

Departamento de Lenguajes y Ciencias de la  
Computación Universidad de Málaga

## ***Bases de Datos***

*(Ingeniería Técnica en Informática de Sistemas)*

### **1. Sistemas de Bases de Datos: Introducción**



E.T.S.I. Informática

*J. Galindo Gómez*

#### ***Introducción: Los SGBD***

- **SGBD** (Sistema Gestor de Bases de Datos) o **DBMS** (*DataBase Management System*):
  - Un **SGBD** está formado por:
    - **Base de Datos (BD)**: Colección de datos sobre un ente particular (empresa, organización, tema...).
    - **Programas (Software)** para acceder y manipular esos datos.
    - Otros elementos que pueden considerarse: **Hardware y Usuarios**.
  - Los **SGBD** surgen, propiamente, a mediados de los años 60's, como un intento de mejorar el **Sistema de Procesamiento de Archivos** utilizado hasta entonces, de forma que una base de datos sea Integrada y Compartida:
    - **Integrada**: Unificación de archivos e información (evitando redundancias, mejorando los accesos...).
    - **Compartida**: Información utilizada por varios usuarios (total o parcialmente).

## Introducción: Los SGBD

- **Sistema de Procesamiento de Archivos:** Tiene una serie de *inconvenientes* que son reducidos en los SGBD:
  - **Redundancia** (datos duplicados) e **Inconsistencia** (datos contradictorios).
  - **Dificultad de Acceso a ciertos datos o información:** Si no existen programas para acceder o calcular cierta información, no puede accederse a ella. Ej.: Calcular totales, o registros con cierta condición...
  - **Aislamiento de Datos:** Los datos pueden estar en varios archivos con distintos formatos, que complican la creación de programas nuevos.
  - **Falta de Integridad:** Es complicado mantener ciertas condiciones en la información. Ej.: Que el saldo sea superior a cierta cantidad, que un empleado no esté adscrito a un número de Departamento que no exista...
  - **Problemas de Atomicidad en las operaciones:** A veces es esencial que para la consistencia de la BD se efectúen varias operaciones como si fueran una única operación, evitando que se produzcan fallos en medio de dicha operación. Ej.: En una transferencia bancaria hay que dar de quitar dinero de una cuenta y añadirlo a la otra.
  - **Problemas en el Acceso Concurrente:** Si varios usuarios acceden a la vez a un dato pueden producirse errores. Ej.: Si se saca dinero de una misma cuenta desde dos sitios distintos.
  - **Problemas de Seguridad:** Dificultad para controlar que ciertos usuarios no accedan a ciertos datos.

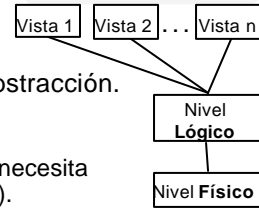
3

## Introducción: Los SGBD

- **Independencia de los Datos:** Puede ser Física o Lógica.
  - Es una de las grandes ventajas de los SGBD.
  - **Objetivo: Que las aplicaciones no dependan de cómo se almacenen los datos físicamente o de su organización lógica.**
    - Puede modificarse la estructura de los datos sin que afecte a los programas. Ej.: Modificar nombres y formato de los ficheros, añadir o borrar campos...
    - El **DBA (DataBase Administrator)** puede modificar esa estructura de los datos para mejorar el acceso y optimizar el uso de los recursos: Añadir o borrar índices, dividir en distintos archivos, usar nuevos discos u ordenadores (bases de datos distribuidas)...
    - El concepto y las ventajas son similares al uso de TDA en programación.
  - Esto implica que las aplicaciones (o los usuarios) tienen una **Visión Abstracta** de los datos: El SGBD esconde detalles sobre cómo se almacenan y mantienen los datos, de forma que facilita el acceso a estos datos, ya que sólo es necesario pedir al SGBD qué datos se requieren, sin especificar cómo conseguirlos.
    - Este es el principal objetivo de la **Arquitectura ANSI/SPARC** de 3 niveles.
      - La idea básica es poner un nivel intermedio entre las aplicaciones (o los usuarios) y el sistema de almacenamiento físico.

4

### Informe ANSI/SPARC (1978): 3 Niveles



- 1. **NIVEL EXTERNO (o de Vistas):** Mayor nivel de abstracción.
  - Es el nivel más cercano a los usuarios y a su forma de ver los datos (su punto de vista y sus necesidades).
  - Cada nivel externo describe sólo la parte de la BD que necesita para su aplicación o uso (de un programa, un usuario...).
    - Esto facilita el control de los permisos de acceso (seguridad).
  - Existen distintos tipos de usuarios: Programador en C (ve los datos como estructuras struct), usuario final de un programa...
- 2. **NIVEL CONCEPTUAL (o Lógico):** Conecta los otros dos niveles.
  - Describe qué datos se almacenan en la BD y qué relaciones existen entre ellos.
  - Los programadores de BD trabajan con este nivel, aunque también tienen su vista particular. Nunca un programa debe acceder directamente al nivel interno.
  - **Independencia Lógica de los Datos:** Cuando modificar este nivel no implica la modificación de los niveles externos (programas).
    - En muchas ocasiones la dependencia lógica de los programas es fuerte.
- 3. **NIVEL INTERNO (o Físico):** El más cercano al almacenamiento físico.
  - Describe cómo se almacenan realmente los datos (estructuras de datos complejas de bajo nivel: Registros, ficheros, bytes, datos, organización...
  - Los DBA pueden conocer y controlar ciertos aspectos de este nivel, pero no es necesario: Puede conseguirse optimizar el sistema y ganar en eficiencia (*tuning*).
  - **Independencia Física de los Datos:** Cuando modificar este nivel no implica la modificación de los niveles externos (programas).

5

### Clasificación de los SGBD

- Dependiendo del **Modelo de Datos** utilizado:
  - Modelo **Relacional**: Un conjunto de tablas.
  - Modelo **Dirigido a Objetos**: Un conjunto de objetos. Una clase es un tipo de objeto con las mismas características (o propiedades) y operaciones (o métodos). Las clases forman una jerarquía (grafo acíclico).
  - Modelo **Relacional dirigido a Objetos**.
  - Otros modelos antiguos: **Jerárquico** y en **Red** (CODASYL).
- Dependiendo del **Número de Usuarios** que soporta:
  - **Monousuario** y **Multiusuario**.
- Dependiendo del **Número de Sitios** en los que está la BD:
  - **SGBD Centralizado**.
  - **SGBD Distribuido** (y conectado mediante una red): Pueden ser **homogéneos** (con mismo software en todos los lugares) o **heterogéneos**. Esto genera SGBD **Federados** (*Multidatabase*), donde existen diversos SGBD autónomos con cierto acoplamiento (o comunicación) entre ellos.
- Otros: **ODBC** (*Open Database Connectivity*), estándar de comunicación con SGBD.
  - De **propósito general** o **específico**.
  - **Sistemas OLTP** (*On-Line Transaction Processing*): Sistemas con muchas transacciones concurrentes y necesitando rapidez (pocos retardos).

6

## Gestión de Bases de Datos

- **ESQUEMA de la BD:** Diseño completo de la BD a nivel conceptual (datos, relaciones entre ellos, restricciones básicas...).
  - Una vez definido el esquema de una BD no es usual cambiarlo.
  - El contenido de una BD (dentro de cierto esquema) es habitual modificarlo, puesto que el mundo cambia. Por eso se dice que un esquema puede tener varios **ejemplares**, **instancias** (*database instance*) o **estados** (*database state*), que tendrán valores distintos en sus datos.
- **Lenguajes de Gestión de BD:**
  - **DDL (Data Definition Language) o Lenguaje de Definición de Datos (LDD):** Se utiliza para definir el ESQUEMA de la BD.
    - Esa información se almacena en el **Diccionario de Datos** (metadatos).
  - **DML (Data Manipulation Language) o Lenguaje de Manipulación de Datos (LMD):** Se utiliza para efectuar operaciones sobre los datos:
    - **Recuperaciones o Consultas a la BD (retrievals, select).**
      - Los Lenguajes de Consulta pueden ser **Procedimentales** (hay que especificar cómo obtener los datos) y **No Procedimentales** (hay que especificar qué datos se desean y no cómo obtenerlos).
        - » En general los Procedimentales son más eficientes.
    - **Actualizaciones o Modificaciones a la BD (updates).**
      - **Insertar (Insert):** Introducir nuevos datos en la BD.
      - **Borrar (Delete):** Borrar datos de la BD.
      - **Modificar (Update):** Modificar valores de datos ya existentes.

7

## Gestión de Transacciones

- **TRANSACCIÓN:** Conjunto de operaciones de una aplicación de Bases de Datos que se efectúan como una única operación lógica.
  - **Atomicidad:** Una transacción se considera como una única operación lógica, aunque esté formada por varias operaciones más simples.
    - Debe controlarse que una transacción no sea ejecutada “a medias”.
    - Una transacción, o se ejecuta completamente o no se ejecuta.
  - **Consistencia:** Antes y después de cada transacción, la BD debe tener valores lógicos, aceptables o consistentes.
    - Si se produce un fallo del sistema en medio de la transacción, se deben anular las primeras operaciones de la transacción ya efectuadas: Los valores de la BD deben ser **consistentes**.
    - La consistencia de la BD debe asegurarse incluso aunque varias transacciones se efectúen concurrentemente.
  - **Durabilidad:** Tras una transacción, los nuevos valores de la BD deben mantenerse a pesar de cualquier tipo de fallo del sistema.
- En sistemas de BD pequeños pueden no cumplirse todos estos requisitos o pueden evitarse los problemas que generan (por ejemplo, prohibiendo transacciones concurrentes).

8

## **Funciones del DBA (administrador)**

- **Control Global de la BD:** El DBA es en general el que controla el funcionamiento global de la BD. Puede ser una o varias personas.
- **Definir el Esquema de la BD:** A partir del modelo conceptual, construye sentencias del DDL que se traducirán en estructuras de datos e información en el diccionario de datos del SGBD usado.
- **Definir Estructuras de Almacenamiento:** Se trata de definir cómo y donde se almacenará cada tipo de datos: Datos de la BD y del diccionario, del sistema de recuperación (*rollback system*), de los índices (*indexes*), de las aplicaciones o herramientas (*tools*), de información temporal...
  - La correcta administración de estas estructuras de almacenamiento puede conseguir aumentar mucho la **eficiencia de un SGBD**, lo cual es muy importante.
- **Conceder/Revocar Permisos de Acceso (*grant/revoke*):** Para consultar, insertar, actualizar o borrar datos, pero también para crear objetos (tablas, vistas, índices, usuarios...), ejecutar o crear programas, conceder permisos a terceros...
- **Especificación de Restricciones de Integridad:** Para no admitir valores imposibles o repetidos en la BD, restricciones de integridad referencial...

9

## **Usuarios de la BD**

- **Programadores de Aplicaciones:** Programan aplicaciones de uso más o menos específico o general en un lenguaje de programación de alto nivel. Pueden distinguirse dos Tipos de Aplicaciones:
  - **Aplicaciones Internas al SGBD:** Utilizan un lenguaje de programación que admita el SGBD y permiten la creación de programas que se almacenan y ejecutan "dentro" del SGBD.
    - Este tipo de programas puede ser muy variado, como por ejemplo:
      - Disparadores (*triggers*) que se ejecutan si se cumple cierta condición.
      - Funciones para usarlas en las sentencias del DML (como en la sentencia SELECT de SQL).
      - Procedimientos para cualquier acción, que pueden ser utilizados por otros usuarios (directamente o a través de otro programa).
    - Ejemplo de este tipo de lenguajes es el PL/SQL del SGBD Oracle.
  - **Aplicaciones Externas al SGBD:** Utilizan un lenguaje externo al SGBD (C/C++, Pascal, Cobol...) y dentro de este lenguaje anfitrión se utilizan llamadas especiales del lenguaje DML (como SQL inmerso, *embedded*) que son traducidas por un *precompilador* a sentencias del lenguaje anfitrión.
    - Existen **lenguajes de cuarta generación** que facilitan el control de los objetos de una BD (manipulación, mostrar los datos en una plantilla de datos, generación de formularios...). Por ejemplo, existe el Developer 2000 del SGBD Oracle.
- **Usuarios Sofisticados y Especializados:** Interactúan con el SGBD sin usar programas intermedios, sino directamente con el DML del SGBD. También pueden escribir sus propios programas para los más variados fines (CAD, *Data Mining*, Sistemas Expertos...).
- **Usuarios Normales:** Interactúan con el SGBD a través de alguno de los programas de aplicación. Pueden ser programas muy generales y potentes o muy específicos (como un cajero automático).

10

## ***Componentes de un SGBD***

- **Componentes de Procesamiento de Sentencias DML/DDL:**
  - **Compilador DML:** Traduce las sentencias DML en instrucciones a bajo nivel para ser ejecutadas por el Motor de Ejecución del SGBD. Suelen usar algoritmos para optimizar las sentencias DML y hacerlas más eficientes.
  - **Precompilador DML:** Traduce las sentencias DML incrustadas en cierto programa en sentencias propias de su lenguaje de programación.
  - **Intérprete DDL:** Interpreta las instrucciones DDL y genera los metadatos necesarios en el diccionario de datos.
  - **Motor de Ejecución:** Encargado de que se ejecuten las sentencias ya compiladas, utilizando los componentes de Gestión de Almacenamiento.
- **Componentes de Gestión de Almacenamiento:** Se encargan del acceso directo a los datos solicitados por el Motor de Ejecución.
  - **Gestor de Autorización e Integridad:** Comprueba que el usuario tiene los permisos pertinentes y que no se violan las restricciones de integridad.
  - **Gestor de Transacciones:** Asegura un estado consistente de la BD, aunque se produzcan fallos del sistema o existan transacciones concurrentes.
  - **Gestor de Archivos:** Gestiona el espacio en disco a través de archivos.
  - **Gestor de Memoria Intermedia:** Gestiona los datos que deben traerse/llevarse de disco a Memoria principal.
- **Otros Componentes Almacenados en Disco:** Archivos de datos, diccionario de datos, índices, datos estadísticos...