

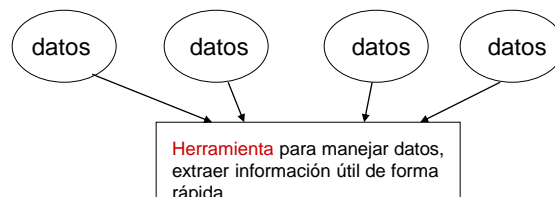
Sistemas de Base de Datos

Introducción

Profesora: Mónica Caniupán
Base de Datos - UBB
2024

Contexto

- ❑ Hoy en día existe una gran cantidad de información disponible (web, sistemas distribuidos, peer-to-peer systems, redes sociales, etc.)
- ❑ Los datos tienen mayor valor cuando se encuentran **organizados**



Base de Datos (BD)

- ❑ Una base de datos es una **colección** de datos relacionados
- ❑ Con **datos** nos referimos a hechos conocidos que se pueden registrar y que tienen un significado implícito, como por ejemplo:
 - ❑ Nombres,
 - ❑ Números de teléfonos,
 - ❑ Direcciones, etc.

Propiedades de las BDs

- ❑ Una BD es una colección de datos **lógicamente coherente** con algún tipo de significado
- ❑ Representa algún **aspecto del mundo real**
- ❑ Se diseña, construye y en ella se almacenan datos con un propósito específico

Ejemplo de BD

- Una BD de una universidad podría contener información sobre:
 - **Entidades:** estudiantes, profesores, cursos, salas, etc.
 - **Relaciones entre las entidades:** estudiantes por curso, profesores dictando cursos, uso de salas por cursos, etc.
- Una BD se implementa en un Sistema de Gestión de BDs (SGBDs)

Sistema de Gestión de BDs

- Un **SGBD** es un software diseñado para almacenar y administrar grandes volúmenes de datos
- Una alternativa al uso de **SGBDs** es almacenar los datos en **archivos** y escribir aplicaciones de software para manejar estos datos

Sistemas de Archivos vs SGBD

- ❑ **Ejemplo:** Una empresa que maneja información de **empleados, departamentos, productos, ventas, etc.**
- ❑ La colección de datos es de 500 GB
- ❑ Existe acceso concurrente a los datos
- ❑ Tareas claves:
 - Responder a consultas de manera rápida
 - Propagar cambios en datos
 - Mantener la seguridad de los datos

Sistemas de Archivos vs SGBD

- ❑ Probablemente no tenemos 500 GB de memoria principal para almacenar los datos
 - Se necesita disco y los datos se deben almacenar en memoria principal cada vez que se necesitan
- ❑ ¿Cómo respondemos las consultas de los usuarios?
 - Programas especiales para responder **cada consulta**

Sistemas de Archivos vs SGBD

- ❑ Si el sistema falla se debe asegurar que los datos retornan a un estado consistente
- ❑ Debemos “**proteger**” los datos de inconsistencias generadas por el acceso concurrente
- ❑ La seguridad de datos vía “claves” de archivos no es suficiente
 - Necesitamos proteger subconjuntos de los datos

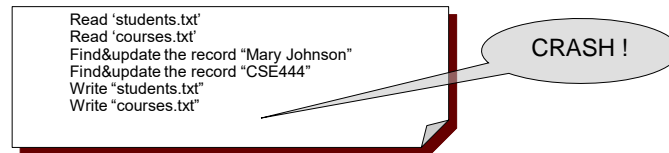
Problemas con los Sistemas de Archivos

- ❑ Queremos agregar el registro “Mary Johnson” al curso “CSE444”
 - Necesitamos escribir una aplicación para ello
- ❑ Programa C:

```
Read 'students.txt'
Read 'courses.txt'
Find&update the record "Mary Johnson"
Find&update the record "CSE444"
Write "students.txt"
Write "courses.txt"
```

Problemas con los Sistemas de Archivos

- ❑ Supongamos que existe una falla mientras se ejecuta el programa



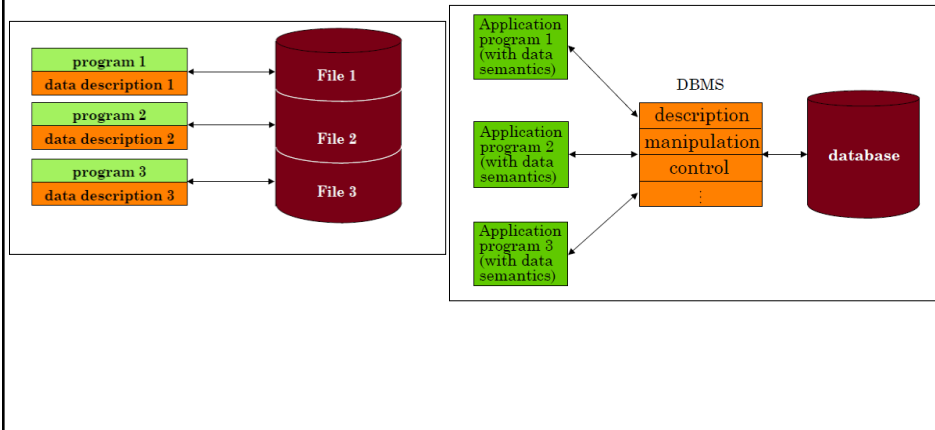
- ❑ ¿Qué problema se genera?
- ❑ ¿Si hay acceso simultáneo al archivo?

Problemas con los Sistemas de Archivos

- ❑ Redundancia de datos
- ❑ Poca capacidad para almacenar datos
- ❑ Dificultad para combinar datos (datos están en archivos separados)
- ❑ Anomalías de actualización e inconsistencias
- ❑ Alto costo para propagar actualizaciones de datos
- ❑ Requiere conocimiento sobre detalles de almacenamiento
- ❑ Falta de lenguaje de consulta estándar
- ❑ Poca seguridad, etc.

Sistemas de Archivos vs SGBD

- Diversas aplicaciones generan datos



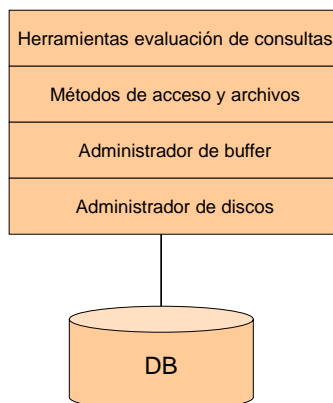
Ventajas de los SGBDs

- Un SGBD permite almacenar **grandes volúmenes** de datos
- Los datos **no se almacenan** como una colección de archivos
- El SGBD ofrece una vista abstracta de los datos y oculta los detalles (**independencia de los datos**)
- El SGBD emplea técnicas sofisticadas para almacenar y recuperar datos de manera eficiente

Ventajas de los SGBDs

- ❑ **Integridad y seguridad de los datos:** el SGBD permite que se cumplan las restricciones de integridad y otras restricciones necesarias
- ❑ **Administración de los datos:** existe una administración centralizada de los datos independiente de la cantidad de usuarios
- ❑ **Acceso concurrente y recuperación:** el SGBD programa accesos concurrentes a los datos

Estructura Básica de un SGBD



Modelos de Datos

- ❑ Un **modelo de datos** es una colección de estructuras descriptivas de datos de alto nivel que oculta muchos detalles de almacenamiento de bajo nivel
- ❑ Un SGBD permite definir los datos que se van a almacenar en términos de un modelo de datos
- ❑ La mayor parte de los SGBDs se basan en el **modelo de datos relacional**

Modelo de Datos Relacional

- ❑ La estructura central de éste modelo es la **relación** (conjunto de registros)
- ❑ La descripción de los datos en términos de un modelo de datos se denomina **esquema**

Modelo de Datos Relacional

- El esquema para una relación específica:
 - Un nombre
 - Un nombre para cada **atributo** (campo o columna)
 - El **tipo de datos** para cada atributo
- **Ejemplo:**
ALUMNOS(**ID**: integer,
NOMBRE: varchar(20),
EDAD: integer,
DIRECCION: varchar(20))

Modelo de Datos Relacional

- Una instancia para la relación **ALUMNOS**:

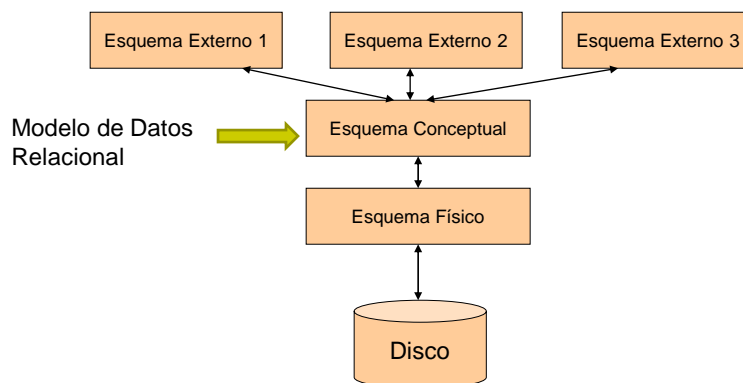
ID	NOMBRE	EDAD	DIRECCION
11	Pedro	20	Rengo 1, Concepción
12	María	22	San Martín 42, Concepción
13	Antonio	21	Prat 33, Concepción

- Cada fila de la relación **ALUMNOS** es un registro que describe a un alumno
- Cada fila sigue el esquema de la relación **ALUMNOS**

Modelo de Datos Relacional

- La descripción se puede hacer mas precisa agregando **restricciones de integridad**
- Es decir, condiciones que deben satisfacer los registros de una relación
 - **Ejemplo:** se podría pedir que cada alumno tenga un valor único para el atributo **ID**

Niveles de Abstracción en los SGBDs



Niveles de Abstracción en los SGBDs

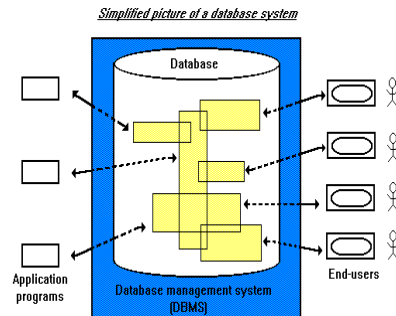
- ❑ Esquema externo:
 - Representa como el usuario ve los datos
 - Suelen existir muchas **vistas** externas
- ❑ Único esquema conceptual
 - Define la estructura lógica de los datos en términos del **modelo de datos**
- ❑ Único esquema físico
 - Especifica detalles adicionales de almacenamiento

Ejemplo

- ❑ Esquema Conceptual
 - **ALUMNOS**(ID: integer, NOMBRE: varchar(20), EDAD: integer, DIRECCION: varchar(20))
 - **CURSOS**(IDC: integer, NOMBRE: varchar(20))
 - **INSCRIBE**(ID:integer, IDC:integer, NOTA:integer)
- ❑ Esquema Físico
 - Relaciones almacenadas como archivos sin ningún orden
 - Índice sobre atributo **ID** de ALUMNOS
- ❑ Esquema Externo
 - Vista: **ALUMNOS_CURSO**(IDC: integer, NUM_ALUM: integer)

Independencia de Datos

- ❑ Al usar un SGBD los programas de aplicación quedan aislados de las modificaciones respecto a la estructura y almacenamiento de los datos
- ❑ **Independencia Lógica**: protección de cambios en la estructura lógica (conceptual) de los datos
- ❑ **Independencia Física**: protección de cambios en la estructura física de los datos



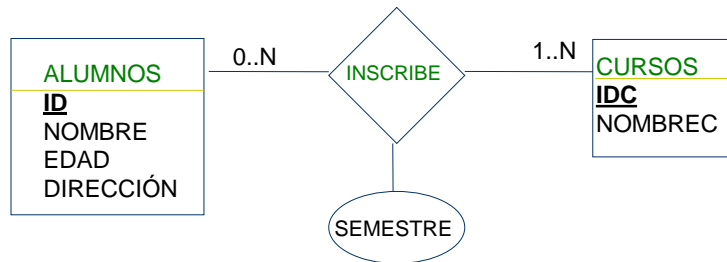
Modelos Semánticos de Datos

- ❑ Los **modelos semánticos de datos** son modelos de **alto** nivel, más abstractos, que permiten una buena descripción inicial de los datos
- ❑ Sirven como punto de partida para el diseño de una BD
- ❑ Son ideales para interactuar con los usuarios de los sistemas
- ❑ El modelo de datos semántico más utilizado es el **Modelo Entidad Relación** (MER)

Modelo Entidad Relación

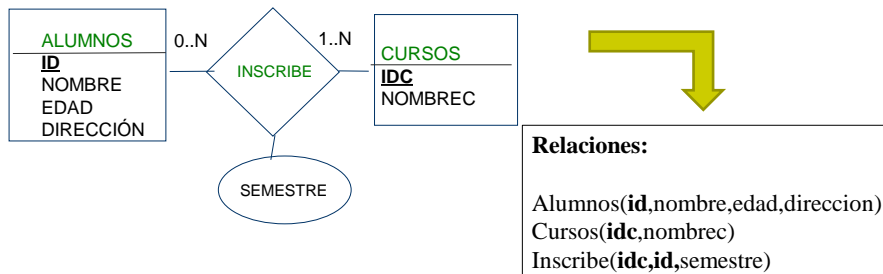
□ Elementos del MER

- Entidades, atributos y relaciones, restricciones de integridad, etc.
- Del MER podemos construir un MR



Modelo Entidad Relación

■ Del MER al MR:



Lenguaje de Datos

- **Lenguaje de Definición de Datos (LDD):** se usa para definir los esquemas externos y conceptual
- **Lenguajes de Manipulación de Datos (LMD):** se utilizan para consultar y modificar los datos
 - Álgebra relacional,
 - Cálculo relacional,
 - SQL

Ejemplo: Consultas a una BD

- ¿Cuál es el nombre del estudiante con **ID** 12?
- ¿Cuántos profesores enseñan **Cálculo I** este semestre?
- ¿Cuántos estudiantes tienen nota mayor a 4 en Cálculo I?
- ¿Existen estudiantes de 18 años en Cálculo I?

Perspectiva Histórica

- ❑ En los 60's habían dos modelos de datos:
 - Modelo de datos de red
 - Modelo de datos jerárquico
- ❑ Modelos complejos y poco flexibles
- ❑ Acceso a datos es través de operaciones de bajo nivel
- ❑ Cambios en la estructura de los datos afecta a las aplicaciones que los usan
- ❑ Los programadores deben familiarizarse con la estructura de la base de datos

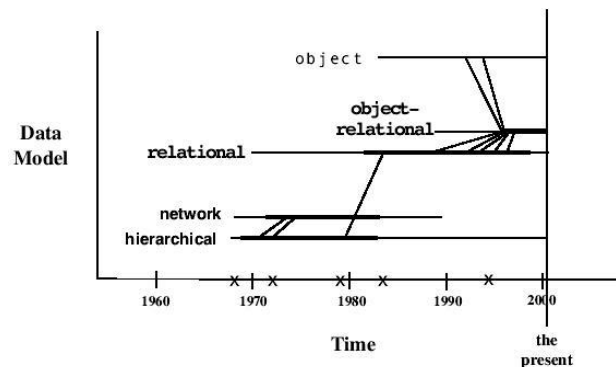
Perspectiva Histórica

- ❑ En 1970, Edgar Codd (IBM) propuso el **Modelo de Datos Relacional**
- ❑ En los 80's se consolida el **modelo de datos relacional** como paradigma dominante de los SGBDs
- ❑ En los 80's se estandarizó el **SQL** (Structured Query Language), desarrollado por la IBM
- ❑ SQL:1999 fue adoptado por el **ANSI** (American National Standards Institute) y la **ISO** (International Organization for Standardization)

Perspectiva Histórica

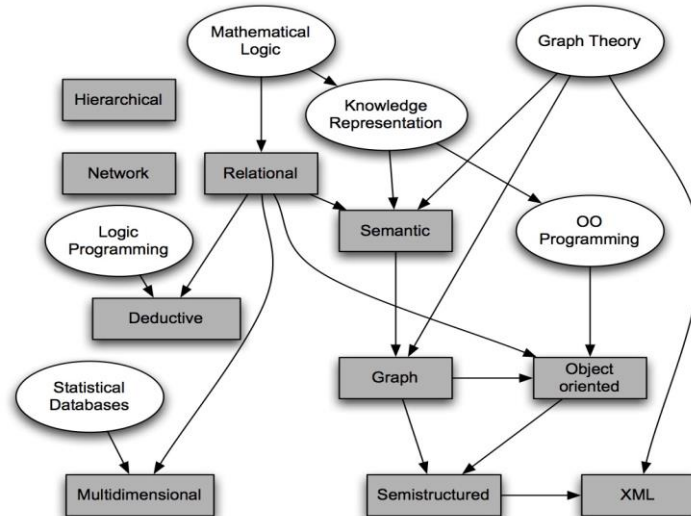
- En los 90's se extendieron los lenguajes de consulta y se generaron otros modelos de datos
 - Oracle, IBM's DB2, Informix UDS de Informix ampliaron sus sistemas para almacenar nuevos tipos de datos (imágenes y textos)
 - Modelos de datos orientado a objetos y modelo de datos relacional-objeto
- Nacen los Data Warehouses
- Actualmente se investigan modelos de datos para la WEB

Perspectiva Histórica



Perspectiva Histórica

1970's



2000's