

Sistemas de Información II

Tema 1. El enfoque de bases de datos

Bibliografía:

Elmasri y Navathe: “Fundamentos de Sistemas de Bases de Datos”
3ª edición, 2002 (Capítulo 1).

Dr. Carlos Castillo

UPF – 2005

Temas del curso

- ♦ Conceptos básicos
- ♦ Sistemas gestores de bases de datos
- ♦ Modelo entidad-relación
- ♦ Modelo relacional
- ♦ Álgebra relacional
- ♦ Representación física de los datos
- ♦ Estructuras en memoria secundaria
- ♦ Datos semiestructurados

Objetivos del curso

Entender los principios detrás del diseño de los sistemas gestores de bases de datos.

Aprender a diseñar y manejar datos en una base de datos.

Temas de esta clase

- ♦ Aplicaciones de bases de datos
- ♦ Minimundo
- ♦ Características del enfoque
- ♦ Roles y actores
- ♦ Ventajas de las bases de datos
- ♦ Implicaciones
- ♦ SQL
- ♦ Resumen

Definición: Base de datos

Una colección de datos relacionados, y una descripción de estos datos, diseñados para cumplir con las necesidades de información de una organización.

(Connolly & Begg)

Definición: Sistema gestor de bases de datos

Un sistema de software que permite a sus usuarios crear, mantener y controlar el acceso a una base de datos.

(Connolly & Begg)

Informix

ORACLE

PostgreSQL

MySQL

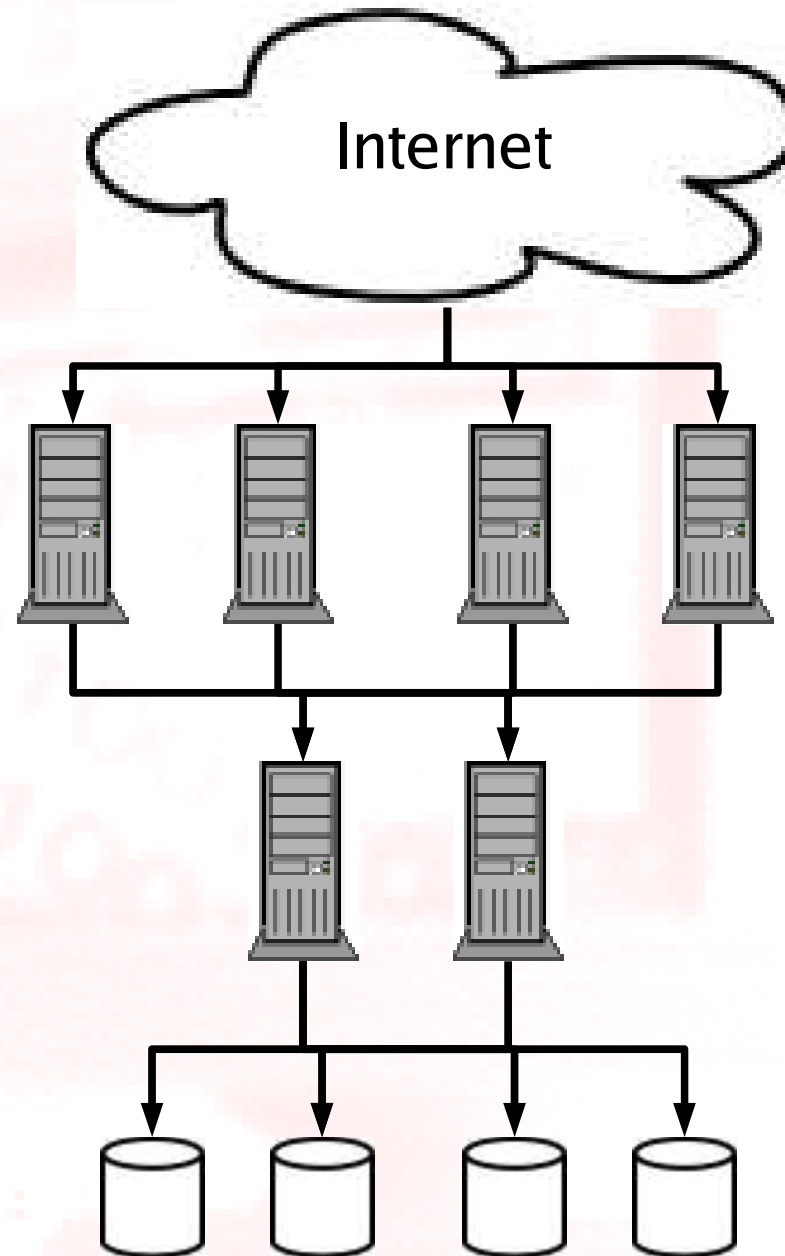
¿Qué es una base de datos?

- Es una colección de datos relacionados.
- Generalmente es una parte de un sistema más grande.

Front-end
Aplicación

Back-end
Bases de datos

Almacenamiento



Aplicaciones tradicionales

- ♦ Inventario
- ♦ Reservas
- ♦ Compras
- ♦ Seguimiento
- ♦ Logística
- ♦ Transporte
- ♦ Educación
- ♦ Estadísticas
- ♦ Bancos
- ♦ Adm. Clientes
- ♦ Bibliotecas
- ♦ Campus global
- ♦ Etc.

¿Qué tipos de datos?

- ♦ Cualquier organización usualmente
 - ♦ Intercambia
 - ♦ Procesa
 - ♦ Almacena
 - ♦ Recupera
 - ♦ Actualiza
 - ♦ Revisa
- ♦ ... datos provenientes de su actividad

¿Qué tipos de datos?

- ♦ Ciertos aspectos del mundo real
 - ♦ Cuantificables
 - ♦ Objetivables
 - ♦ Útiles
- ♦ Vista parcial (“minimundo”)

Ejemplo

- ♦ Entidades
 - ♦ Alumnos
 - ♦ Curso
 - ♦ Secciones
 - ♦ Notas
 - ♦ Requisitos
- ♦ ¿Qué atributos podría tener cada uno?

Ejemplo (cont.)

- ♦ Atributos por entidad
 - ♦ Alumnos: nombre, código, año, especialidad
 - ♦ Curso: nombre, código, créditos, depto
 - ♦ Secciones: número, curso, semestre, año
 - ♦ Notas: alumno, sección, nota
 - ♦ Requisitos: curso, requisito
- ♦ Pueden ser variables

Expectativas (ej.: Banco)

- ♦ Ítems de datos
 - ♦ clientes, operaciones, empresas, etc.
- ♦ Registro de cada transacción
- ♦ Transacciones secretas
- ♦ Usuarios concurrentes
- ♦ No pueden haber errores
- ♦ Tolerancia a fallas externas

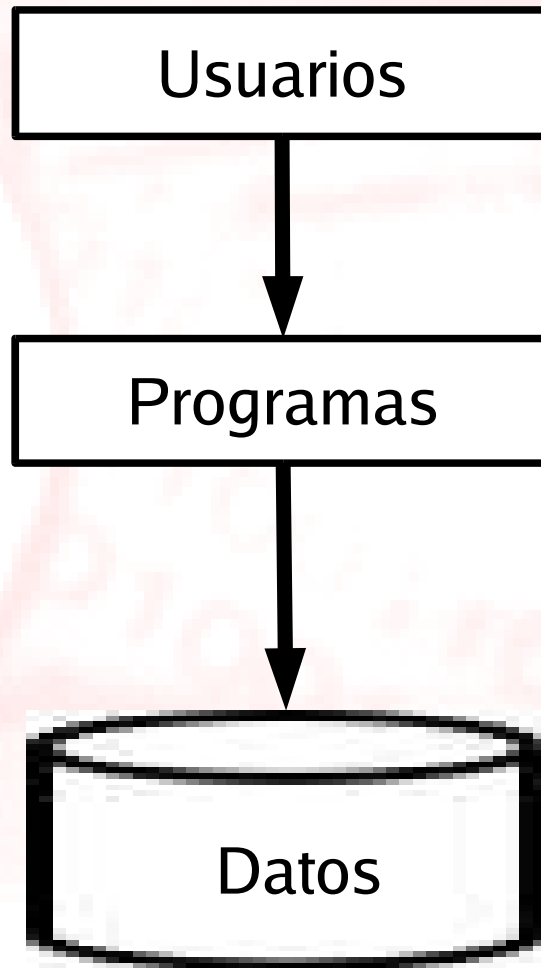
Expectativas ... (cont.)

- ♦ Reportes de todos los aspectos del sistema
- ♦ Nuevos productos del banco se venden
 - ♦ Hipotecas con tasa de crédito que varía dependiendo de ciertos factores
 - ♦ Premio si un cliente invita a otro cliente que resulta ser un buen cliente
 - ♦ Promociones en casos especiales
 - ♦ etc.

¿Cómo se cumple con estas expectativas?

- ♦ Siguiendo una metodología
- ♦ Modelando los datos en términos abstractos
 - ♦ Ciertos tipos de dato prefijado
 - ♦ Cierta estructura formal (tablas, columnas, filas)
- ♦ Utilizando un gestor de bases de datos
 - ♦ Transacciones, concurrencia, etc.

Enfoque típico



Usando ficheros

- ♦ Fichero = Tabla de datos

Personas.txt

Nombre,Apellido,Año_Nacimiento

Juan,Perez,1957

John,Smith,1980

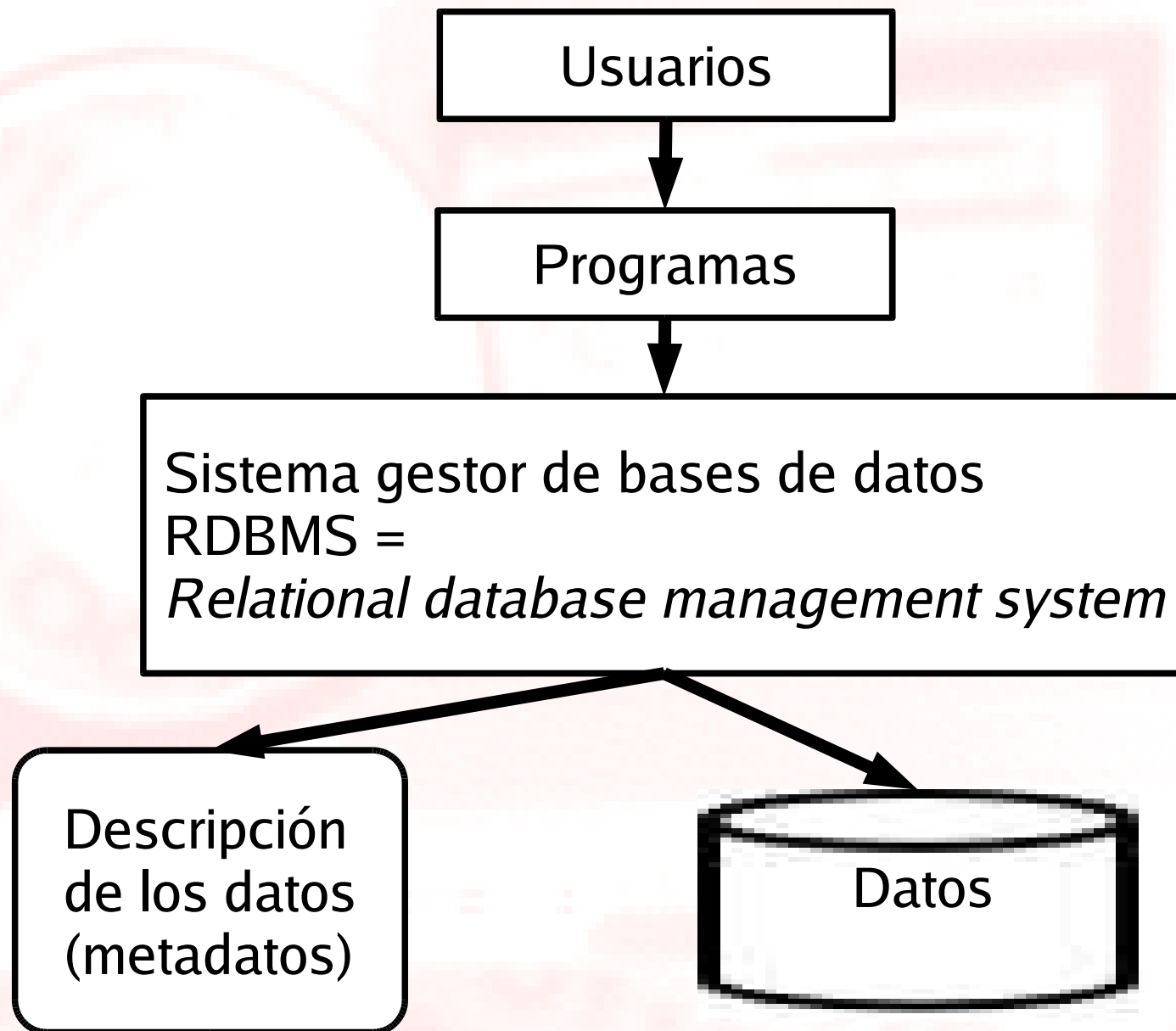
...

- ♦ Tipos de reporte: todas las personas con un año de nacimiento, etc.
- ♦ Nuevo reporte = nuevo programa

¿Por qué no usar ficheros?

- ♦ No se puede ...
 - ♦ Buscar rápido
 - ♦ Modificar fragmentos pequeños
 - ♦ Hacer consultas complejas
 - ♦ Modificar varios ficheros a la vez
- ♦ No es eficiente consultar partes pequeñas de los ficheros sin leerlos completos en memoria
- ♦ No hay restricciones de integridad
- ♦ **Falta de flexibilidad**

Enfoque de bases de datos



Características del enfoque

- ♦ Datos autodescriptivos
 - ♦ Uso de un catálogo de metadatos
- ♦ Independencia de los datos
 - ♦ Otros programas para los mismos datos
- ♦ Múltiples vistas de los datos
- ♦ Compartir datos
- ♦ Transacciones multiusuario

Desarrollar programas en menos tiempo y que queden mejor

Términos clave

- ♦ Esquema de la base de datos
 - ♦ Descripción de la estructura
- ♦ Estado de la base de datos
 - ♦ Descripción del estado actual

Ejemplo

- ♦ Base de datos de autores
- ♦ Tablas
 - ♦ autores
 - ♦ editoriales
 - ♦ libros
- ♦ Tipos de dato
 - ♦ TEXT - Texto
 - ♦ INTEGER – Números enteros
 - ♦ DATE – Fecha

Ejemplos de tablas

autores

idautor INTEGER
nombre TEXT
apellido TEXT
telefono INTEGER
direccion TEXT

libros

idlibro INTEGER
idautor INTEGER
ideditorial INTEGER
nombre TEXT
paginas INTEGER
publicado DATE

editoriales

ideditorial INTEGER
nombre TEXT
direccion TEXT

Observar:

- Nombres de tabla
- Nombres de columna
- Tipos de dato
- Identificadores
- Relaciones
- Restricciones de integridad

Roles y actores

- ♦ Administrador de datos y bases de datos
- ♦ Diseñadores de bases de datos
- ♦ Desarrolladores de aplicaciones
- ♦ Usuarios finales

Administrador base de datos

- ♦ Administración de datos
 - ♦ Planificación de la base de datos
 - ♦ Desarrollo de estándares (ej.: nombres), políticas y procedimientos
 - ♦ Diseño lógico de la base de datos
- ♦ Administración de bases de datos
 - ♦ Diseño e implementación de la base de datos
 - ♦ Acceso y control (usuarios y passwords)
 - ♦ Mantenimiento (ej.: mantenimiento para mejor performance)
 - ♦ Respaldo

Diseñador de base de datos

- ♦ Diseño lógico de la base de datos
- ♦ ¿Qué datos deben ser almacenados?
 - ♦ Definir vista parcial
- ♦ ¿Cómo deben organizarse los datos?
- ♦ ¿Cuáles son los usos típicos de los datos?

Programador de aplicaciones

- ♦ Programas que usen las bases de datos
- ♦ Interfaces a otros sistemas
 - ♦ Web
 - ♦ E-Mail
 - ♦ Sistemas de ficheros
 - ♦ etc.

Usuarios finales

- ♦ Paramétricos o simples
- ♦ Expertos

Ventajas de las bases de datos

- ♦ Control de la redundancia
 - ♦ Menos uso de espacio
- ♦ Restricción de accesos
- ♦ Almacenamiento persistente de objetos
- ♦ Múltiples interfaces
- ♦ Relaciones complejas entre datos
- ♦ Restricciones de integridad
- ♦ Copias de seguridad y recuperación

¿Cuándo no usar BD relacional?

- ♦ Problemas muy simples
- ♦ Las bases de datos agregan un costo
 - ♦ Tamaño y complejidad del software
 - ♦ Costo de la base de datos, hardware y entrenamiento
- ♦ BD incluye un sobrecosto en tiempo de ejecución
 - ♦ Sistemas en tiempo real

Aplicaciones emergentes

- ♦ Sistemas de información geográfica
- ♦ Bases de datos multimedia
 - ♦ Video
 - ♦ Música
 - ♦ Sonido
 - ♦ Fotografías
- ♦ Sistemas de proceso analítico on-line
 - ♦ OLAP
- ♦ Bases de datos distribuídas

Resumen

- ♦ Bases de datos + Software
= Sistema de Bases de datos
- ♦ Abstracción de los datos
- ♦ Actores: administradores, usuarios finales, diseñadores, programadores.