

# TEMA 1. Introducción a las Bases de Datos

## » Sistemas de archivos «

---

### » Sistemas de archivos de sistemas orientados al proceso.

La parte relevante es el proceso de tratamiento de los datos en vez de los propios datos, creándose en formatos propios dependiente de la aplicación (*Archivos de texto, archivos de gráficos, hojas de cálculo...*)

#### » Problemas

- ❖ **Redundancia e inconsistencia de los datos**
  - **Redundancia** → cuando un dato se encuentra almacenado en lugares diferentes
  - **Inconsistencia** → cuando éstos datos además presentan valores diferentes.
- ❖ **Espacio de almacenamiento elevado**
  - La redundancia de datos provoca un aumento del espacio de almacenamiento, ya que el mismo dato se encuentra almacenado en más de un lugar al mismo tiempo.
- ❖ **Dependencia de la estructura del fichero**
  - **Dificultad para el acceso a los datos** → Se necesita conocer la estructura de los tipos de ficheros que se utilicen creando el software necesario para acceder.
  - **Dificultad para modificar la estructura de un fichero** → se tendrían que actualizar todos los programas que acceden a estos tipos de ficheros para que puedan reconocer los cambios producidos
- ❖ **Mantenimiento de la integridad**
  - Se tienen que crear restricciones de integridad para todos los programas que usen este dato, consumiendo tiempo y recursos.
    - *Ej. Los meses del año: se debe garantizar que sea un número entre 1 y 12*
- ❖ **Acceso concurrente**
  - Se tiene que garantizar que no se produzcan problemas de consistencia en los datos ante accesos simultáneos por distintos usuarios a los mismos en todos los programas.

## » Bases de datos «

---

### » Surgen como intento de solucionar los problemas de los sistemas de archivos clásicos.

En ellas los datos se encuentran organizados y relacionados en un depósito común centralizado. Son sistemas orientados a los datos, teniendo los datos la importancia, y no el tratamiento sobre ellos.

#### » Ventajas

- ❖ **Datos interrelacionados y estructurados** en una estructura única y centralizada
- ❖ **Independencia** en los datos y sus tratamientos
- ❖ **Se evitan las redundancias**, no controladas, e **inconsistencias de los datos**
- ❖ Permite **acceso de varios usuarios o aplicaciones de forma simultánea**
- ❖ **Garantiza la disponibilidad y la integridad de los datos**
- ❖ Permite **establecer mecanismos de seguridad** (usuario, contraseña y privilegios) que permiten **determinar las acciones que un usuario puede realizar en la base de datos**

#### » Inconvenientes → *Provocan que el coste de implantación y mantenimiento de una base de datos sea mayor*

- ❖ **Necesidad de personal cualificado** ya que la complejidad de las bases de datos es mayor
- ❖ Recursos de hardware **más potentes**
- ❖ Al estar todos los datos en un único depósito central **pueden producirse problemas de disponibilidad** ante fallos de la base de datos → **Las aplicaciones que hacen uso de la base de datos no pueden acceder a los datos**

## » Sistema Gestor de Base de Datos (SGBD) «

---

» Conjunto coordinado de programas y lenguajes que permite que los usuarios puedan trabajar con la base de datos.

- ❖ **Definición** → define la **estructura** de la base de datos **junto con las relaciones** que existen
  - Lenguaje de Definición de datos o **DDL**
- ❖ **Manipulación** → realiza operaciones de **consulta, inserción, actualización y eliminación** de datos
  - Lenguaje de Manipulación de Datos o **DML**
- ❖ **Control** → ofrece mecanismos para el **mantenimiento de la base de datos**: gestión de usuarios y permisos, copias de seguridad, control de concurrencia, diccionario de datos...
  - Lenguaje de Control de Datos o **DCL**

## » Modelo de datos «

---

» Conjunto de herramientas conceptuales que permite describir los diferentes elementos de un problema que se quiere modelar

## » Nivel de abstracción en una base de datos «

---

» Arquitectura definida en tres niveles de abstracción para diseño de base de datos

- ❖ **Nivel externo** → representa cómo los usuarios perciben la base de datos los usuarios.
- ❖ **Nivel conceptual** → describe de forma única la estructura de la base de datos: **datos y relaciones entre ellos**. Independiente a los usuarios.
- ❖ **Nivel interno o físico** → especifica cómo se almacenan los datos de la base de datos en el disco duro.

» Independencias

- ❖ **Física** → Capacidad de poder modificar el esquema interno sin tener que modificar el esquema externo. Los cambios son a nivel interno de cómo se almacenan los datos en disco sin tener que modificar los programas que acceden a los datos.
- ❖ **Lógica** → Capacidad de modificar el esquema conceptual sin tener que modificar el esquema externo. Se modifica la estructura de la base y/o las relaciones que tienen entre ellos sin modificar los programas que acceden a estos datos

» Las independencias lógicas son más difíciles de conseguir que las físicas «

## TEMA 2. Diseño conceptual de Bases de Datos

### » Entidad «

---

» Cualquier elemento que es relevante para el problema que se está modelando.

Una entidad puede ser un concepto, un objeto, una persona, un lugar, una cosa...

» No pueden existir dos entidades con el mismo nombre

» Dos tipos de entidades

- ❖ **Entidades fuertes**  
Existen por sí mismas, no dependen de otra entidad.

Paciente
----------

- ❖ **Entidades débiles**

Aquellas que dependen de otra entidad, la entidad fuerte, para su existencia.



## » Atributos «

### » Características que poseen las entidades y permiten diferenciar ocurrencias.

Una entidad puede ser un concepto, un objeto, una persona, un lugar, una cosa...

#### » Tipos de atributos

- ❖ **Simple**

No se descompone en otros atributos más pequeños con significado propio.



- ❖ **Compuesto**

Atributo que se puede descomponer en atributos más pequeños

➤ NombreC → Nombre, Apellido1, Apellido2

- ❖ **Multivaluado**

Puede tener más de un valor.



- ❖ **Derivado**

Se puede obtener a partir de otro u otros atributos.



## » Subclases y superclases «

### » Todas las entidades de las subclases pertenecen a la superclase.

Las subclases no tienen clave propia, sino que la heredan de su superclase.

#### » Herencia de atributos y relaciones

- ❖ **De atributos**

Además de sus propios atributos, poseen los de su propia clase.

- ❖ **De relaciones**

Además de sus propias relaciones, heredan las relaciones de su superclase

## » Especialización y generalización «

#### » Especialización y generalización

- ❖ **Especialización - Top-Down**

Se parte de la superclase y se crean subclases en función de los atributos o relaciones que poseen sólo un subconjunto de las entidades.

- ❖ **Generalización - Bottom-Up**

Se procede de forma inversa a la especialización, en las que generalizando se encuentran atributos y/o relaciones comunes que se unifican en una superclase.

#### » Restricciones

- ❖ **Participación** → Indica la obligatoriedad o no de que las entidades que forman la superclase participen en las subclases

- **Participación total**

**Todas** las entidades que forman la superclase tienen que participar en, al menos, una de las subclases. → Línea gruesa

➤ **Participación parcial**

Pueden existir entidades que **no participen** en las subclases. → Línea fina

❖ **Solapamiento y disyunción**

➤ **Solapadas**

Cada entidad de la superclase puede participar en **más de una** subclase.

➤ **Disjuntas**

Cada entidad de la superclase puede participar en **una única** subclase.

» **Combinaciones posibles**

❖ **Solapamiento**

➤ **Total** → Pueden participar **en más de una subclase**, teniendo que participar en al menos **una**.

➤ **Parcial** → Pueden participar **en más de una subclase**, existiendo entidades que **no participen** en ninguna.

❖ **Disyunción**

➤ **Total** → Pueden participar **en sólo una subclase**, teniendo que participar en al menos **una**.

➤ **Parcial** → Pueden participar **en sólo una subclase**, existiendo entidades que **no participen** en ninguna.