistema de Administración de BD.

- El primer criterio se traduce en ocho características generales: Abstracción, encapsulación, modularidad, jerarquía, control de tipos, concurrencia, persistencia y genericidad.
- El segundo criterio se traduce en cinco características principales: Persistencia concurrencia, recuperación ante fallos del sistema, gestión del almacenamiento secundario y facilidad de consultas.

Características de SGBDOO



RENDIMIENTO

- Como se puede observar la Persistencia, al igual que la **Concurrencia**, son características del SGBDOO heredadas tanto del SGBD como del Modelo de Objetos.
- La **Persistencia** en el caso del SGBD hace referencia a la conservación de los datos después de la finalización del proceso que los creó.
- En el caso del Modelo de Objetos, se refiere no sólo a la conservación del estado de un objeto, sino también a la conservación de la clase, que debe trascender a cualquier programa individual.
- De forma que todos los programas interpreten de la misma manera el estado almacenado.
- La **Concurrencia** heredada del SGBD se refiere a la capacidad del sistema para gestionar a múltiples usuarios interactuando concurrentemente sobre el sistema.
- Mientras que la **concurrencia** heredada del Modelo de Objetos, hace referencia a la capacidad de distinguir a un objeto activo de otro que no lo está.

- **Persistencia:** Es la capacidad que tiene el programador para que sus datos se conserven al finalizar la ejecución de un proceso, de forma que se puedan reutilizar en otros procesos.
- **Concurrencia:** Se relaciona con la existencia de muchos usuarios interactuando concurrentemente en el sistema.
- Éste debe controlar la interacción entre las transacciones concurrentes, para evitar que se destruya la consistencia de la BD.
- **Recuperación:** Proporcionar como mínimo el mismo nivel de recuperación que los SBD actuales.
- De forma que, tanto en caso de fallo de hardware como de software, el sistema pueda retroceder hasta un estado coherente de los datos.
- **Gestión del almacenamiento secundario:** Es soportada por un conjunto de mecanismos que no son visibles al usuario.
- Tales como gestión de índices, agrupación de datos, selección del camino de acceso, optimización de consultas, etc.

Gestión del almacenamiento secundario

- Estos mecanismos evitan que los programadores tengan que escribir programas para mantener índices, asignar el almacenamiento en disco, o trasladar los datos entre el disco y la memoria principal.
- Creándose de esta forma una independencia entre los Niveles Lógicos y Físicos del sistema.

Facilidad de Consultas

- Permitir al usuario hacer cuestiones sencillas al BD. Este tipo de consultas tienen como misión proporcionar la información solicitada por el usuario de una forma correcta y rápida.

INTEGRIDAD

- Los sistemas de objetos no soportan generalmente restricciones de integridad declarativas.
- En vez de ello, requieren que tales restricciones se hagan cumplir por medio de código procedural (por métodos o programas de aplicación).

Niveles:

- ✚ Ningún soporte del sistema.
- ✚ Validación de referencias.
- ✚ Mantenimiento del sistema
- ✚ “Semántica personalizada”

Comprobación: BDOO

1. ¿Es correcta la siguiente equivalencia para motores de BDNoSQL?

Colección <==> Base de datos

Documento <==> Tabla

Par objeto:valor <==> Registro

✚ Respuesta: **Falso**

2. En general, las aplicaciones para móviles (apps) prefieren utilizar motores de bases de datos tradicionalmente relacionales debido a la fiabilidad, estabilidad y consolidación de sus proveedores.

✚ Respuesta: **Falso**

3. Tecnología orientada a objetos pura.

✚ Respuesta: **Java**

4. ¿Cuál de las siguientes tecnologías son modelos mixtos objeto-relacionales?

✚ Respuesta: **SQL Server**

5. Las tecnologías de bases de datos NoSQL implican que no son relacionales, pero si pueden utilizar lenguaje SQL

🚩 Respuesta: **Falso**

15.1 | ¿Que es la Inteligencia de Negocios?

¿Qué es la inteligencia de negocios?

La inteligencia de negocios o Business Intelligence (BI, por sus siglas en inglés) es una tendencia tecnológica con una clara meta corporativa que es la de transformar la información de valor en conocimientos que ayuden a mejorar significativamente el control, las decisiones y estrategias corporativas.

La inteligencia de negocios incluye metodologías, en conjunción con procesos, herramientas y tecnologías que reúnen, depuran y transforman datos desde los sistemas de información tradicionales en las empresas. La información obtenida puede presentarse como resúmenes, reportes o bien como base para el análisis y constituirse en conocimiento clave para la toma de decisiones.



Figura 1. Beneficios del Business Intelligence

15.2 | Oportunidades con Business Intelligence

"El conocimiento es información interrelacionada y extrapolada; la inteligencia que orienta nuestra conducta y comportamiento."

¿Qué es el BI?

Es una tendencia tecnológica y metodológica que transforma la información de valor en conocimiento que ayuda a mejorar significativamente las decisiones y estrategias corporativas.



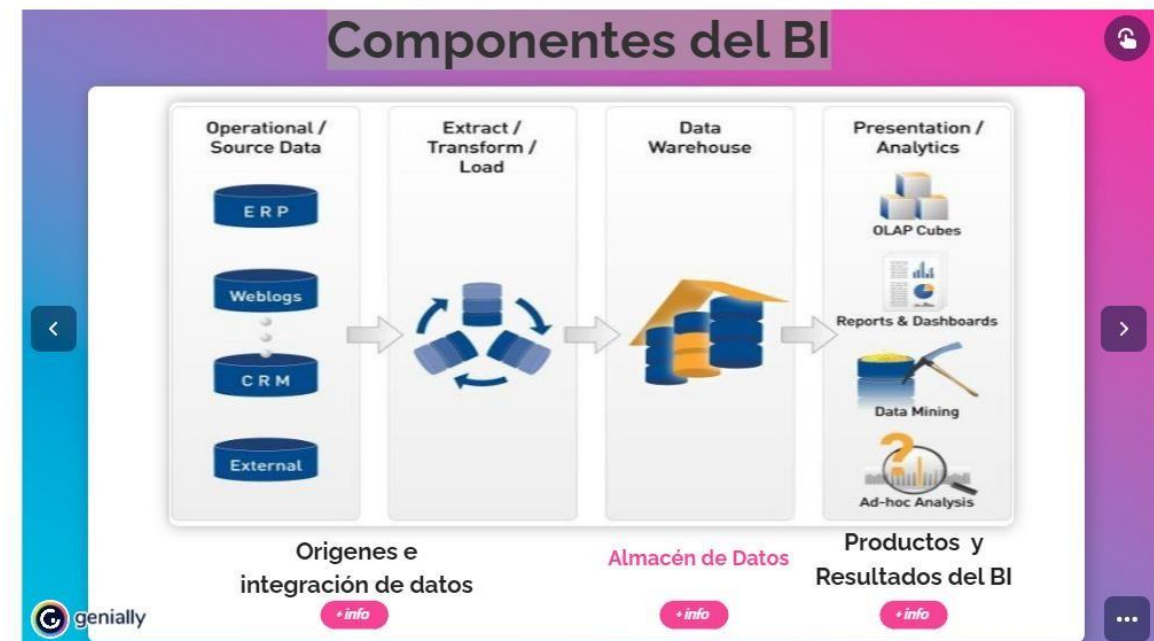
¿Para qué el BI?



Aplicaciones y productos notables del BI



Componentes del BI



Orígenes e integración de datos

Las fuentes u orígenes de datos son externos al sistema, obteniéndose desde sistemas tradicionales OLTP hasta sistemas empresariales ERP, Internet, hojas de cálculo entre otros. La integración de las fuentes externas se consigue a través de un proceso denominado ETL que consolida y facilita la incorporación al sistema.

Almacenes de datos

Los almacenes de datos (data warehouses) conforman el componente medular del sistema, deposito natural del proceso de integración. La inteligencia de negocios no se limita solo a los almacenes de datos podría usar datos directamente de las fuentes externas como ERP entre otros.

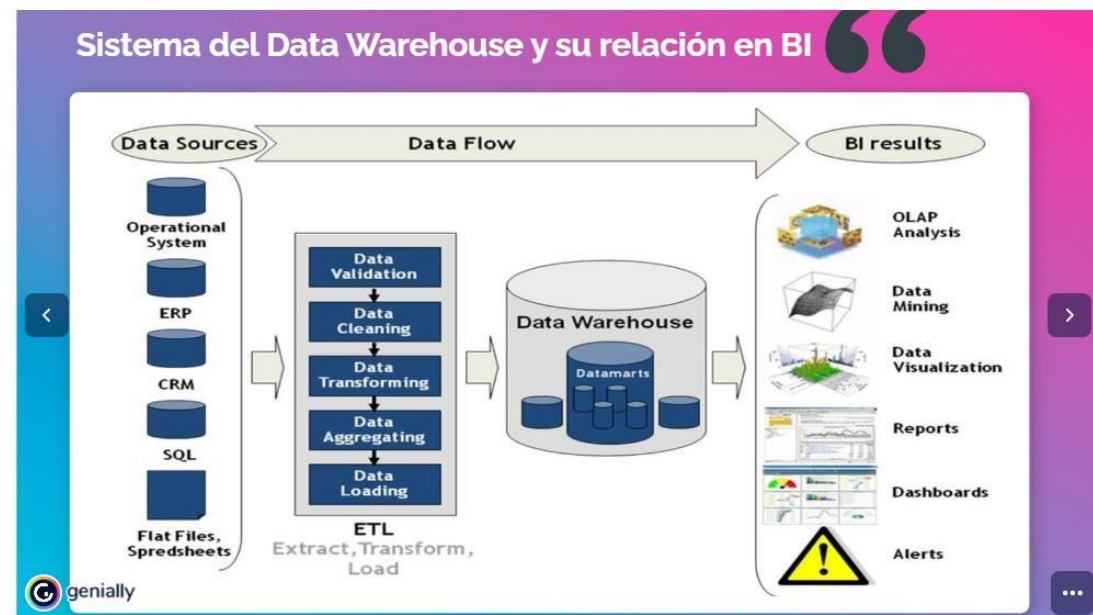
Resultados y productos del BI

Aquí se concentran distintas tecnologías que permiten la plena apreciación y aplicación del BI, desde la presentación a través de reportería, consulta y tableros de información hasta procesamiento analítico complejo como la minería de datos y los servicios OLAP.

Data Warehouse

Sistema que recupera y consolida datos obtenidos desde diversas fuentes externas, caracterizados por:

- Orientar su diseño a temas y aspectos
- Almacenar datos históricos
- Integrar y homogeneizar los datos
- Un ritmo de actualización no periódico
- Información no volátil estable a lo largo del tiempo



DATA WAREHOUSE

Mercado

Existen desde hace mucho antes del BI, típicamente es un almacén de datos dimensional (DDS), para integrar datos desde distintas fuentes.

Competencia

Consultas directas, hojas de cálculo y tablas pivot.

VS

BUSINESS INTELLIGENCE Mercado

Tendencia relativamente reciente, puede o no incluir un DDS, muchas alternativas existentes algunas específicas y otras de índole general.

Competencia

Dashboard BSC, servicios OLAP, data mining, big data.



Oportunidades con Business Intelligence

Paso 1: Seleccionar los sistemas fuentes/externos

Paso 2: Integrar los datos (paquetes ETL)

Paso 3: Normalizar en almacén de datos DDS

Paso 4: Procesar resúmenes, reportes, reportes, tableros o analíticos en línea.

Paso 3: La conformación del almacén de datos dimensional puede requerir de pruebas y ensayos para depurar los datos.

Paso 4: Los procesos en línea OLAP para el BI requieren el tratamiento del almacén de datos como una base de datos multidimensional.

15.3 | Información y Conocimiento







El conocimiento

El conocimiento es una mezcla de experiencia, valores, información y know-how (saber cómo) que sirve como marco para la incorporación de nuevas experiencias e información, y es útil para la acción. Se origina y aplica en la mente de los conocedores.

En las organizaciones con frecuencia no sólo se encuentra dentro de documentos o almacenes de datos, sino que también está en rutinas organizativas, procesos, prácticas, y normas.

El conocimiento se deriva de la información, así como la información se deriva de los datos. Para que la información se convierta en conocimiento es necesario realizar acciones como:






-  Comparación con otros elementos de información.
-  Predicción de consecuencias.
-  Búsqueda de conexiones.
-  Conversación con otros portadores de conocimiento.

El conocimiento es información interrelacionada y extrapolada; la inteligencia que orienta nuestra conducta y comportamiento.

La información

Conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), y que por lo tanto son de utilidad para quién debe tomar decisiones, al disminuir su incertidumbre.

Los datos se pueden transformar en información añadiéndoles valor:

-  **Contextualizándolos:** Se sabe en qué contexto y para qué propósito se generaron.
-  **Categorizándolos:** Se conocen las unidades de medida que ayudan a interpretarlos.
-  **Calculándolos:** Los datos pueden haber sido procesados matemática o estadísticamente.
-  **Corrigiéndolos:** Se han eliminado errores e inconsistencias de los datos.
-  **Condensándolos:** Los datos se han podido resumir de forma más concisa (agregación).

Por tanto, la información es la comunicación de conocimientos o inteligencia, y es capaz de cambiar la forma en que el receptor percibe algo, impactando sobre sus juicios de valor y sus comportamientos.

Información = Datos + Contexto (añadir valor) + Utilidad (disminuir la incertidumbre)

Datos

Los datos son la mínima unidad semántica, y se corresponden con elementos primarios de información que por sí solos son irrelevantes como apoyo a la toma de decisiones: número de teléfono, nombre, dirección, etcétera. Estos se recogen en masa en el nivel operativo del negocio.

15.4 | Productos y resultados del BI

Toda solución de inteligencia de negocios en el mercado se encuentra en uno de dos grandes subsegmentos:

1. **Plataforma de Business Intelligence:** que son herramientas completas que proporcionan mayor flexibilidad en su creación y se utilizan cuando hay necesidad de analizar aplicaciones complejas con muchos cálculos.

2. **Sistemas de Business Intelligence Empresarial:** que son herramientas de funcionalidad específica, permiten mapear los distintos orígenes de datos y proveer los reportes y resúmenes menos analíticos a sus usuarios tal y como sean definidos.

La siguiente es una lista no exhaustiva de aplicaciones y productos notables del BI, ampliamente conocidos:

- En **sistemas de cuadros de mando integral, BSC** (Balance Scorecard, por sus siglas en inglés); más conocido como dashboard o tableros de información que dan un rápido sumario de alto nivel sobre los principales indicadores clave de desempeño del negocio en áreas como clientes, operaciones, finanzas y aprendizaje accediendo a esto a través de interfaces como medidores, cuadros, mapas coloreados, entre otros.
- En **sistemas decisionales, DSS y ESS (Decision Support System y Executive Support System, por sus siglas en inglés)**: Con reportes y resúmenes de desempeño e información estratégica para apoyar la toma de decisiones de la administración y la alta dirección.
- **Herramientas para minería de datos (data mining) y análisis estadísticos**: herramientas usadas para desempeñar modelado predictivo o para descubrir la relación causa efecto entre dos métricas.
- **Servicios de cubos de análisis, OLAP**: son herramientas del Business Intelligence usadas para proveer de capacidades analíticas a los administradores de negocios.



En resumen, una solución de Business Intelligence completa permitirá:

- ✚ Observar ¿qué está ocurriendo?
- ✚ Comprender ¿por qué ocurre?
- ✚ Predecir ¿qué ocurriría?
- ✚ Colaborar ¿qué debería hacer el equipo?
- ✚ Decidir ¿qué camino se debe seguir?

Es decir, con la información plenamente comprendida, la toma de decisiones nos conducirá a acciones que mejoren la calidad de la inteligencia misma, en un círculo de feedback.



15.5 | Componentes del Business Intelligence

Los componentes esenciales del Business Intelligence lo conforman la siguiente clase de componentes de datos y de procesos:

- ✚ **Servicios de análisis y minería de datos:** Tecnologías para procesamiento del análisis de cubos de datos OLAP y descubrimiento de patrones de comportamiento del negocio.
- ✚ **Datawarehousing:** Procesos en los cuales se conforman los almacenes de datos o **datawarehouse** para interrelacionar y manejar los aspectos de negocios para reporteria y procesos analíticos.
- ✚ **Datamarts:** Son subconjuntos de almacenes de datos enfocados en áreas de negocio particulares.

- Integrador de orígenes de datos: Para manejo de fuentes de datos con aplicaciones heterogéneas, éste proceso se maneja a través de los denominados paquetes ETL.

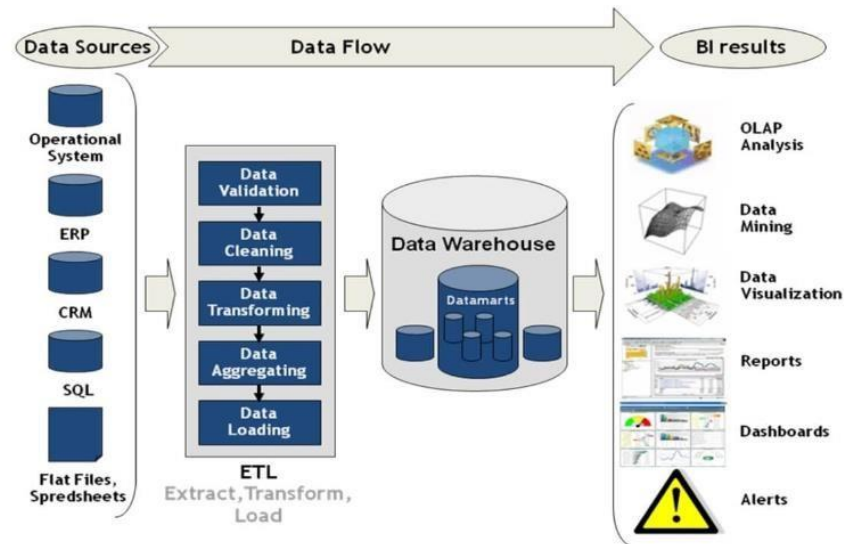


Figura 2. Conformación de un datawarehouse y su aplicación en BI

Como puede apreciarse la conformación del datawarehouse o almacenes de datos constituye un componente clave del Business Intelligence pero como señala Rainardi (2008) la actividad en la inteligencia de negocios no se limita solo a datos en los almacenes de datos, esta puede usar datos directamente desde un ERP o cualquier otro sistema empresarial para su cumplir su propósito.

En la siguiente sección se amplían los fundamentos del datawarehouse. La figura 3 ilustra el cuadro completo de componentes típicos en una solución de Business Intelligence,

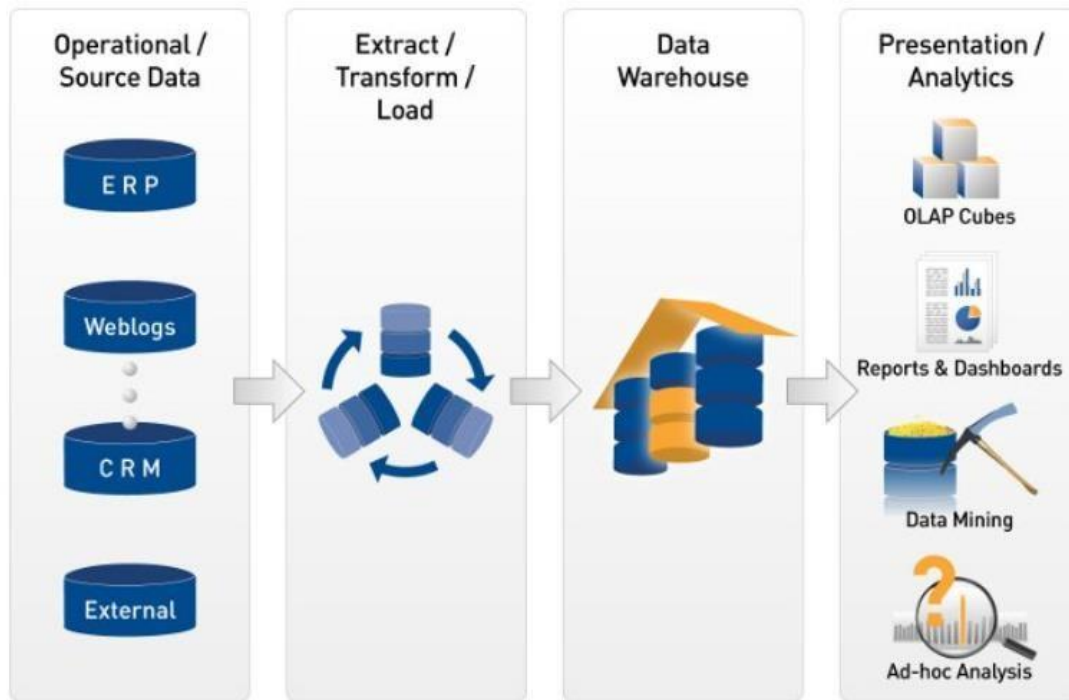


Figura 3. Componentes de una solución de Business Intelligence

16.1 | ¿Qué es un Data Warehouse?

Los almacenes de datos o Data Warehouse constituyen como ya se dijo, el componente medular del Business Intelligence.

En Rainardi (2008) se define Data Warehouse como “un sistema que recupera y consolida datos periódicamente desde otros sistemas fuentes hacia un depósito de datos normalizado o dimensional. Usualmente mantiene años de histórico y es consultado por la inteligencia de negocios y otras actividades analíticas”.

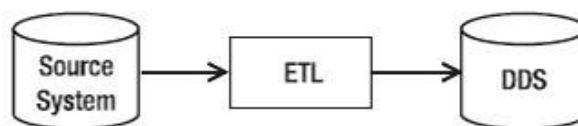


Figura 4. La forma más simple de un sistema de *Data Warehouse*

En la figura 4 se aprecia la forma más simple para entender la constitución de un almacén de datos. La consolidación de los datos desde **sistemas fuentes** (que no forman parte del sistema) sucede a través de un proceso de integración denominado **ETL** (Extraction, Transform and Load, por sus siglas en inglés) que permite la recuperación, transformación y carga de los datos en el **depósito de datos dimensional, DDS (Dimensional Data Store, por sus siglas en inglés)**. Note aquí en esto último, que la definición anterior menciona tanto una estructura dimensional como normalizada, más adelante volveremos a este tema.

El sistema de Data Warehouse anteriormente ilustrado funciona con el mínimo de dos componentes: el ETL y el DDS. Al no advertir ningún componente de interfaz de aplicación en este sistema de Data Warehouse mínimo el usuario podría utilizar por ejemplo SQL directo para realizar consultas al mismo.

Sin embargo, el cuadro completo estructural de un sistema Data Warehouse lo complementan como ya se dijo las aplicaciones o usos del mismo en el marco de la inteligencia de negocios y otras actividades afines: procesamiento analítico OLAP, minería de datos, reportes, dashboard, hojas de cálculo, tablas pivot, entre otros. La figura 5 ilustra un diagrama completo del sistema Data Warehouse que resalta los distintos componentes y flujo del mismo.

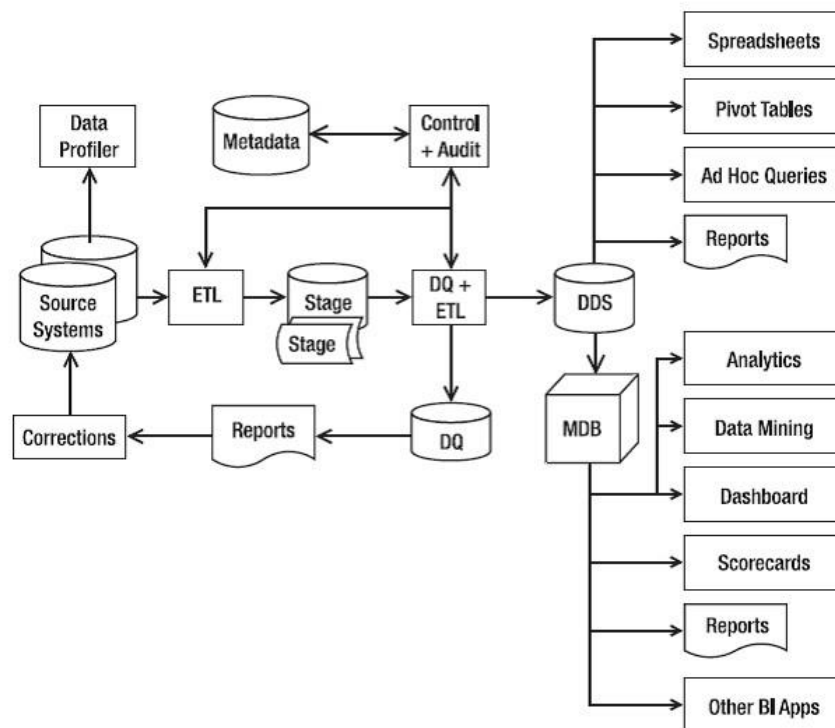


Figura 5. Diagrama del sistema *Data Warehouse*

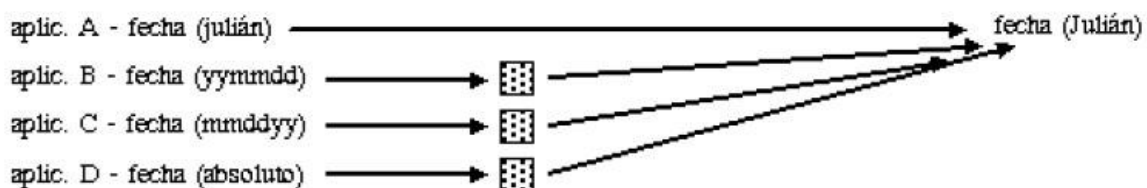
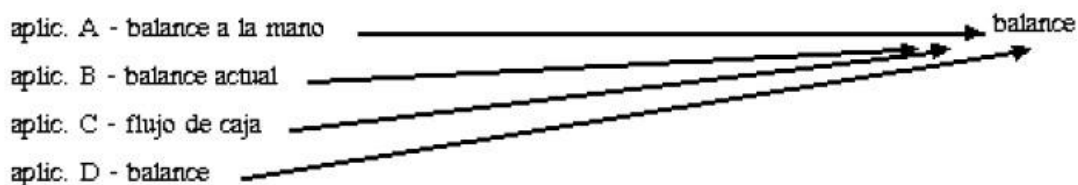
Source Systems (Sistemas fuente)	Son los sistemas OLTP que contienen los datos que desee cargar en los datos almacén.
Data Profiler (perfilador de datos)	Es una herramienta que tiene la capacidad para analizar los datos, tales como encontrar el número de registros en cada tabla, el número de filas que contiene valores NULL, etc.
ETL (Extract – Transform – Load)	Es un sistema que tiene la capacidad de conectar con el sistema fuente(origen), transformar los datos, y la carga en un sistema de destino (el sistema de destino no necesariamente tiene que ser un almacén de datos).
Stage Area (area de ensayo)	Almacenamiento temporal de las tablas origen. Su objetivo es no sobrecargar el sistema fuente(origen) con transacciones o consultas.
DQ - Data Quality (Calidad de datos)	Aplica las normas de calidad de control de datos. Cuando no cumplen con la calidad se almacenan en una base de datos Data Quality y se reporta para su revisión y corrección.
DDS - Dimensional Data Store	Los datos están organizados en un formato de dimensiones que es más adecuado para el análisis. Contiene data integrada de varios sistemas fuente(origen).
Metadata – Metadatos	Es una base de datos que contiene información sobre la estructura de datos, el significado de datos, el uso de datos, los datos de la normas de calidad y otra información acerca de los datos.

Audit Control - Sistema de Auditoria ETL	Supervisa las actividades operacionales de la ETL los procesos y los registros de sus estadísticas operacionales. Se utiliza para la comprensión de lo que pasó durante el proceso de ETL.
MDB – Multidimensional Database	Repositorio final de DDS, es una forma de base de datos donde se almacenan los datos en celdas y la posición de cada celda es definida por una serie de variables llamadas dimensiones. Cada celda representa un evento de negocios, y los valores de las dimensiones muestran dónde, con quién y cuándo ocurrió dicho evento.

16.2 | Características del Sistema Data Warehouse

Las características clave de un Data Warehouse son las siguientes:

1. **La integración o consolidación de los datos:** Los datos se transforman en la integración que toma aspectos como su codificación, medidas, rangos, convenciones de nombres entre otros.

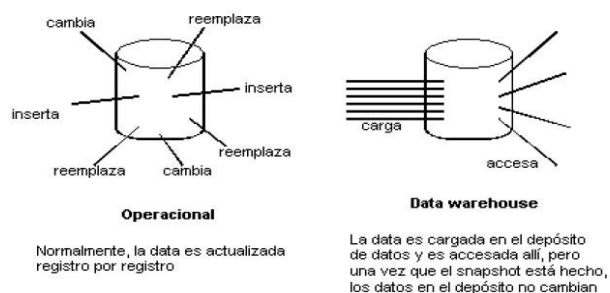


Cuando los datos se mueven a la data Warehouse desde las aplicaciones orientadas al ambiente operacional, los datos se integran antes de entrar al depósito.

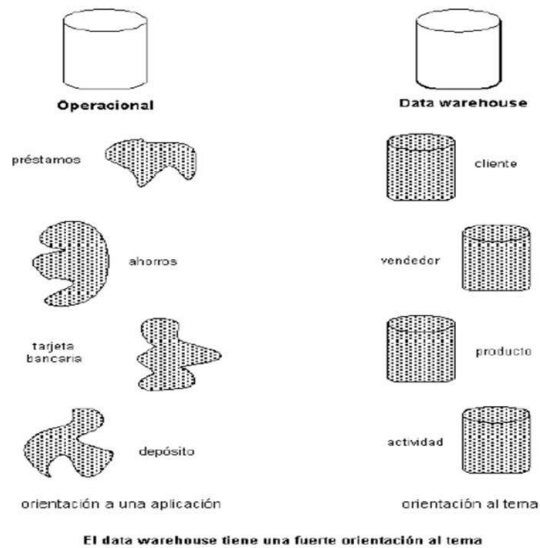
2. **Acceso transversal a datos actuales e históricos:** El almacenamiento de datos actuales y larga data histórica, permitirá la consulta en cualquier punto del tiempo, por ello los datos en los Data Warehouse se denominan de tiempo variante.



3. **La periodicidad en las actualizaciones:** La actualización del sistema se realiza por lotes (batches) y no por cada transacción equivalente en los OLTP, la periodicidad en la actualización se establecerá de acuerdo a los requerimientos del negocio y como mínimo la frecuencia de actualización tomará como la base de frecuencia de actualización de los OLTP, si por ejemplo los sistemas transaccionales tienen una frecuencia de actualización de una vez por día no tiene sentido que el Data Warehouse actualice más de una vez por día.
4. **Información no volátil estable a lo largo del tiempo:** La información es útil cuando es estable, las actualizaciones del Data Warehouse se obtienen por carga inicial de datos en los procesos de extracción, no se permite la actualización rutinaria por parte de los usuarios.



5. **La orientación a temas:** Los almacenes de datos se construyen para efectuar consultas en ellos, estas consultas son orientadas a temas o aspectos (sujetos) del negocio, por ejemplo, se desea consultar las ventas anuales por región.



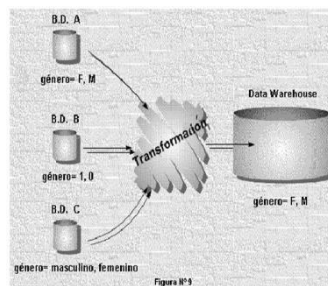
16.3 | Integración con ETL y la Transformación de Datos

ETL significa extracción, transformación y carga. Es el proceso para **extraer** datos desde bases de datos y/o archivos operacionales, y **transformar** los datos antes de **cargar** los resultados en el *Data Warehouse*.

Uno de los desafíos de cualquier implementación del *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 fuentes de datos contribuyen al sistema Data Warehouse.

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.

Extraer los datos desde una o varias bases de datos operacionales y transformarlos en datos requeridos para cargar al depósito o Data Warehouse se conoce en el contexto de **la integración o consolidación de los datos**.



Enfoques y Arquitectura ETL

Existen diferentes enfoques para implementar los procesos ETL. El enfoque tradicional es el de extraer los datos desde los sistemas fuente, colocarlos en un área de ensayo, luego transformarlos y cargarlos en el depósito de datos final. De manera alternativa, otro enfoque consiste en realizar las transformaciones en memoria y cargarlos directamente en el depósito de datos sin usar área de ensayo.

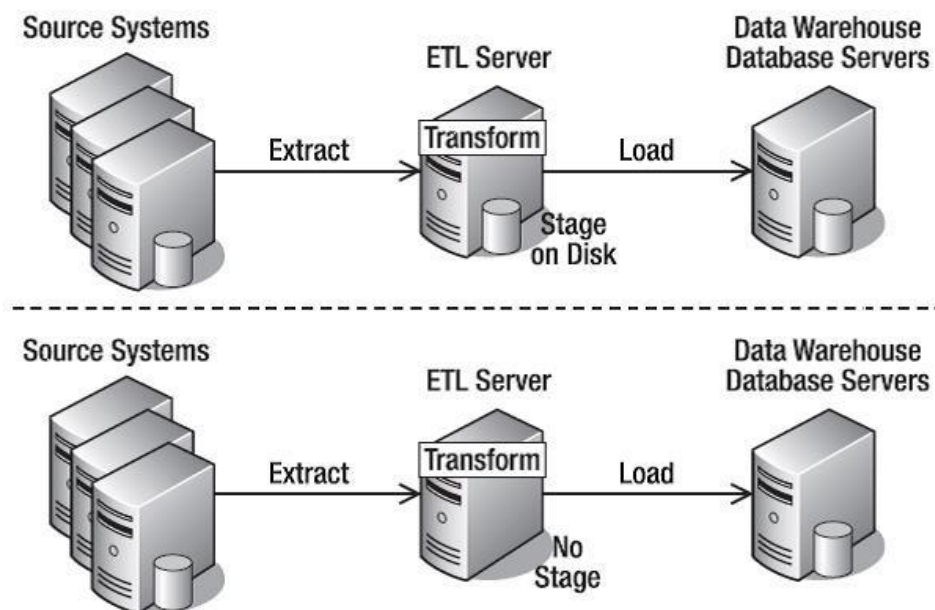


Figura 7. Integración en base de datos de ensayo o integración directa al Data Warehouse

Otra arquitectura alternativa consiste en un proceso de extracción, carga y transformación, ELT (Extract, Load and Transform, por sus siglas en inglés), en el enfoque ELT se extraen los datos de origen, se cargan al Data Warehouse y luego se realizan las transformaciones en el servidor destino. La decisión de usar el enfoque ELT tiene que ver con las capacidades del servidor del Data Warehouse.

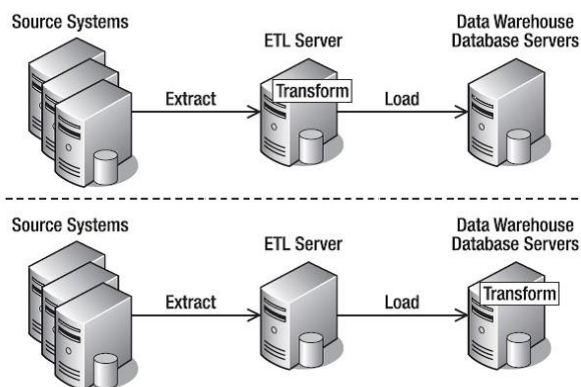


Figura 8. ETL y ELT: opciones sobre donde realizar las transformaciones.

En términos de quién mueve los datos desde los sistemas fuente y dónde se realiza el proceso ETL, las figuras 9 y 10 ilustran los enfoques que pueden utilizarse.

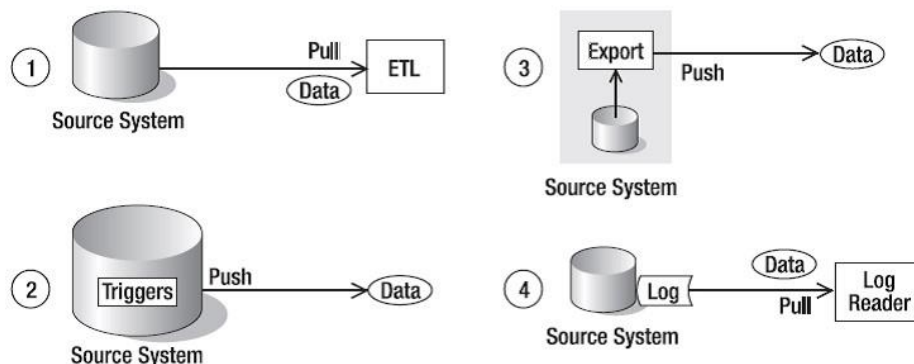


Figura 9. 1) Enfoque tradicional de extracción de los datos; 2) Uso de disparadores desde los sistemas fuente; 3) Exportando regularmente desde los sistemas fuente; y 4) Acceso al log transaccional para extraer los cambios.

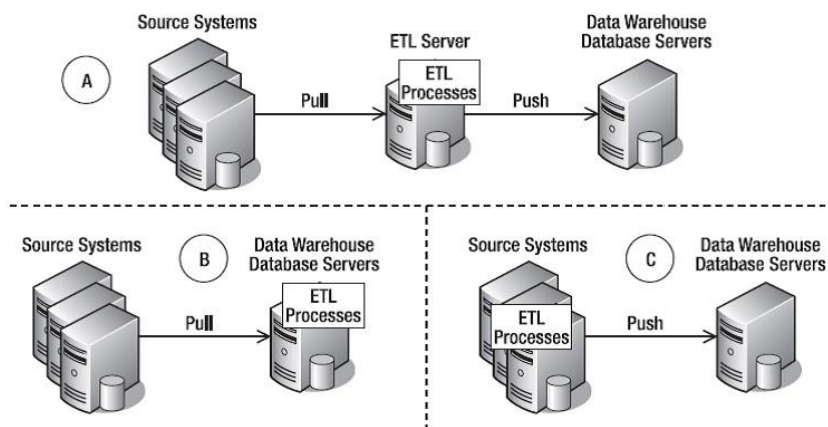


Figura 10. A) Ejecutar el proceso ETL en un servidor separado -pull/push; B) Ejecutar el proceso ETL en el servidor del MDB -pull; y C) Ejecutar el proceso ETL en los sistemas fuente -push.

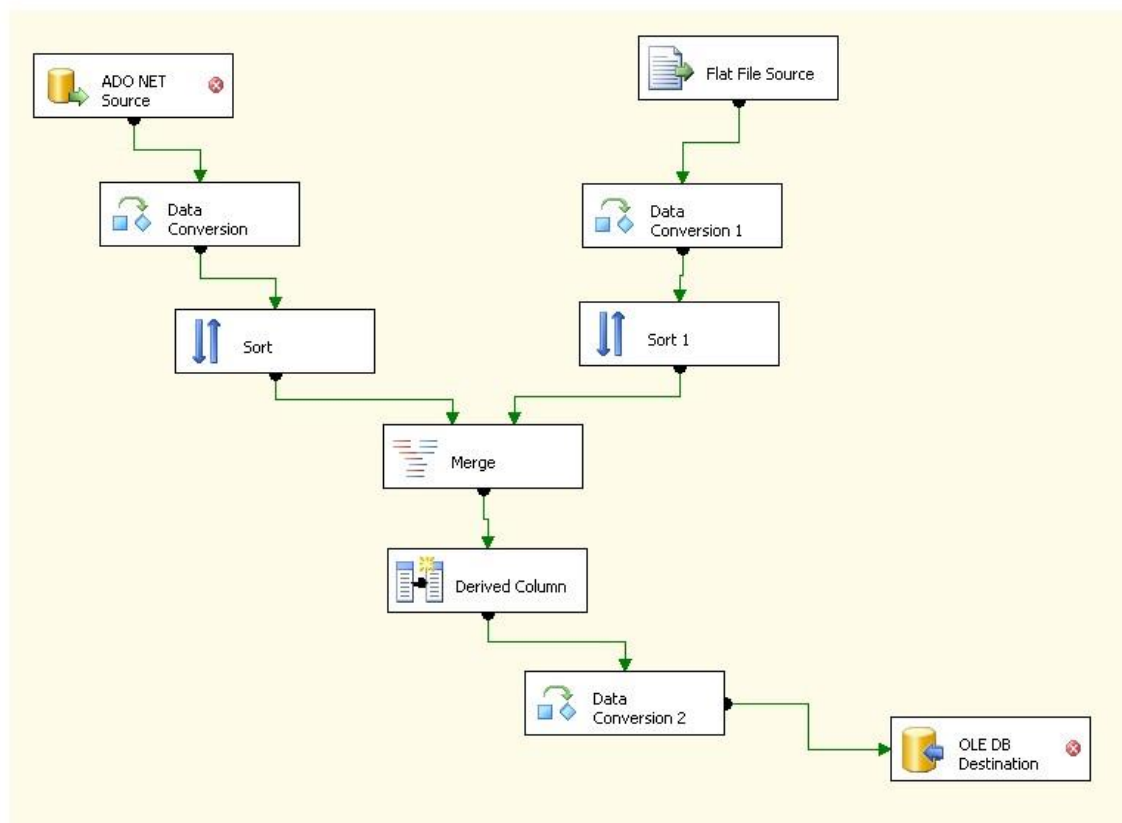
Finalmente, la decisión de usar determinado enfoque del ETL tiene que ver con las capacidades del servidor ETL y del servidor del Data Warehouse además de los costos involucrados por licenciamiento de servidores adicionales necesarios.

Paquetes ETL

El proceso ETL se suele manejar con herramientas administrativas en forma de unidades de procesamiento denominados **paquetes ETL**, el paquete encapsula las tres operaciones ETL.

Los paquetes permiten entre otras cosas administrar los formatos y convenciones en los sistemas fuente y también en los sistemas destino, pero lo más relevante consiste en el acoplamiento del flujo de transformaciones que pueden tomar forma de conversiones, búsquedas, sustituciones,

agregaciones, eliminaciones, entre otras muchas más rutinas de transformación previo a la carga consistente de datos y consolidación del **Data Warehouse** como tal.

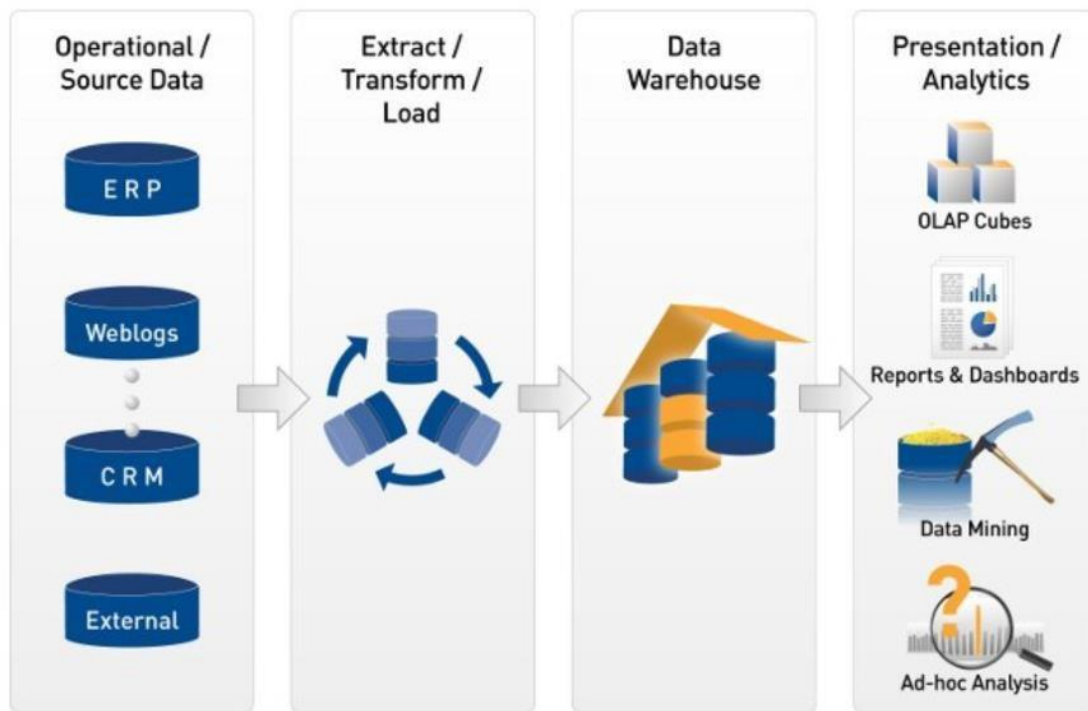


Ejemplo de flujo de transformaciones en un paquete ETL que integra dos orígenes de datos diferentes, desarrollado en Microsoft BI Development Studio.

Como ya se dijo, las bases de datos operacionales diseñadas para el soporte de varias aplicaciones de producción, frecuentemente difieren en sus formatos, por lo que la operación de **transformación** se considera el eje principal en el proceso ETL. No deberían utilizarse paquetes ETL que busquen solamente extraer datos desde orígenes y cargarlos inmediatamente a un destino.

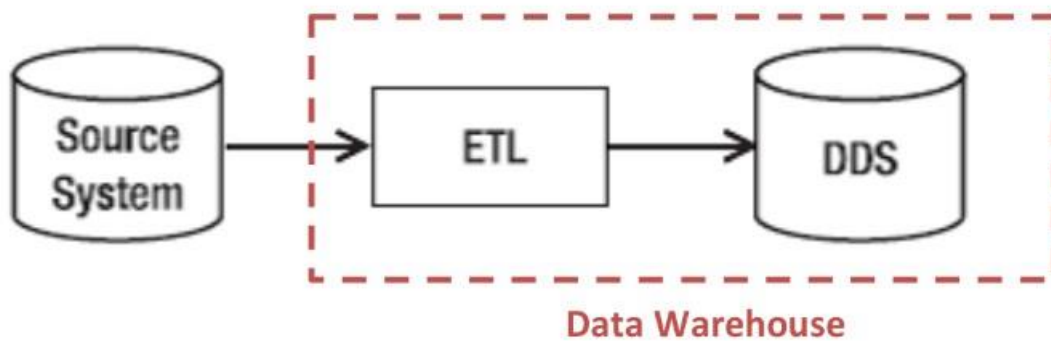
Los procesos ETL han estado presentes desde hace décadas, se han ido desarrollando y mejorando mucho desde su introducción.

Data Warehouses y BI



¿Qué es un Data Warehouse?

Un sistema que recupera y consolida datos periódicamente desde otros sistemas fuente hacia un depósito de datos normalizado o dimensional. Usualmente mantiene años de histórico y es consultado por la inteligencia de negocios y otras actividades analíticas.



¿Por qué un Data Warehouse?

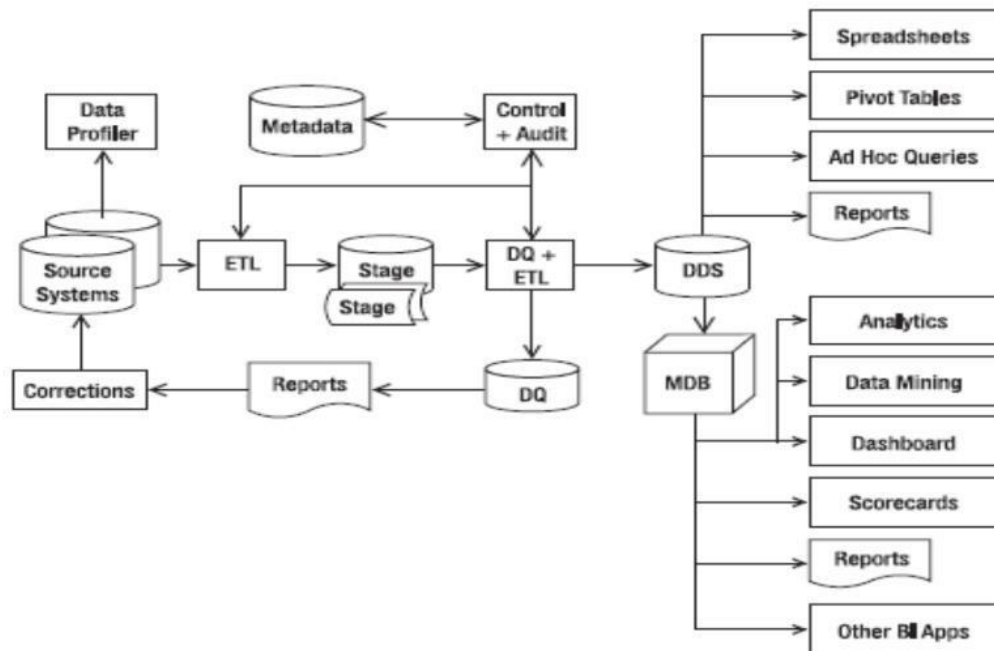
- La necesidad de consultar información histórica y resumida, incluso concentrada desde varios sistemas fuente. Resulta poco práctico hacerlo en cada sistema transaccional.
- Los sistemas y bases de datos transaccionales están orientados como aplicaciones del día al día, no hay cabida allí para realizar la función de un data warehouse. principio: ¡separación de intereses!

¡Las organizaciones necesitan saber! Analizar los negocios, entender cómo y dónde.

¿Qué no es un Data Warehouse?

- No es una base de datos transaccional (OLTP)
- No está orientada a la aplicación; en un data warehouse se orientan los datos a aspectos o temas (dimensiones) sobre los mismos:
 - ✚ Productos, vendedores, clientes, proveedor, sucursal, etc.
- No es un sistema que cambia continuamente. Además, contiene información resumida que no se encuentra en el entorno operacional.

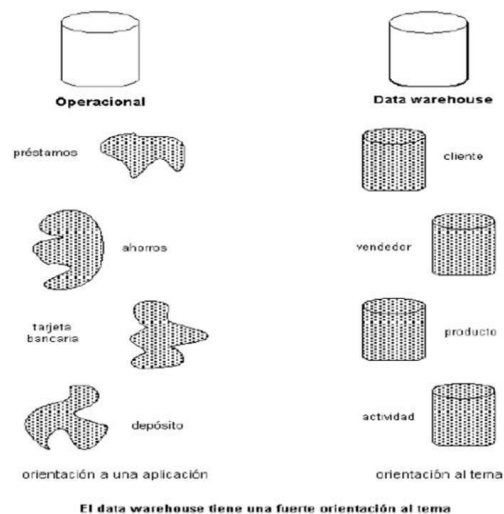
Diagrama de un sistema completo de Data Warehouse



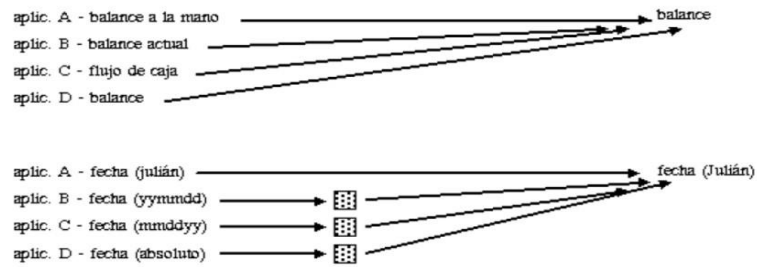
Características de un Data Warehouse

- ✚ Orientado a temas
- ✚ Integración o consolidación de los datos
- ✚ Acceso transversal a los datos históricos y actuales
- ✚ La periodicidad en las actualizaciones
- ✚ Información no volátil estable a lo largo del tiempo

Orientation a temas

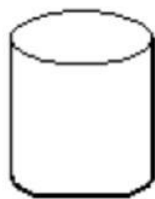


Integración y consolidación de los datos



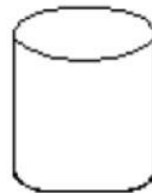
Cuando los datos se mueven al data warehouse desde las aplicaciones orientadas al ambiente operacional, los datos se integran antes de entrar al depósito.

Acceso transversal a datos actuales e históricos



Operacional

De tiempo variante



Data warehouse

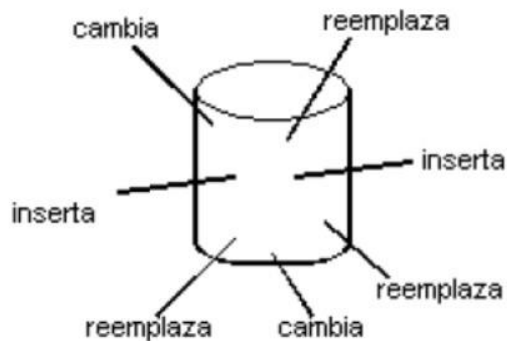
Valor actual de los datos:

- Horizonte de tiempo: 60-90 días
- La clave puede, como no, tener un elemento de tiempo
- Los datos pueden ser actualizados

Datos instantáneos:

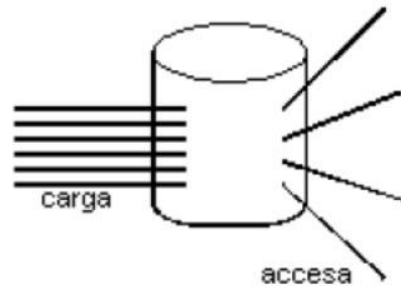
- Horizonte de tiempo: 5-10 años
- La clave contiene un elemento de tiempo
- Una vez que el snapshot se realice, el registro no puede ser actualizado

Información no volátil estable



Operacional

Normalmente, la data es actualizada registro por registro



Data warehouse

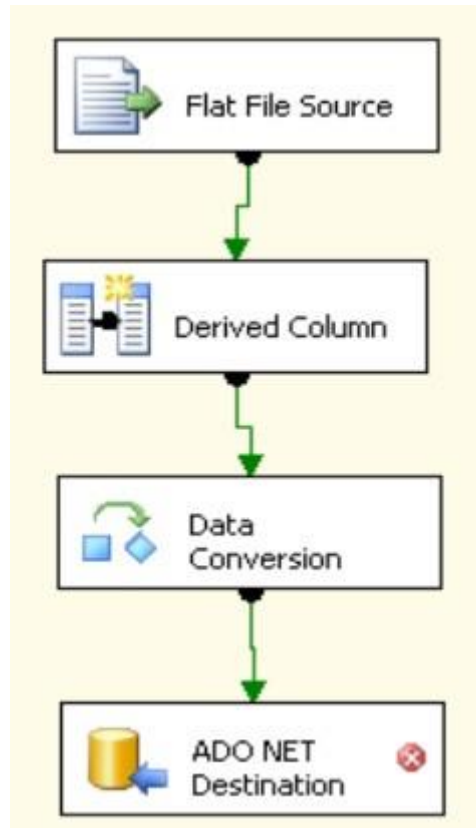
La data es cargada en el depósito de datos y es accesada allí, pero una vez que el snapshot está hecho, los datos en el depósito no cambian

Integración de los datos - ETL

- ETL significa **extracción, transformación** y carga.
- Es el proceso para extraer datos desde bases de datos y/o archivos operacionales, y transformar los datos antes de cargar los resultados en el Data Warehouse.
- 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 fuentes de datos contribuyen.

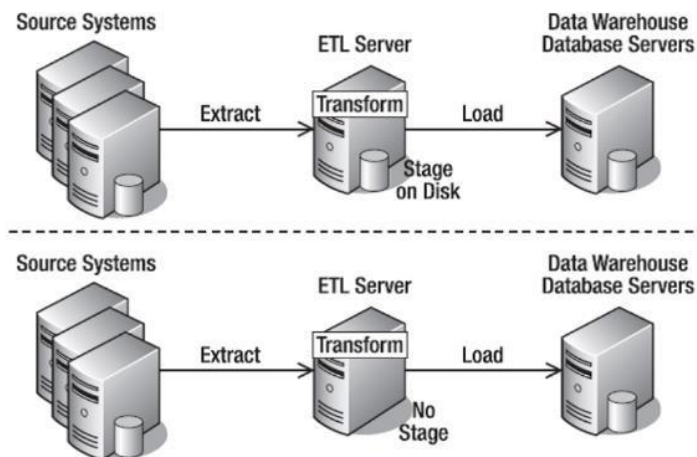
Paquetes ETL

En las herramientas de integración los paquetes ETL permiten llevar a cabo el flujo de procesos de extracción, transformación y carga desde sistemas fuente hacia los sistemas destinados.



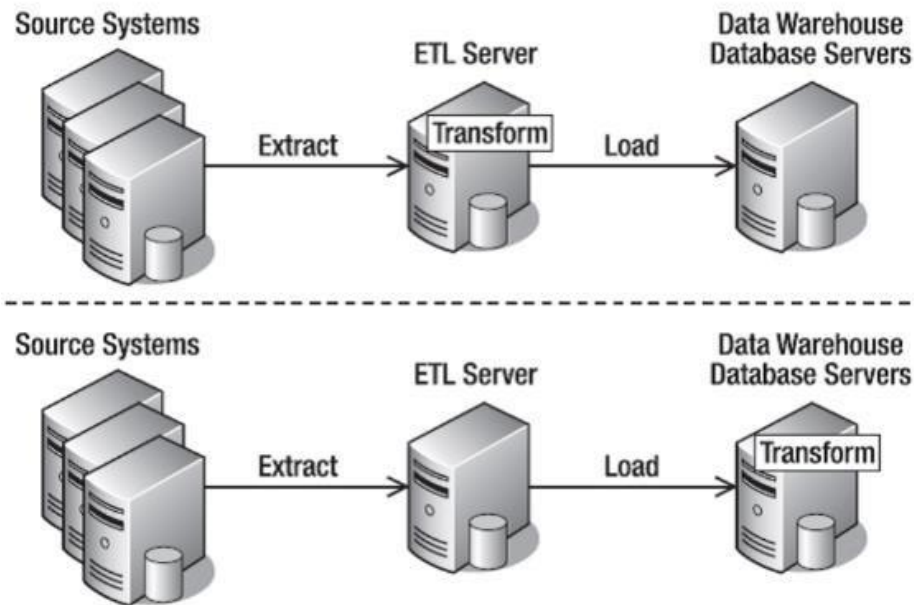
Ejemplo de un paquete ETL en un proyecto de integración con SSIS.

Enfoques y arquitecturas ETL: Área de ensayo



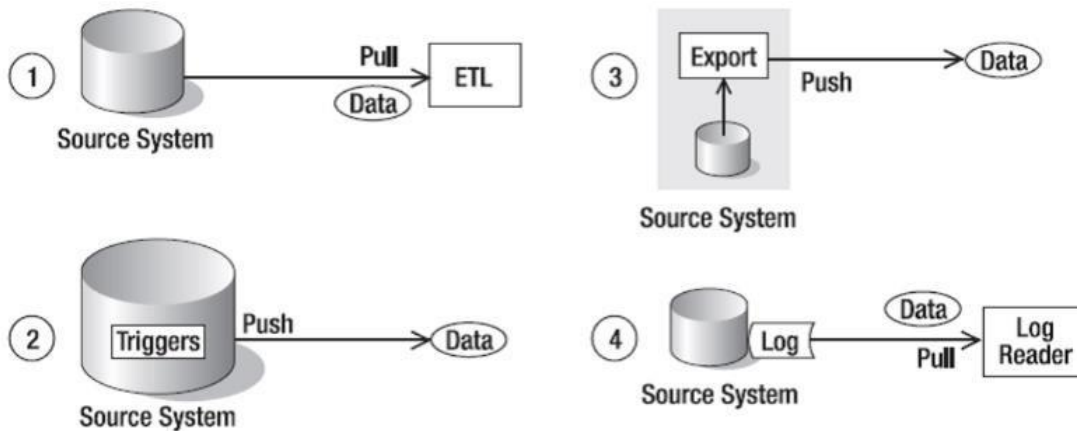
Integración en base de datos de ensayo o integración directa al Data Warehouse.

Enfoques y arquitecturas ETL: ¿Dónde realizar las transformaciones?



ETL y ELT: opciones sobre donde realizar las transformaciones

Enfoques y arquitecturas ETL: ¿Quién mueve los datos?



1. Enfoque tradicional de extracción de los datos
2. Uso de disparadores desde los sistemas fuente

3. Exportando regularmente desde los sistemas fuente y
4. Acceso al log transaccional para extraer los cambios

17.1 | Procesamiento Analítico en Línea, OLAP

OLAP se refiere a la actividad de procesamiento analítico con bases de datos multidimensionales, no así a la base de datos multidimensional misma. Los conjuntos de datos de la base de datos multidimensional se denominan **cubos OLAP**.

Los cubos OLAP se suelen representar mediante una rejilla o matriz que permite hacer las proyecciones dimensionales correspondientes: cada celda representa eventos de negocios mesurables, los **eventos** se interpretan a partir de su localización definida por un número de jerarquías dimensionales.

La figura 12 ilustra con un ejemplo la representación de un cubo OLAP que incluye un número de jerarquías **dimensionales** a decir: tiempo organizado en año, cuatrimestre y mes; producto organizado con categoría de producto.

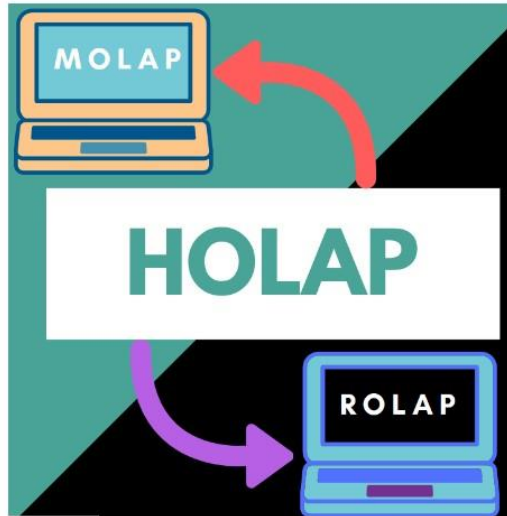
		Year Name ▼	Quarter Name ▼	Month Name ▼	
		Calendar 1996	Calendar 1997	Calendar 1998	Grand Total
Category Name ▼	Product Name ▼	Unit Price	Unit Price	Unit Price	Unit Price
☐ Beverages		2033.8	5058.35	4719.5	11811.65
☐ Condiments		705.6	2211.85	1687.85	4605.3
☐ Confections		1333.4	3547.63	2668.27	7549.299999999999
☐ Dairy Products		1857.6	5070	2948.2	9875.8
☐ Grains/Cereals		522.6	2337.2	1304.5	4164.3
☐ Meat/Poultry		1294.3	3795.48	2327.55	7417.33
☐ Produce		726.6	2376.55	1683.3	4786.45
☐ Seafood		936.3	3218.02	2136.46	6290.78
Grand Total		9410.2	27615.08	19475.63	56500.9100000001

Ejemplo ilustrativo de un cubo OLAP con jerarquías dimensionales

17.2 | Eventos, dimensiones y cubos OLAP




¿Pero, que son eventos y que son dimensiones?

Vamos por partes, primero repasemos brevemente un poco la que ya abordamos en las secciones previas. Las bases de datos dimensionales existen desde el tiempo en que comenzamos a trabajar con el concepto de almacenes de datos dimensionales (Dimensional Data Store, DDS por sus siglas en inglés) o Data Warehouse como comúnmente les conocemos. Este concepto ha evolucionado a lo que hoy conocemos como bases de datos multidimensionales (Multidimensional Database, MDB por sus siglas en inglés) en el contexto del procesamiento analítico en línea. A decir, una evolución en forma y fondo, los motores de bases de datos multidimensionales pueden ser tan diferentes de un motor de bases de datos relacional tradicional con el fin de favorecer el rendimiento de OLAP.



Motores de implementación Nativo OLAP, OLAP Relacional e Híbrido.

La estructura lógica de una base de datos multidimensional desde el punto de vista de su modelación sigue las mismas reglas establecidas desde los tiempos de los DDS, es decir un esquema dimensional puede usar uno de los siguientes tres enfoques:

-  Esquema estrella
-  Esquema de copo de nieve
-  Esquema constelación

Más adelante ampliaremos con ejemplos cada uno de estos enfoques de modelado dimensional.

Como puede apreciarse, la evolución ha favorecido la implementación de tecnologías eficientes para el procesamiento analítico, sin embargo, la esencia se mantiene tal y como se había anunciado desde los Data Warehouse: su orientación dimensional a través de temas, sujetos o aspectos del negocio, potenciando su extensibilidad dimensional.

Ahora volvemos a las preguntas, entonces **¿Qué son los eventos y dimensiones?** Los eventos o hechos marcan la información factual sobre la que se centra el procesamiento analítico. Estos hechos deben incorporar atributos de medición que caracterizan el hecho mismo; por ejemplo, los ingresos sobre ventas, en cantidad monetaria; la cantidad de pedidos, en unidades, etcétera. Si el elemento escogido no es medible no es un hecho.

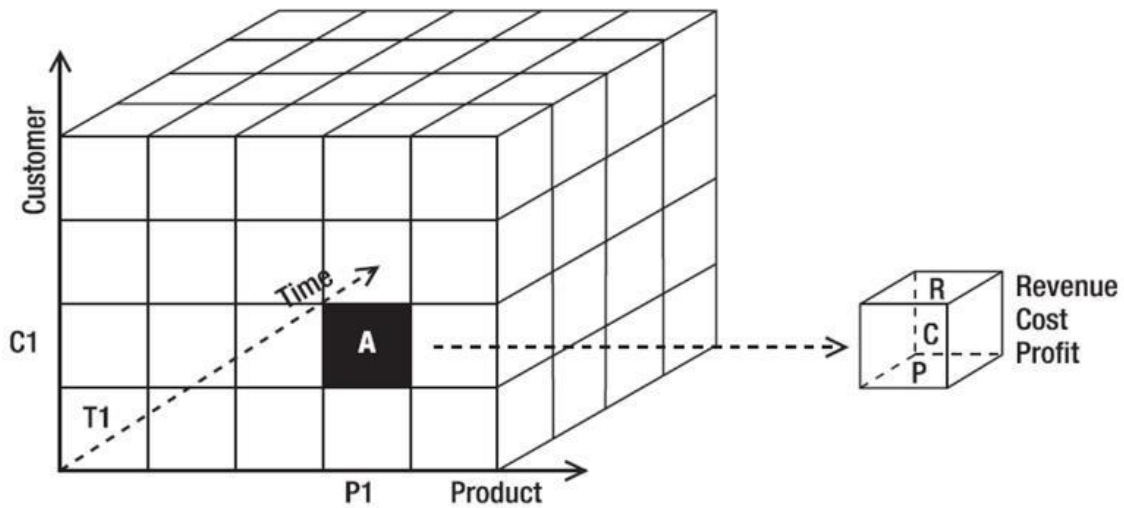


Figure 12-2. Visualization of a multidimensional database with three dimensions

- **Dimension - Dimensiones:** Customer (cliente), Product (producto), Time (tiempo)
- **Fact - Hechos – Eventos:** Revenue (Ingresos), Cost (costo), profit (ganancia)

Dimension – Dimesiones Customer: (cliente), Product (producto), Time

Fact –Hechos –Eventos Revenue: (Ingresos), Cost (costo), profit (gnancia)

Por otro lado, las **dimensiones** ayudan a entender, interpretar y proyectar los hechos desde distintas perspectivas: los clientes que les vendemos, los productos que vendemos o que ordenamos, los proveedores a quienes les ordenamos, los vendedores, el tiempo en que se realiza los eventos etcétera.

Dimensiones: Cliente, vendedor, producto, tiempo, etc.




Las dimensiones pueden y deben jerarquizarse cuando corresponda, por ejemplo la dimensión producto podrá organizarse por clase de producto y esta a su vez, en categoría.

La dimensión de tiempo es obligatoria en la determinación de los cubos OLAP, también podrá jerarquizarse cuando convenga por ejemplo determinar el tiempo de la realización de los eventos de negocios en años, cuatrimestres, meses, días etcétera.

17.3 | Modelado de bases de datos multidimensional

Un modelo o esquema dimensional es la base para la constitución de la estructura lógica para el almacenamiento de bases de datos multidimensionales.

Los esquemas para las bases de datos multidimensionales son, como ya se dijo, los siguientes:

-  Esquema de estrella
-  Esquema de copo de nieve
-  Esquema de constelación

A continuación, brindamos una muy breve explicación de cada uno de ellos junto con un ejemplo ilustrativo.

Esquema de estrella

En el esquema de estrella, existe una única tabla central que modela los eventos de negocios, alrededor de ésta se concentran todas las tablas dimensionales sin más niveles.

Las características, en resumen, del esquema en estrella son:




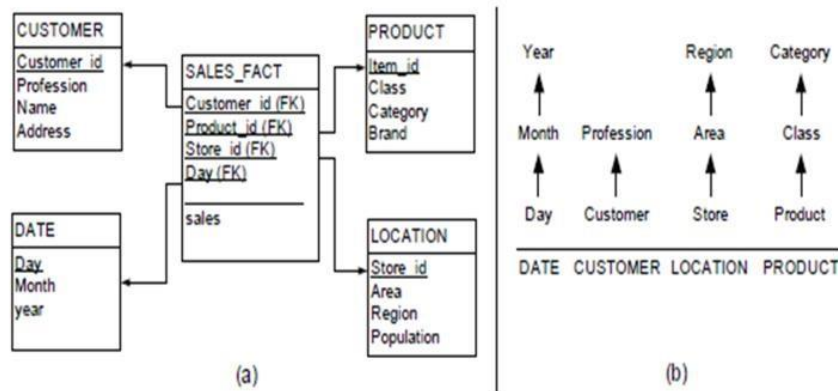
-  Una única tabla central de hechos.
-  Una o más tablas dimensionales de primer nivel, que marcan dependencia directa hacia la tabla de hechos.
-  Si existen jerarquías en las dimensiones deberán incluirse en las tablas dimensionales de primer nivel.

Figure 1. (a) The schema of the data warehouse; (b) the dimension hierarchies of the example

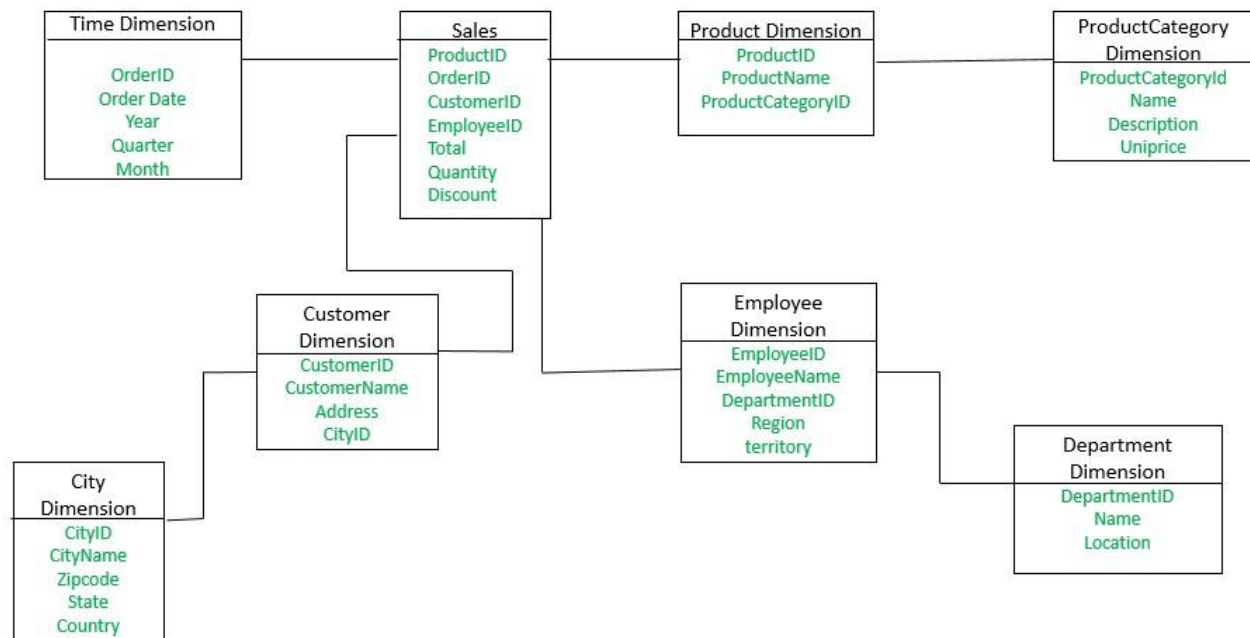


Esquema de copo de nieve

En el esquema de copo de nieve, existe de igual forma una única tabla central que modela los eventos de negocios, alrededor de ésta se concentran tablas dimensionales de primer nivel, a partir de estas podrán conectarse tablas de segundo nivel o más niveles para modelar apropiadamente la jerarquía dimensional.

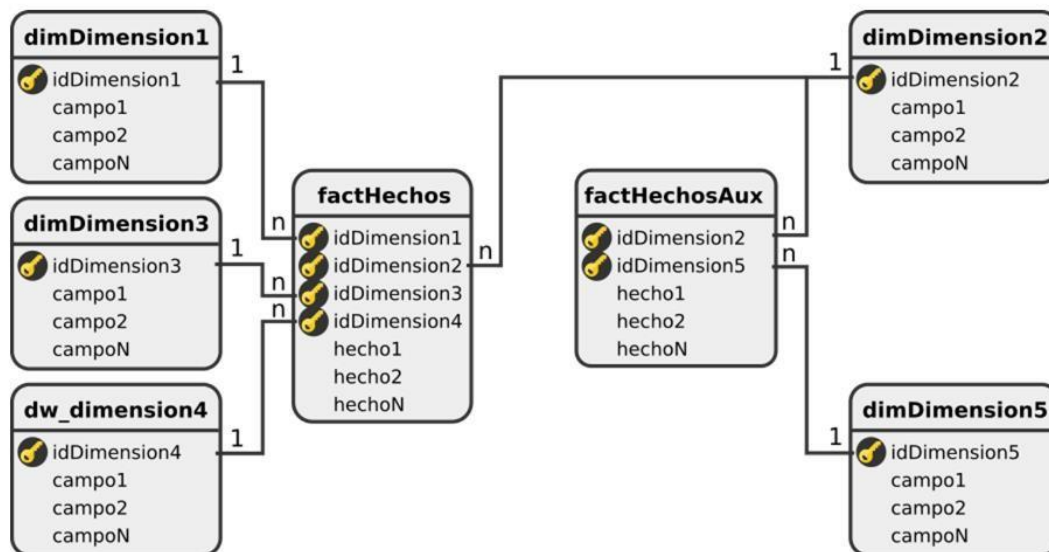
Las características, en resumen, del esquema en copo de nieve son:

- ✚ Una única tabla central de hechos.
- ✚ Una o más tablas dimensionales de primer nivel, que marcan dependencia directa hacia la tabla de hechos.
- ✚ Una o más tablas dimensionales de segundo nivel, tercer nivel o más, que representan apropiadamente la jerarquía dimensional.

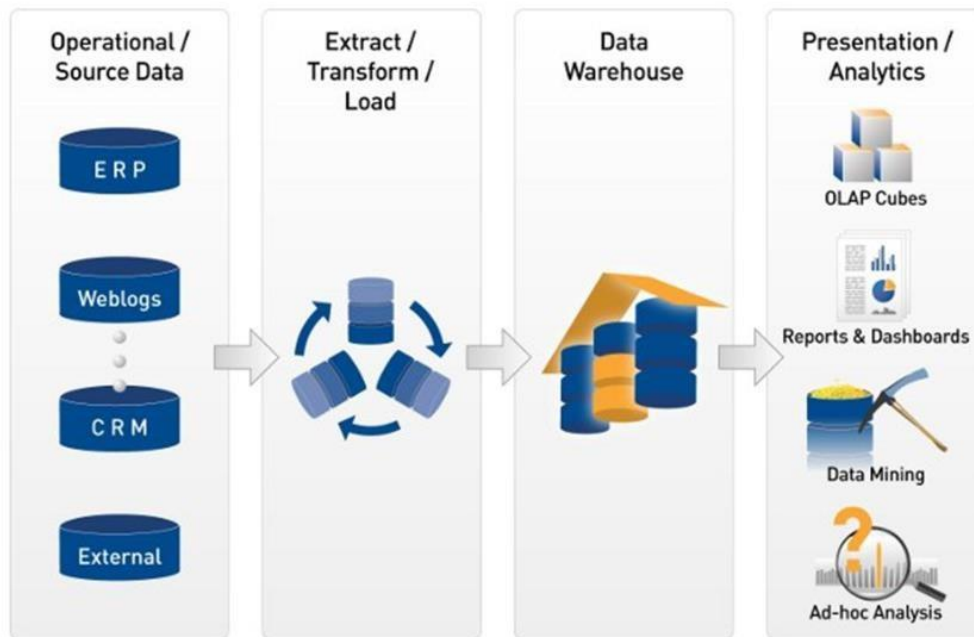


Esquema de constelación

En el esquema de constelación, básicamente fundimos los dos esquemas anteriores, el de copo de nieve y el de estrella, para formar una representación que permite más de una tabla de hechos y múltiples niveles de tablas dimensionales.



OLAP en el BI



Bases de datos multidimensionales

Una **base de datos multidimensional** es una forma de base de datos donde los datos se almacenan en celdas y la posición de cada **celda** es definida por un número de jerarquías llamadas dimensiones. Cada una de las celdas representan eventos o hechos del negocio, y el valor de las dimensiones muestra cuándo, con quién y dónde el evento ocurrió. Cada evento se cuantifica mediante un conjunto de medidas.

Análisis OLAP

OLAP se refiere a la actividad de procesamiento analítico en línea para con las bases de datos multidimensionales; se ancla en el análisis de cubos OLAP de información.

Un cubo OLAP consiste en una celda que retiene hechos localizados a partir de dimensiones que fijan un arreglo multidimensional circundante. Un cubo OLAP es una pieza de la base de datos multidimensional.

Los hechos o eventos marcan la información factual sobre la que se centra el análisis. Estos hechos deben incorporar atributos de medición que caracterizan el hecho mismo; por ejemplo, los ingresos sobre ventas, en cantidad monetaria.

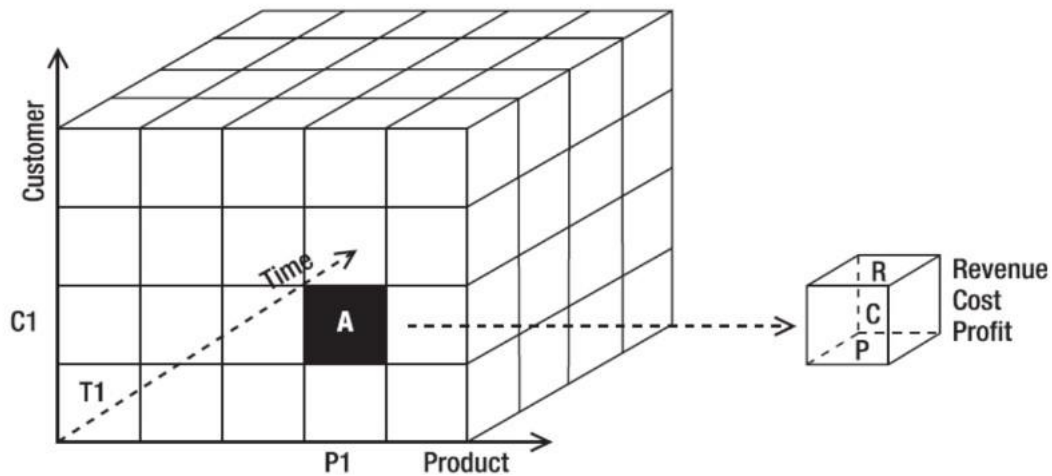


Figure 12-2. Visualization of a multidimensional database with three dimensions

- **Dimension - Dimensiones:** Customer (cliente), Product (producto), Time (tiempo)
- **Fact - Hechos – Eventos:** Revenue (Ingresos), Cost (costo), profit (ganancia)

Análisis OLAP: Dimensiones

Las dimensiones ayudan a entender y proyectar los hechos desde distintas perspectivas o vistas: □

Cliente, vendedor, producto, tiempo, etc.,

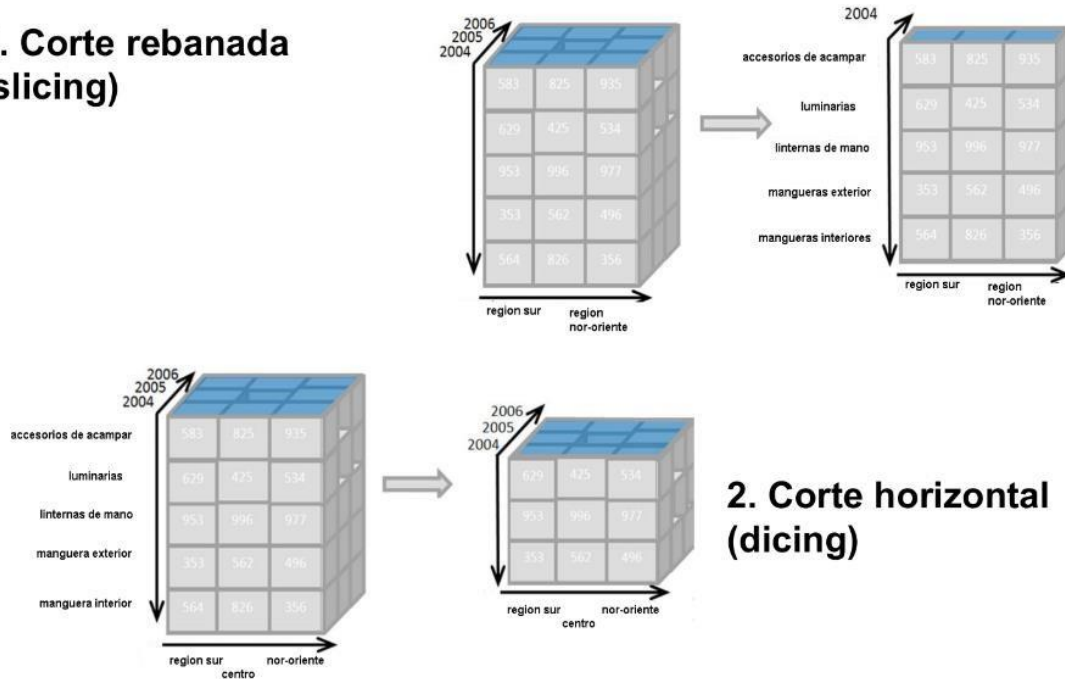
✚ La dimensión de tiempo es obligatoria en la determinación de los cubos.

Las dimensiones pueden y deben jerarquizarse cuando corresponda:

- ✚ Tiempo diario, mensual, semestral, anual.
- ✚ Producto Categoría.

Cubos OLAP: Operaciones (1)

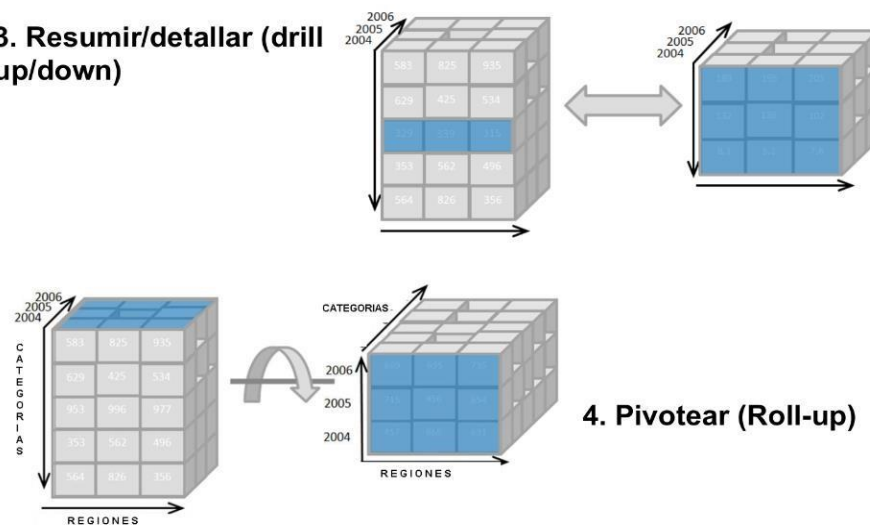
1. Corte rebanada (slicing)



2. Corte horizontal (dicing)

Cubos OLAP: Operaciones (2)

3. Resumir/detallar (drill up/down)



4. Pivotar (Roll-up)

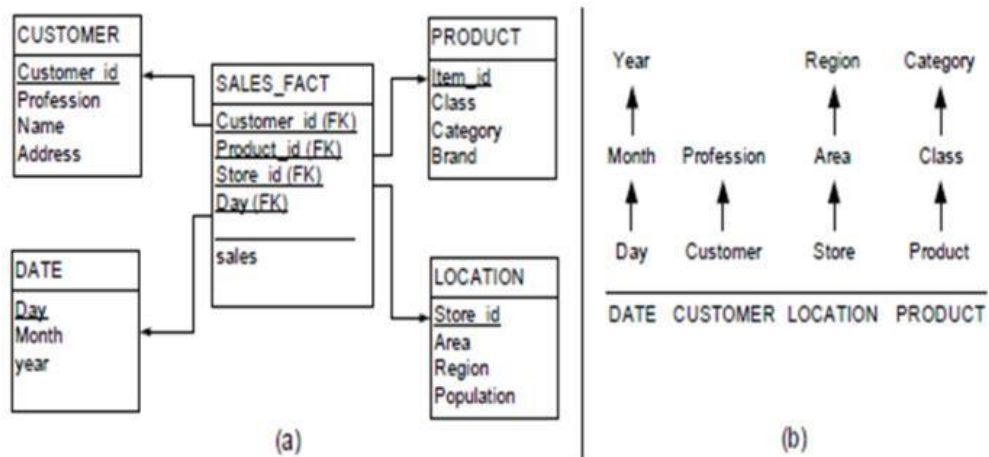
Modelado dimensional

Un modelo o esquema dimensional es la base para la constitución de la estructura lógica para el almacenamiento de bases de datos multidimensionales.

Los esquemas para las bases de datos multidimensionales son los siguientes:

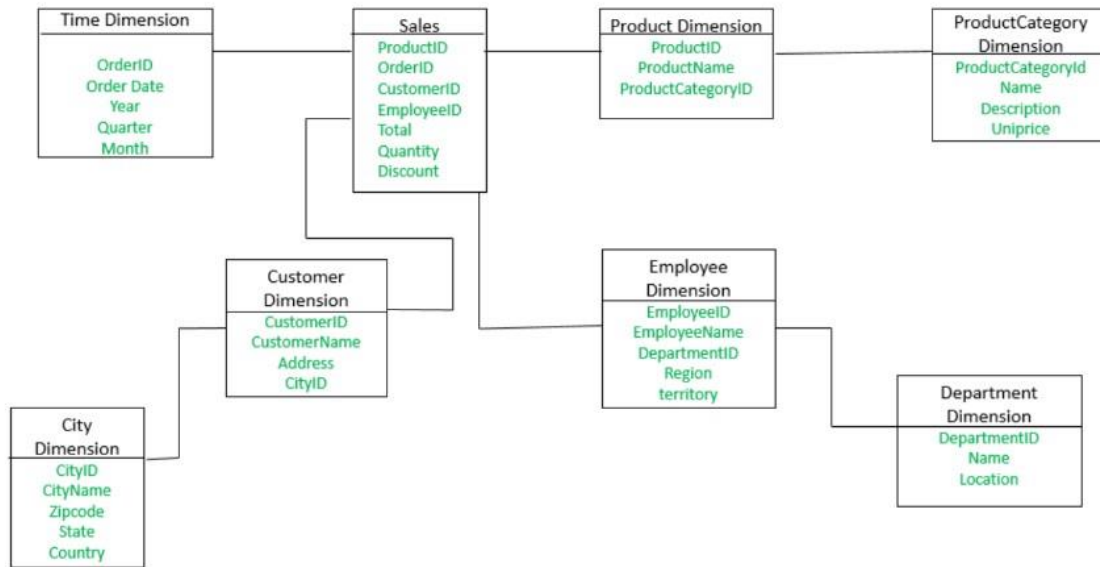
- Esquema de estrella
- Esquema de copo de nieve
- Esquema de constelación

Esquema de estrella



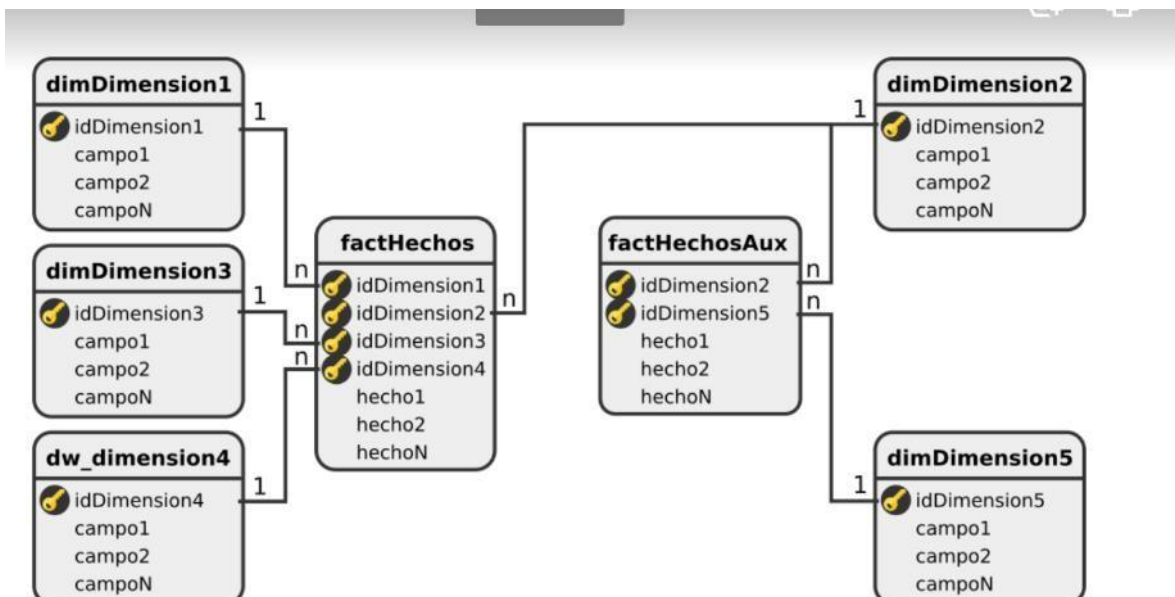
- En este modelo existe una única tabla de hechos y solo un nivel para las tablas dimensionales, las jerarquías dimensionales se deben incluir en el primer nivel con sus claves surragadas.

Esquema de copo de nieve



- En este modelo sigue utilizándose una única tabla de hechos pero pueden existir más de un nivel en las tablas dimensionales.

Esquema de constelación



- ✚ Es un modelo híbrido respecto de los dos anteriores, que permite utilizar más de una tabla de hechos y cualquier número de niveles en tablas dimensionales.

Implementación OLAP

La implementación OLAP puede darse con distintos motores, destacando el motor nativo multidimensional que es mucho más ligero y con buena performance. La implementación BDM puede ser en cualquiera de los siguientes motores:

- ✚ **MOLAP** – Motor nativo no relacional, matricial.

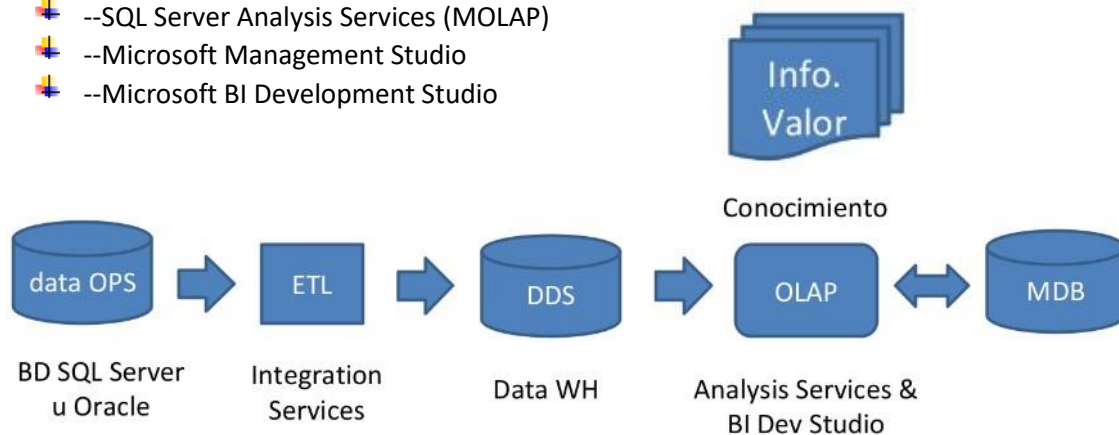
- ✚ **ROLAP** – Motor relacional, ajustado a las reglas de los SGBDR

HOLAP – Motor híbrido.

Reuniendo todo para el BI

Herramientas necesarias

- ✚ --SQL Server Integration Services
- ✚ --SQL Server Analysis Services (MOLAP)
- ✚ --Microsoft Management Studio
- ✚ --Microsoft BI Development Studio



CUESTIONARIO DE BASE DE DATOS II

1. ¿Cuál es la diferencia primordial entre dimensiones y hechos en la configuración de un cubo?

Respuesta: Las dimensiones representan proyecciones y sujetos para el análisis dimensional.

2. ¿Cuál es la diferencia entre el esquema constelación y el esquema copo de nieve para el modelado multidimensional?

Respuesta: El esquema constelación solo usa una tabla de hechos y el esquema snowflake puede usar varias tablas de hechos.

3. ¿Una base de datos multidimensional es una data warehouse?

Respuesta: Si, es un tipo de DW en el que la información se representa y almacena de forma diferente.

4. ¿Cuáles son motores de implementación para bases de datos multidimensionales?

Respuesta: HOLAP, ROLAP, MOLAP.

5. Una jerarquía de dimensiones representa solo un nivel de dimensión en el modelo o esquema dimensional.

Respuesta: Falso

CUESTIONARIO II

1. La dimensión tiempo no es considerada una dimensión mínima obligatoria en todo cubo OLAP

Falso

2. Marque las que no apliquen como dimensiones

Ordenes

Proveedores

Envíos

Regiones

3. OLAP se refiere a:

La actividad del procesamiento analítico con las MDB

La base de datos multidimensional

Procesamiento analítico en línea

Una rejilla o matriz multidimensional almacenada

4. Marque a las que aplique como grupo de medidas

Regiones

Proveedores

Ordenes

Pedidos

5. Cuales aplican como beneficios del Business Intelligence para las empresas

Ayuda con la automatización de procesos

Le permite a la empresa contar con una herramienta de cuadro de mando integral

Le permite a la empresa desarrollar una gestión de calidad optima

Contar con conocimiento refinado sobre procesos de negocios claves

6. OLAP no se refiere a:

La actividad del procesamiento analítico con las MDB

La base de datos multidimensional

Procesamiento analítico en línea

Una rejilla o matriz multidimensional almacenada

7. El desafío principal de cualquier implementación de Data Warehouse es el problema de:

Cargar los datos

Integrar datos

Extraer los datos

Transformar los datos

8. Marque las que no aplique como competencias del BI

Consultas directas

Dashboard

Hojas de calculo

OLAP y minería de datos

9. Modelado dimensional que faculta la jerarquización de dimensiones

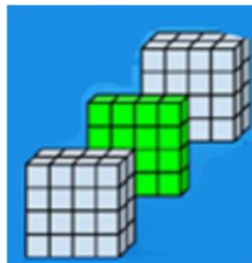
Modelo de Estrella

Modelo de Constelación

Modelo Entidad Relación

Modelo Copo de Nieve

10. Marque que operación analítica de cubo OLAP representa la imagen a continuación



Corte de rebanas slicing

Corte horizontal dicing

Resumir detallar (drill up/down)

Pivotear (Roll up)

11. Esquema dimensional que permita una sola tabla de hechos

Entidad-relación

Estrella

Copo de nieve

Constelación

12. En los paquetes ETL la extracción es la función relevante del proceso, sin esta no existe la verdadera integración

Falso

Verdadero

13. ETL se refiere a

La transformación de datos

Extraer, transformar y cargar

Proceso de integración y consolidación de los datos

Ensayo de datos

14. Observe la siguiente imagen a continuación y seleccione el tipo de operación que está siendo aplicada en el cupo

		Trimester Name ▼			
		Trimester 2, 2019		Total general	
Categoría ▼	Producto ▼	Cantidad Req	Precio	Cantidad Req	Precio
⊞ Abarrotes de Cocina		6	130	6	130
⊞ Herramientas		6	70	6	70
⊞ Ropa y Accesorios		8	174	8	174
Total general		20	374	20	374

Corte de rebana (Slicing)

Corte horizontal (Dicing)

Resumir detallar (Drill up/down)

Pivotear (Roll up)

15. Marque el orden que corresponde para que con la cadena de valor agregado en la inteligencia de negocios

Datos -> información -> conocimiento

Conocimiento -> datos -> información

Información -> datos -> conocimiento

Información -> conocimiento -> datos

16. Marque las que aplique como características de una data Warehouse

Orientación a procesos

La información cambia constante

Almacenamiento de datos históricos

Orientación a aspectos y sujetos de los negocios

17. Subconjunto de almacenes de datos enfocados en aspectos del negocio específicos

Data marts

Cubo

OLAP

Orígenes de datos

18. Etapa sobre la metodología en donde se observan los productos del BI

II etapa, sobre la recuperación de los datos de origen

III etapa, sobre integración

IV, etapa sobre resultados y aplicaciones...

I, etapa sobre la elección de los sistemas de origen

19. El o los sistemas de origen no son parte realmente del contexto del sistema de data warehousing

Verdadero

Falso

20. Un sistema para análisis multidimensional que se implementa completamente con base de datos multidimensionales almacenando los datos como celdas en matrices o rejillas

MOLAP

HOLAP

ROLAP

WOLAP

21. Un sistema de data Warehouse cuya arquitectura no incluye un espacio para almacenamiento temporal de los datos de origen

Integración al DDS con área de ensayo

Integración directa DDS

Sistema push/pull por separado

Sistema ETL

22. Marque competencias del data warehousing tradicional

Data mining

Tabla pivot

OLAP

Consultas SQL

23. Un sistema de data warehouse como mínimo debe incorporar

Integración ETL

ERP

DDS

Servicio OLAP

24. Explique por qué se necesitan y usan claves sustitutas jerárquicas (hsk) al integrar la data warehouse. Ilustre con un ejemplo.

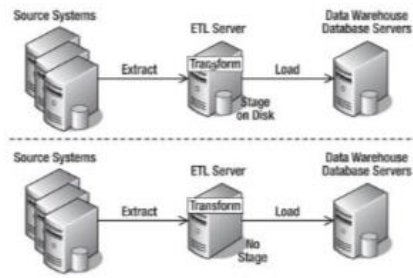
25. Marque el argumento que es correcto respecto a la integración ETL

La integración ETL es el proceso utilizado para cargar y utilizar la data warehouse

La integración ETL permite mantener un almacén de datos históricos

26. Marque en el área de la imagen la dimensión que no se encuentra jerarquizada

27. Marque el área de imagen que corresponda con una integración con ensayo



28. Investigue sobre la big data y marque las diferencias con respecto a expectativas del BI