Bases de Datos

Tema 06. Otros modelos de datos

Marta Elena Zorrilla Pantaleón

Rafael Duque Medina

DPTO. DE MATEMÁTICAS, ESTADÍSTICA Y COMPUTACIÓN

Tabla de contenido

Bibliografía

- Básica
  - Connolly y Begg (2005): Sistemas de Bases de Datos.
  - Kimball, R., Ross, M. The data warehouse toolkit: the complete guide to dimensional modelling. John Wiley & Sons, cop. 2002
  - Piattini et al. (2006): Tecnología y Diseño de Bases de Datos.
  - Silberschatz, A., Korth, H.F., Sudarshan, S., Fundamentos de Bases de, 6ª edición, Madrid, 2011.
- Complementaria
  - Abiteboul et al. (1999): Data on the Web. From Relations to Semistructured Data and XML. Morgan Kaufmann.
  - Elmasri y Navathe (2007): Fundamentos de Sistemas de Bases de Datos.
  - Inmon, W. H. Building the Data Warehouse. Willey & Son. 2002.

Modelos basados en Objetos – introducción

- Las tecnologías de BD, desde su aparición en los 70, se ha caracterizado por su excepcional productividad y un impresionante impacto económico
- GENERACIONES DE BASES DE DATOS

  1ª SGDB MODELOS JERÁRQUICOS Y EN RED

  Década de los 70. IMS de IBM y el IDMS de Cullinet

  2ª SGBD RELACIONALES

  Década de los 80. ORACLE, DB2, INGRES, INFORMIX, SYBASE, SQL Server, etc.

  3ª SGBD ACTIVOS, ORIENTADOS A OBJETOS, XML, etc.

  - Proporcionan capacidades de **gestión de datos** al igual que sus predecesoras; **gestión de objetos**, permitiendo la definición de tipos de datos más complejos y encapsulamiento de la semántica de los datos, así como otras nuevas capacidades. Y algunos proporcionan incluso **gestión de conocimiento**, soportando un gran número de reglas complejas para inferencia automática de información y mantener las restricciones de integridad entre datos.
  - Década de los 90. ORACLE, DB2,SYBASE, etc.

Modelos basados en Objetos – introducción (y 2)

- Los productos de la 1ªgeneración proporcionaban soluciones a los problemas de tipo administrativo (gestión de personal, reserva de plazas, etc.), pero resultaban inadecuados para responder a consultas no planificadas, debido a la falta de independencia ya mencionada y a sus interfaces de bajo nivel.
- La llegada de los productos relacionales cambia esta situación e incrementa los campos de aplicación de las bases de datos.
- Pero estos no responden a las nuevas necesidades de gestión de datos:
  - Tratamiento y recuperación de información textual
  - Almacenamiento y gestión de información geográfica
  - Aplicaciones médicas (reconocimiento de patrones, seguridad,…)
  - Publicación digital (aplicaciones multimedia)
  - Aplicaciones científicas (sensores, biomédicas, etc.)
  - Otras

  por lo que surge la 3ª generación de SGBD que incorpora las capacidades de orientación a objetos, disparadores, nuevos tipos de datos, etc.

Debilidades de los SGBD Relacionales

El MR adolece de las siguientes debilidades que en el modelo objetual se resuelven:

- Representación pobre de las entidades del "mundo real": el proceso de normalización generalmente lleva a la creación de relaciones que no corresponden con entidades del "mundo real". La fragmentación de una entidad del "mundo real" en varias relaciones es ineficiente llevando a muchos "joins" en el procesamiento de consultas.
  - Por ejemplo, un libro tiene un título, varios autores, una editorial y un conjunto de palabras clave. En el MR se requieren tres entidades (libro, libro-autores, libro-palabras) mientras que en modelo OO sería un objeto con 4 atributos, 2 de ellos multivaluados.
- Sobrecarga semántica: un solo constructor, la relación, para representar tanto los datos como las relaciones entre ellos (una relación 1 a N, puede tener varias representaciones semánticas tiene, gestiona, posee,…).
- Soporte limitado de las restricciones de integridad y de negocio.

Debilidades de los SGBD Relacionales (y 2)

- Estructura de datos que no permite heterogeneidad: el modelo relacional asume homogeneidad horizontal y vertical. Cada tupla tiene los mismos atributos. Los valores en un una columna deben pertenecer al mismo dominio. La intersección de fila y columna debe ser un valor atómico. como por ejemplo distancia o intersección entre objetos de un plano.
- Operaciones restringidas: operaciones de conjuntos y orientadas a tuplas, operaciones que provee SQL. Pero SQL no permite definir nuevas operaciones
- Manejo complicado de consultas recursivas: Es difícil manejar consultas sobre relaciones (relationships) que una relación (relation) tiene consigo misma
- Desajuste de impedancia (en el SQL embebido): debido a que se mezclan diferentes paradigmas de programación. SQL es un lenguaje declarativo y C es un lenguaje procedural (uso de cursores). SQL provee tipos de datos que no tienen los lenguajes tradicionales (Interval, Date).

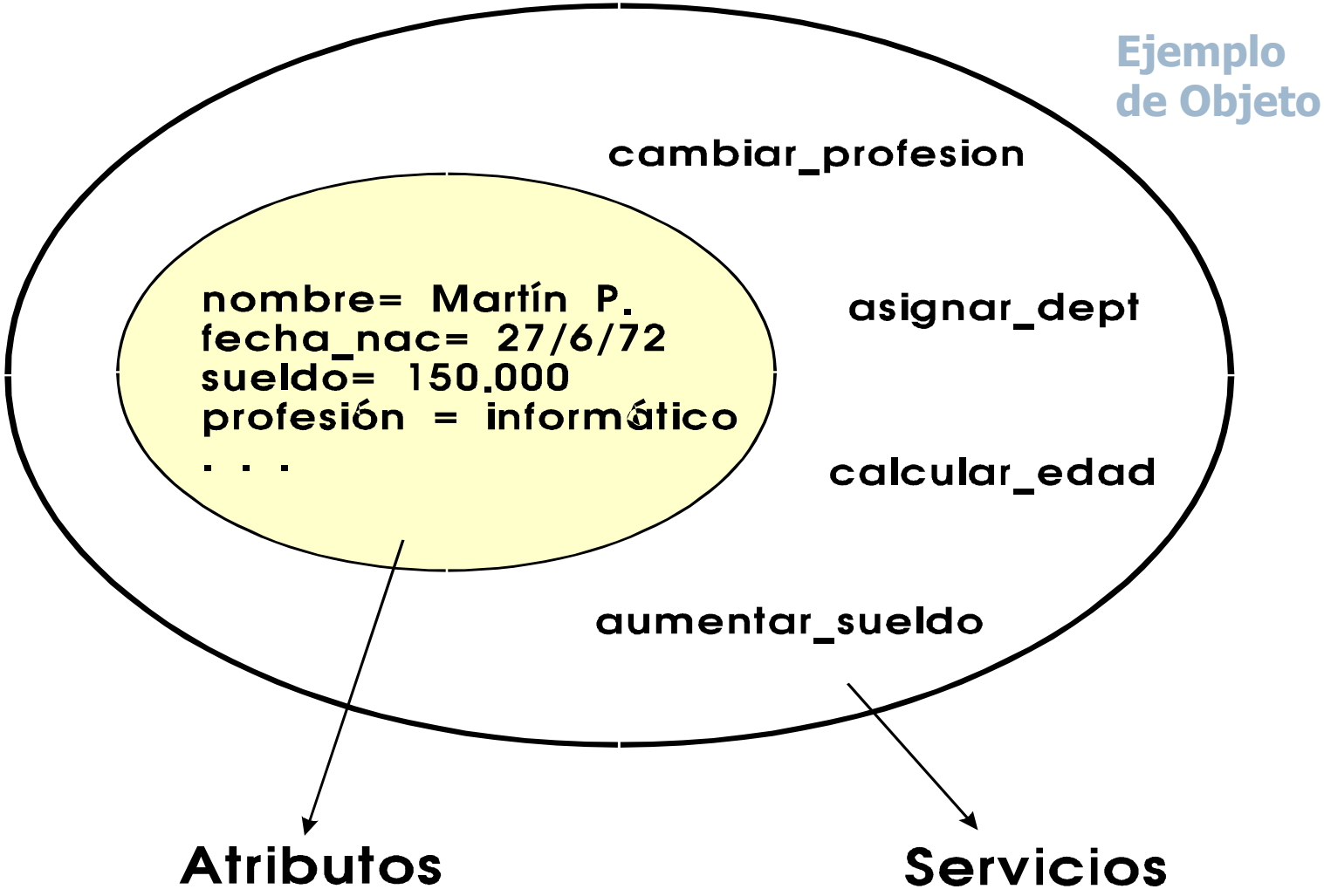
Debilidades de los SGBD Relacionales (y 3)

- Mal soporte de las transacciones de larga duración: que suelen ser más comunes para objetos complejos.
- Evolución de esquemas más difícil: los administradores deben intervenir en el cambio de la estructura de la BD, y los programas que acceden a tales estructuras deben ser modificadas para reflejar estos cambios de estructura. Se necesitan sistemas que permitan una evolución natural de los esquemas.
- Pocas facilidades para navegar por los datos: acceso asociativo basado en el contenido y no basado en el movimiento entre registros individuales (acceso navegacional como OQL).

Modelos basados en Objetos – conceptos básicos

- Aplicaciones complejas requieren tipos de datos complejos (atributos multivaluados, registros anidados, herencia, operaciones específicas, etc.)
- El Modelo OO sigue el paradigma objetual de los lenguajes de programación con algunas limitaciones en su implementación
- Sistema Orientado a Objetos (OO):
  - Conjunto de objetos que se comunican entre sí mediante mensajes.
- Objeto:
  - Conceptualmente, es una entidad percibida en el sistema.
  - Un objeto se describe por sus propiedades, también llamadas atributos - estructura del objeto- y por los servicios que puede proporcionar - comportamiento del objeto-.
  - El estado de un objeto viene determinado por los valores que toman sus atributos, valores que siempre han de cumplir las restricciones impuestas sobre ellos.

Modelos basados en Objetos – conceptos básicos (y 2)



Modelos basados en Objetos – conceptos básicos (y 3)

- Características de los Objetos:
  - Poseer un estado que puede cambiar.
  - Tener una identidad única.
    - Identificador de Objeto (OID): Característica especial que lo identifica unívocamente.
  - Soportar interrelaciones con otros objetos.
  - Poseer un comportamiento.
  - Poder recibir y enviar mensajes.
- Tipo de Objeto:
  - Clasificación de objetos, que define la estructura y el comportamiento de objetos que comparten características comunes.
- Clase:
  - Implementación de un tipo de objeto. Los objetos son los "ejemplares" o instancias de las clases.

Modelos basados en Objetos – interacciones entre objetos

- **Estáticas**:
  - Generalización: los tipos y las clases se organizan en jerarquías o retículos de super/subtipos (super/subclases) que presentan herencia y que corresponden al concepto "**es-un**", en las cuales los subtipos (o subclases) heredan los servicios o atributos de sus ancestros.
  - Agregación: permite construir objetos compuestos o complejos, corresponde a la noción "**parte-de**".
- **Dinámicas**:
  - Mensajes: sirven para que los objetos soliciten la prestación de servicios y entreguen, en su caso, los resultados que se obtienen cuando se lleva a cabo un cierto servicio.
  - Un mensaje es una petición desde un objeto (remitente) a otro objeto (receptor) para que el segundo ejecute alguno de sus métodos.

Modelos basados en Objetos – abstracción

- **Abstracción**
  - Proceso de identificar los aspectos esenciales e ignorar el resto.
  - Implica centrarnos en qué es y qué hace un objeto antes de pensar en cómo implementarlo.
  - Tiene dos aspectos fundamentales:
    - **Encapsulación**: un objeto contiene la estructura de datos y el conjunto de operaciones que pueden ser usadas para manipularla.
    - **Ocultamiento**: Los aspectos internos de un objeto están ocultos al exterior. Provee una forma de independencia de datos.

- Estos mecanismos suponen una manera de realizar **modularización**: un objeto es una "**caja negra**" que puede ser construida y modificada independientemente del resto del sistema, respetando que el interfaz externo no cambie.

Modelos basados en Objetos – polimorfismo

- **Polimorfismo**: capacidad de que un mensaje sea interpretado de maneras distintas, según el objeto que lo recibe. Existen dos tipos:
  - De subclase: cuando un servicio definido en una clase se redefine en alguna de sus subclases manteniendo el mismo nombre. Entonces un mensaje enviado a un objeto que pertenece a una cierta clase de la jerarquía puede invocar cualquiera de estos servicios, según sea la clase a la que pertenezca el objeto que lo recibe.
  - De sobrecarga: utilizando el mismo nombre para servicios distintos, no situados en una jerarquía de generalización.
  - El polimorfismo necesita que la vinculación (*binding*) entre el nombre de un servicio y su implementación se establezca en tiempo de ejecución (vinculación dinámica o tardía).

SGBD basados en objetos

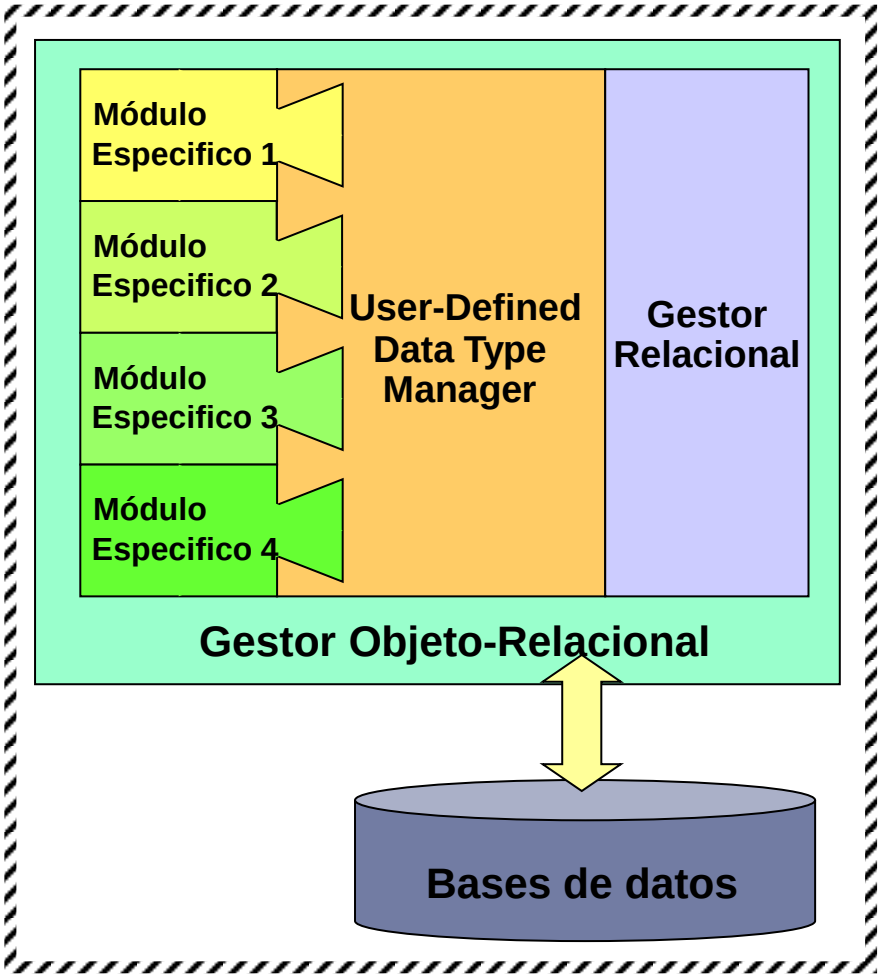
- Dos alternativas
  - SGBD-OR
    - Extienden el modelo relacional para que sean capaces de soportar los conceptos de la orientación al objeto (arrays, multisets, tipos, tablas tipadas,…).
    - Ej: ORACLE, IBM DB2, SYBASE,...
  - SGBD-OO
    - Soportan un modelo de objetos puro (no existe modelo universal) al que añaden la persistencia (el programador manipula directamente los datos, lo que lleva a que es más fácil producir errores que dañen la BD aunque, por otra parte, se consigue mejor rendimiento). Siguen distintas estrategias:
      - Ampliar un lenguaje de programación OO existente con capacidades de BD (GemStone).
      - Proporcionar bibliotecas de clases con las capacidades tradicionales de las bases de datos (persistencia, transacciones, concurrencia,…) Ontos, ObjectStore y Versant
      - Incrustar estructuras de un lenguaje de BD orientado a objetos en un lenguaje host tradicional. Caso de O2, que extiende el C.
      - Ampliar un lenguaje de BD con capacidades OO, caso del SQL 2003 y Object SQL (OQL, propuesto por ODMG)

SGBD basados en objetos (y 2)

- En la actualidad el mercado de SGBD relacionales de objetos es el preferido para dar soporte a nuevos tipos de datos, principalmente por las siguientes razones:
  - Costes: ya tienen mucho dinero invertido en SGBD y aplicaciones como para quitarlos y sustituirlos por sistemas orientados a objetos
  - Dilatada experiencia
  - Lenguaje SQL declarativo
  - Optimización y rendimiento para la operativa transaccional validado, mientras que en SGBDOO la optimización de consultas compromete el encapsulamiento y los bloqueos a nivel de objeto pueden perjudicar el rendimiento.
- Pero también presentan inconvenientes como
  - Mayor coste y complejidad del gestor al dar soporte a la extensión.
  - El SQL se ha vuelto extremadamente complejo .
- Por otra parte, aunque los SGBDOO no tienen mucho éxito comercial con las estrategias actuales es posible que aparezca una nueva generación de estos sistemas usando SGBDOR como backend.

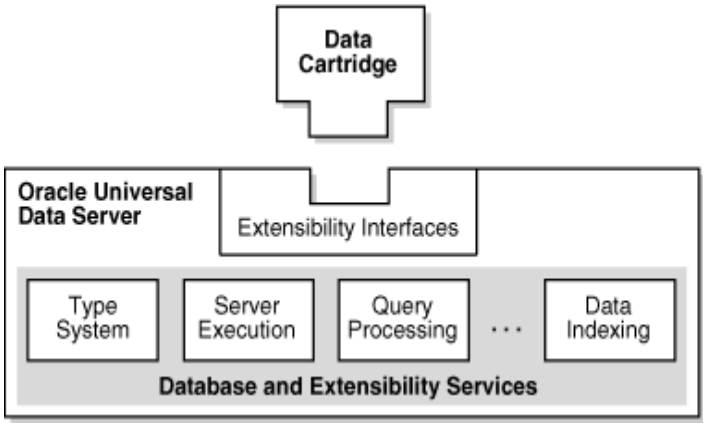
SGBD basados en objetos – Arquitectura OR

- Los SGBD-OR mantienen el núcleo relacional e incorporan el **User-Defined Data Type Manager (UDTM)** que hace de interfaz entre cada uno de los módulos específicos y el RDBMS
- Estos módulos específicos, pueden ser definidos por el usuario o comprados (gestión de textos, imágenes, series temporales, etc.) y una vez que se incorporan al gestor se utilizan como un tipo de dato del sistema



SGBD basados en objetos - Oracle

- Ejemplo de módulos específicos de Oracle:
  - The **Text cartridge** uses the tokenized serial byte stream database model are used to implement display compress, reformat, and indexing behavior.
  - The **Image cartridge** uses the database model for structured large objects to implement compress, crop, scale, rotate and reformat behavior.
  - The **Spatial cartridge** is for use with goemetric objects (points, lines, polygons); it implements project, rotate, transform and map bahavior.
  - The **Video cartridge** uses the structured large object database model to support serial (dynamic) image data compression, play, rewind and pause bahavior.



Resumen

- El **modelo relacional** permite un modelado sencillo, consultas potentes y una protección elevada.
- El **modelo orientado a objetos** es más adecuado para la gestión de tipos de datos complejos, permite la integración con otros lenguajes y presenta un buen rendimiento.
- El **modelo relacional de objetos** conjuga ambos, permite definición de tipos de datos complejos, lenguajes de consulta potentes y protección elevada (gestión de transacciones y recuperación).
- Existen también **sistemas de mapeo OR** que suministran una vista de objetos de datos almacenados relacionalmente. Caso de Hibernate, que proporciona un mapeo OR a Java.
  - Estos son mejor aceptados y más utilizados que los SGBDOO basados en lenguajes de programación persistentes ya que
    - Facilitan la programación de aplicaciones ofreciendo un modelo de objetos y un lenguaje que interroga directamente sobre el modelo de objetos
    - Aunque puede sufrir de sobrecarga y presenta capacidades limitadas de consulta aunque permite utilizar SQL directamente

XML y Bases de Datos

- ¿**Qué son** las Bases de Datos XML?
  - Un documento XML es una BD en el sentido estricto del término porque
    - Almacena información (documentos).
    - Puede responder a un esquema (DTD, XML Schema)
    - Tiene lenguajes de consulta (XPah, XQuery)
    - y API's de Programación (SAX, DOM, JDOM…)
  - **Ejemplos** de posibles BD XML:
    - Fichero de configuración de una aplicación
    - Plantilla de un fax
    - Formulario para solicitar dietas de viajes
    - Temario de una asignatura
    - Todos los informes de un departamento
  - En general, una BD XML es una BD cuya unidad de almacenamiento lógica es el documento XML.

XML y Bases de Datos

- ¿**Por qué** surgen las BD XML?
  - Necesidad de almacenar y recuperar datos poco estructurados con la eficiencia de las BD convencionales.
  - Aparecen dos clases de SGBD que soportan documentos XML
    - **XML-Enabled: desglosan un documento XML en su correspondiente modelo relacional o de objetos.**
    - **XML Nativos: respetan la estructura de documento, permiten hacer consultas sobre dicha estructura y recuperan el documento tal y como fue insertado originalmente**

XML y Bases de Datos – almacenamiento

- Existen varias aproximaciones para organizar y **almacenar documentos XML** de cara a su consulta y recuperación:
  - Usar un SGBD para almacenar los documentos XML como texto.
    - Se almacenan documentos XML completos como textos muy largos en columnas de tipo CLOB (SGBD objeto-relacional) o en objetos de clase texto (SGBD-OO). También se puede utilizar el nuevo tipo de dato XML
  - Usar un SGBD para almacenar los elementos XML de los documentos como elementos de datos.
    - Si todos los documentos XML tienen una estructura basada en un DTD/Schema, es posible volcar sus partes a estructuras relacionales o a objetos de un SGBD (mapeo).
  - Usar un Sistema de BD para almacenar documentos XML de forma directa (BD XML nativa).

Sistemas de BD Nativos XML

"Las bases de datos XML nativas son BD que almacenan XML usando un formato que permite un procesamiento más rápido" (DBXml Group)

- Almacenan la información en formato nativo.
- No traducen el XML a estructuras relacionales u objetos.
- Sus esquemas permiten reglas de almacenamiento e indexación.
- Mantienen el modelo XML intacto.
- Soportan lenguajes de consulta XML y las operaciones se realizan en XML
- No tienen ningún modelo de almacenamiento físico subyacente concreto. Pueden ser construidas sobre BDR, jerárquicas, orientadas a objetos o bien mediante formatos de almacenamiento propietarios.
- Existen retos pendientes para la integridad global de la BD.
  - Integridad referencial inter-documento.
  - Restricciones semánticas inter-documento.

Sistemas de BD Nativos XML (y 2)

- Ventajas
  - No necesitan mapeos adicionales.
  - Conservan la integridad de los documentos.
  - Permiten almacenar documentos heterogéneos en la misma colección.
    - Las colecciones juegan en las bases de datos nativas el papel de las tablas en las DB relacionales
- Ámbito de Uso
  - Documentos con anidamientos profundos.
  - Importancia de preservar la integridad de los documentos.
  - Sistemas con XML orientado a los documentos, más que a datos.
  - Búsquedas de contenido.
- Áreas de Aplicación
  - Portales de información corporativa.
  - Catálogos de Datos.
  - Almacenamiento de información médica.
  - BD de personalización.

Sistemas de BD Nativos XML - productos

- Algunos productos:
  - dbXML - comercial – almacenamiento propietario
  - Virtuoso - comercial – almacenamiento propietario
  - eXist – open source – almacenamiento relacional
  - BirdStep RDM XML – comercial – almacenamiento OO
  - Tamino - comercial - propietario

Integración de XML en otros SGBD

- Inclusión de **XML en SQL**:
  - Nueva parte del estándar para crear y manipular documentos XML.
  - ISO/IEC 9075-14: XML-Related Specifications (**SQL/XML**).
- **SQL/XML** permite:
  - Almacenar documentos XML en bases de datos relacionales y objeto-relacionales.
  - Consultar dichos documentos mediante "XQuery" y "XPath".
  - Publicar o exportar datos objeto-relacionales en forma de documentos XML.
    - Mapeo de Tablas-XML y viceversa.
- Para ello ofrece:
  - Nuevo tipo predefinido.
  - Nuevos operadores y funciones predefinidos para crear y manipular valores de tipo XML.
  - Reglas para mapear tablas, esquemas y catálogos a documentos XML.

Integración de XML en otros SGBD (y 2)

- Soporte de XML en SGBD
  - IBM, Oracle implementan casi por completo el SQL/XML
  - Microsoft soporta similares características pero con sintaxis propietaria
  - Todos soportan XQuery dentro del SQL
  - Existen diferencias en su implementación física (almacenamiento)
    - Oracle 10g basado en CLOB o tablas OR
    - Microsoft 2005 y 2008 almacenado como BLOB en formato interno propietario
    - DB2 V9 basado en CLOB

Almacenes de datos o Data warehouses

- Actualmente, las organizaciones disponen de diversos sistemas de información que les dan soporte a los procesos transaccionales de la organización (ventas, recursos humanos, CRM...).
- La cantidad de información que almacenan es alta y, si se hace un adecuado tratamiento y análisis, puede aprovecharse para la toma de decisiones empresariales (business intelligence)
- Existen diferentes herramientas que se utilizan en este contexto:
  - Aplicaciones analíticas (OLAP)
  - Reporting y querying
  - Data mining
  - Técnicas de visualización de datos
- Pero el núcleo central es el almacén de datos o data warehouse

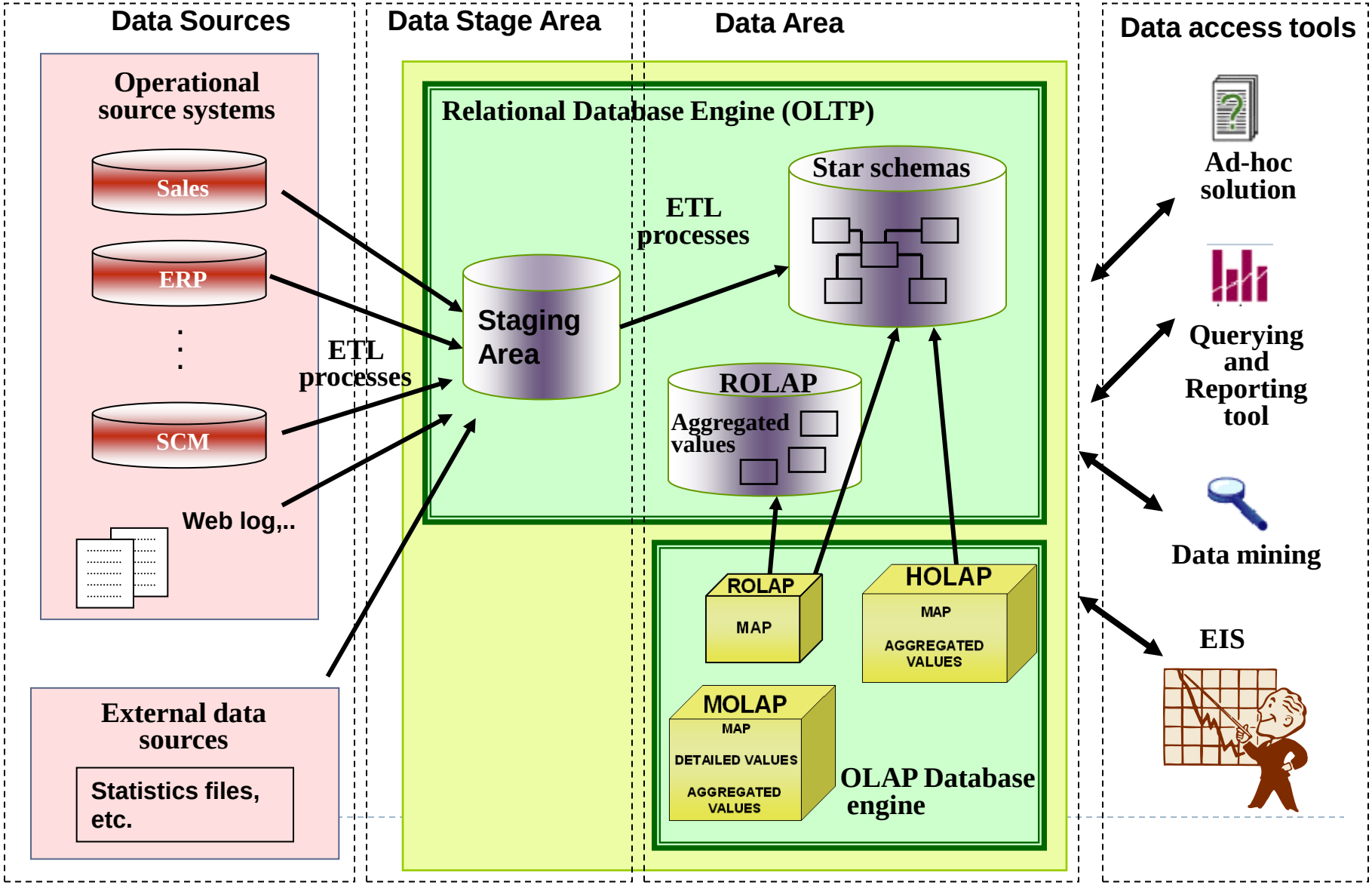
¿Por qué un Data Warehouse?

- Los datos se encuentran en diferentes sistemas de información (uds de medida, convención de nombres y formatos distintos, etc.) que requieren ser integrados
- Estos sistemas no están orientadas a la toma de decisiones (indicadores o medidas de rendimiento de negocio), sino a registrar transacciones (la matrícula, el pedido, etc.).
- La estructura de BD en 3FN no es la adecuada para responder de forma ágil a consultas complejas, con cálculo de agregados y que puedan ser analizadas bajo diferentes perspectivas.
- Por lo que se requiere un sistema de información específico dirigido por las necesidades de los usuarios de negocio, alimentado desde las fuentes de datos operacionales de la organización y construido y presentado desde una perspectiva sencilla

¿Qué es un Data Warehouse?

- Es un sistema de información que:
  - Contiene la información estratégica para la toma de decisiones
  - Se utiliza para analizar datos, detectar tendencias y diseñar estrategias
  - Recoge datos que provienen de diferentes sistemas operacionales (integración), consolidados a una determinada fecha (variante en el tiempo) y centrados en una determinada materia de negocio (ventas, consumos, uso del sitio Web...).
  - Su estructura se diseña para dar respuesta ágil a las consultas y facilitar la distribución de sus datos, no para soportar procesos de gestión.
  - No se actualizan sus datos, sólo son incrementados (no volátil).

Componentes de un DW



Componentes de un DW (y 2)

- Fuentes de datos de donde provienen la información de negocio
- Data Stage: área dedicada a los procesos ETL (*Extraction, Transformation, Load*):
  - extracción de los datos.
  - filtrado de los datos: limpieza, consolidación, etc.
  - carga inicial del almacén
  - sincronización periódica que propaga los cambios de las fuentes externas al almacén de datos
- Área de Datos: almacena los datos para la toma de decisiones. Estos se pueden alojar en un gestor relacional o multidimensional (OLAP) aunque generalmente ambos son usados. El relacional para almacenar la información detallada y la BD OLAP para construir los cubos e incluir las agregaciones y los indicadores cuyo cálculo es complejo.
- Herramientas de Consulta: permiten acceder a los datos y actuar sobre ellos (OLAP, reporting, minería de datos).

Componentes de un DW (y 3)

- Herramientas de consulta:
  - Los sistemas de informes o consultas avanzadas:
    - Utilizan los operadores clásicos: concatenación, proyección, selección, agrupamiento, … (en SQL y extensiones) para mostrar el resultado generalmente de manera tabular.
  - Las herramientas OLAP
    - proporcionan facilidades para “manejar” y “transformar” los datos.
    - **producen otros “datos”** (más agregados, combinados).
    - ayudan a analizar los datos porque producen ***diferentes vistas*** de los mismos utilizando operadores específicos: *drill, roll, pivot, slice & dice*
  - Las herramientas de Minería de Datos:
    - son muy variadas: permiten “extraer” patrones, modelos, descubrir relaciones, regularidades, tendencias, etc.
    - **producen “reglas” o “patrones”** (**“conocimiento”**).

El modelo de datos dimensional

- El área de datos generalmente se diseña según el modelo de datos dimensional
  - en un esquema dimensional se representa una actividad que es objeto de análisis (hecho) y las dimensiones que caracterizan la actividad (dimensiones).
  - la información relevante sobre el hecho (actividad) se representa por un conjunto de indicadores (medidas o atributos de hecho).
  - la información descriptiva de cada dimensión se representa por un conjunto de atributos (atributos de dimensión).

Esquema de hechos y dimensiones - lógico

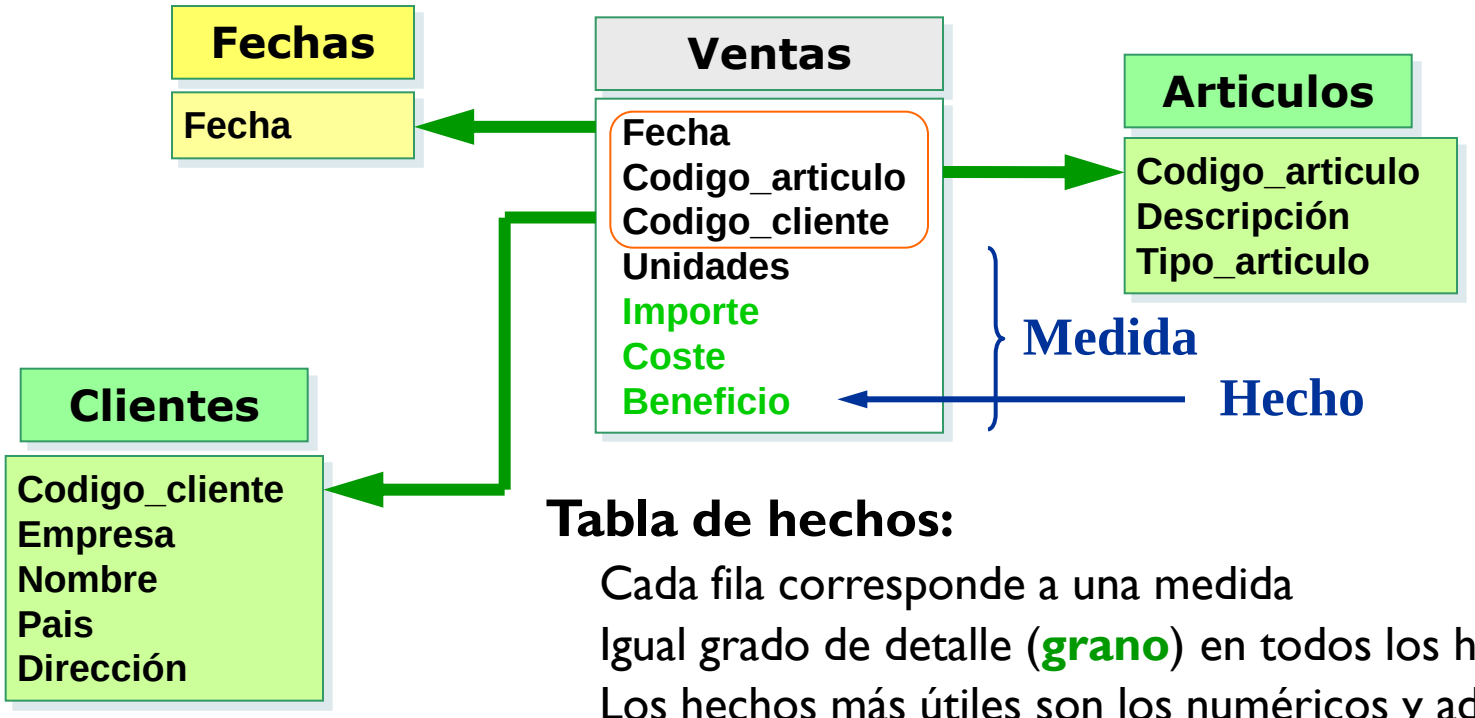


Tabla de hechos:

Cada fila corresponde a una medida

Igual grado de detalle (**grano**) en todos los hechos

Los hechos más útiles son los numéricos y aditivos

Tablas de dimensión:

Contienen descriptores textuales

Son los puntos de entrada en la tabla de hechos

El Modelo de datos dimensional - físico

Tabla de hechos:

Atributos (hechos) [aditividad]

Claves de referencia

Clave simple (autonumérica)

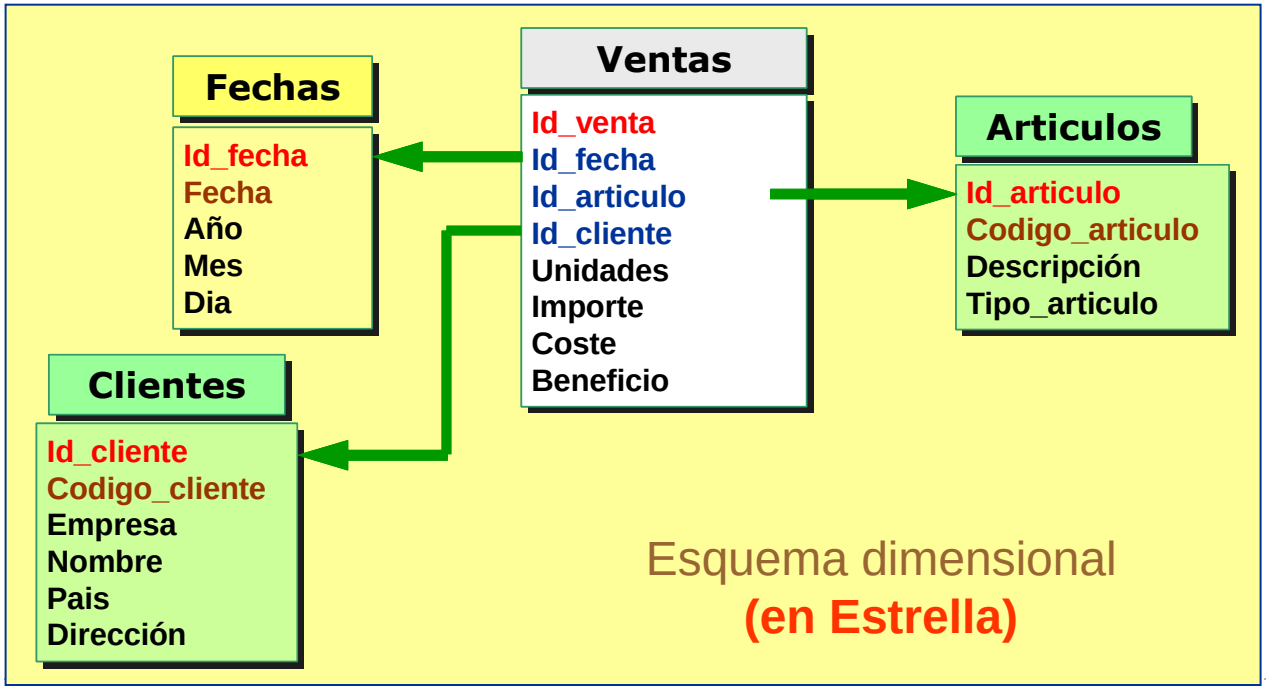
Tablas de dimensión:

Claves de gestión (sincronización operacional)

Atributos [criterios de agregación]

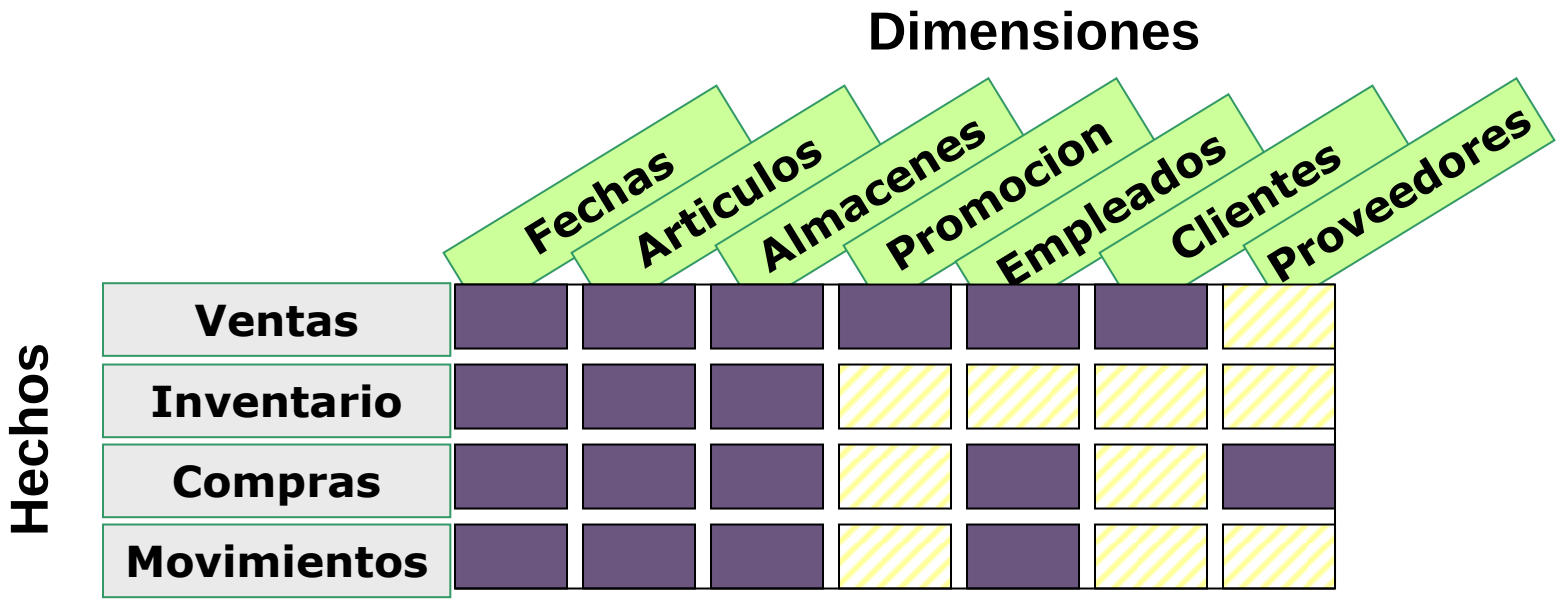
Claves simples (autonuméricas)

Claves autonuméricas ➔ Independencia ante cambios de claves de producción



BD dimensional

- Una BD dimensional está compuesta por varias estrellas en las que las dimensiones y hechos pueden ser compartidos
- A cada estrella se conoce como Data Mart

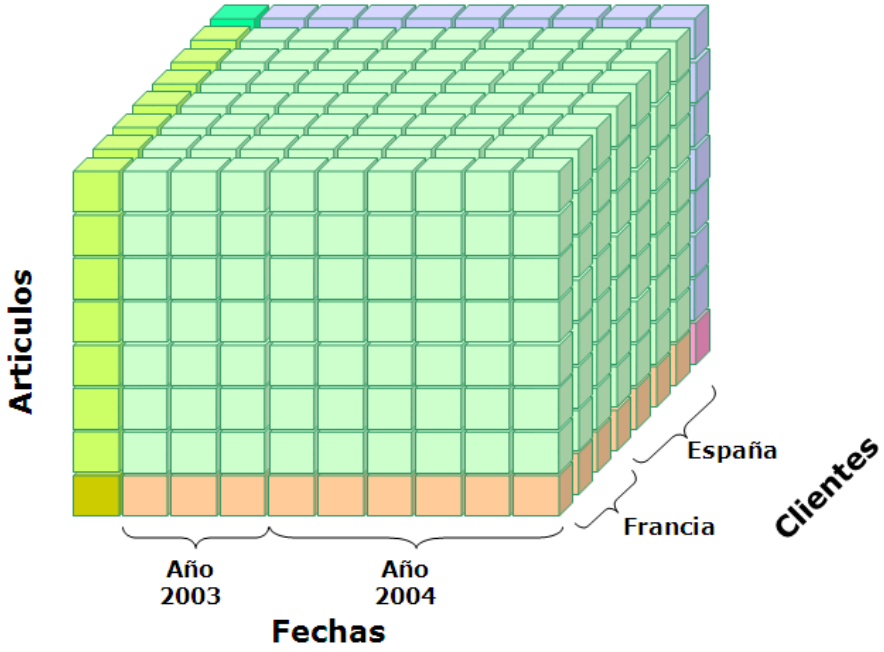


El modelo de datos relacional frente al dimensional

	Modelo	
	Relacional	**Dimensional**
Objetivos	Actualización de los datos	Consultas estratégicas
Datos	Actualizados dinámicamente	Históricos estáticos
Usuario	Sólo consultas elementales	No puede actualizar datos
Consistencia	Se persigue	Se da por supuesta
Redundancias	Se impiden	Se permiten
Consultas estr.	Costosas y presentación difícil	Sencillas y presentación fácil
Diseño	Alejado del usuario final	Próximo al usuario final

Cubos OLAP

- Estructura de almacenamiento que permite realizar diferentes combinaciones de datos para **visualizar** los resultados de una organización (indicadores) hasta un determinado grado de detalle, permitiendo **navegar** por sus dimensiones y **analizar** sus datos desde distintos puntos de vista.



Almacenamiento de los cubos

Opciones de almacenamiento

Rendimiento → MOLAP

Base Datos Multidimensional

Los datos que subyacen en los hipercubos son almacenados junto con las agregaciones en una estructura multidimensional

Capacidad → ROLAP

Base Datos Relacional

Los datos que subyacen en los hipercubos son almacenados junto con las agregaciones en una estructura relacional

HOLAP

Sistema híbrido

Los datos que subyacen en los hipercubos son almacenados en una estructura relacional y las agregaciones en una estructura multidimensional

DOLAP

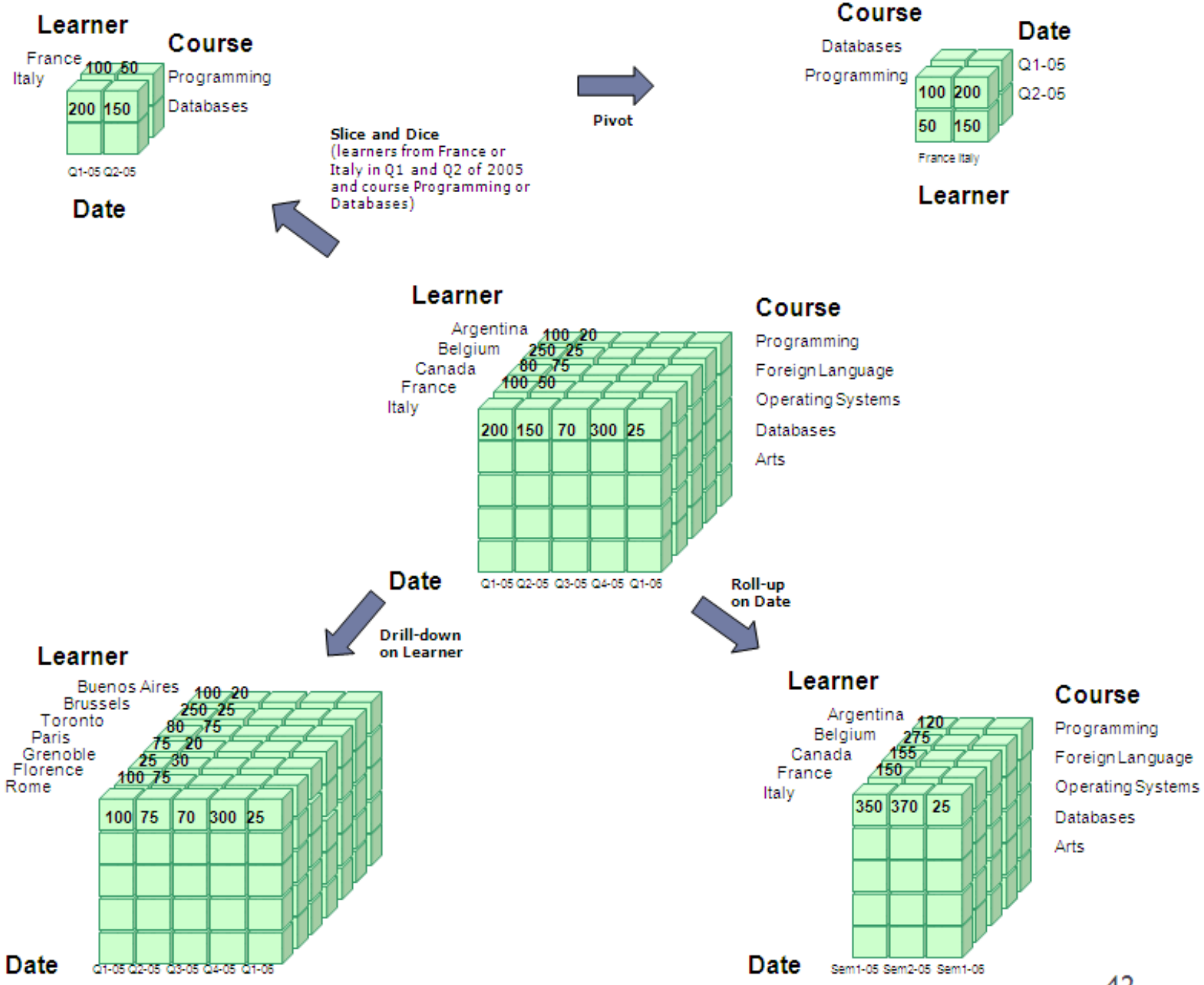
Desktop OLAP

Instalación MOLAP en un equipo cliente

Herramientas OLAP

- Lo interesante no es poder realizar consultas que, en cierto modo, se pueden hacer con selecciones, proyecciones, concatenaciones y agrupamientos tradicionales.
- Lo bueno de las herramientas OLAP son sus operadores de refinamiento o manipulación de consultas.

Roll up (drill-up): resumir los datos
Subir en la jerarquía o reducir las dimensiones
Drill down (roll down): el contrario del anterior
bajar en la jerarquía o introducir nuevas dimensiones
Slice and dice:
Selección y proyección
Pivot (rotar):
Reorientar el cubo
Drill:
Se utilizan las coordenadas dimensionales especificadas por un usuario para una celda en un cubo para moverse a otro cubo a ver información relacionada


Learner
Course
France
Italy
100 50
200 150
Programming
Databases
Q1-05 Q2-05
Date
Slice and Dice
(learners from France or Italy in Q1 and Q2 of 2005 and course Programming or Databases)
Pivot
Course
Date
Databases
Programming
Q1-05
Q2-05
100 200
50 150
France Italy
Learner
Learner
Argentina
Belgium
Canada
France
Italy
100 20
250 25
80 75
100 50
200 150 70 300 25
Programming
Foreign Language
Operating Systems
Databases
Arts
Q1-05 Q2-05 Q3-05 Q4-05 Q1-06
Date
Drill-down on Learner
Roll-up on Date
Learner
Buenos Aires
Brussels
Toronto
Paris
Grenoble
Florence
Rome
100 20
250 25
80 75
75 20
25 30
100 75
100 75 70 300 25
Q1-05 Q2-05 Q3-05 Q4-05 Q1-06
Date
Learner
Argentina
Belgium
Canada
France
Italy
120
275
155
150
350 370 25
Course
Programming
Foreign Language
Operating Systems
Databases
Arts
Sem1-05 Sem2-05 Sem1-06
Date

Herramientas OLAP

Las herramientas de OLAP se caracterizan por:

- ✓ ofrecer una visión multidimensional de los datos (matricial).
- ✓ no imponer restricciones sobre el número de dimensiones.
- ✓ ofrecer simetría para las dimensiones.
- ✓ permitir definir de forma flexible (sin limitaciones) sobre las dimensiones: restricciones, agregaciones y jerarquías entre ellas.
- ✓ ofrecer operadores intuitivos de manipulación: *drill-down, roll-up, slice-and-dice, pivot*.
- ✓ ser transparentes al tipo de tecnología que soporta el almacén de datos (ROLAP o MOLAP).