

Introducción a las BDs

Docente: Ing. Romina Rontomé

¿Qué es una Base de Datos?

Una base de datos es:

Una colección de datos relacionados. Con “datos” nos referimos a los hechos (datos) conocidos que se pueden grabar y que tienen un significado implícito.

Propiedades implícitas:

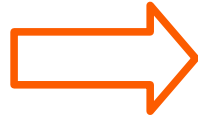
- Representa algún aspecto del mundo real, lo que en ocasiones se denomina minimundo o universo de discurso. Los cambios introducidos en el minimundo se reflejan en la base de datos.
- Es una colección de datos lógicamente coherente con algún tipo de significado inherente. No es correcto denominar base de datos a un surtido aleatorio de datos.
- Se diseña, construye y rellena con datos para un propósito específico. Dispone de un grupo pretendido de usuarios y algunas aplicaciones preconcebidas en las que esos usuarios están interesados.

¿Qué es una Base de Datos?

- El objeto → que sea en todo momento precisa y fiable, debe ser un reflejo exacto del minimundo que representa, deben reflejarse los cambios tan pronto como sea posible.
- Puede ser de cualquier tamaño y complejidad.

¿Qué es un Sistema de Gestión de Bases de Datos?

SGBD (Sistema de Gestión de BD)



Es un sistema de software de propósito general que facilita los procesos de **definición**, **construcción**, **manipulación** y **compartición** de bases de datos entre varios usuarios y aplicaciones.

- Definir → especificar los tipos de datos, estructuras y restricciones de los datos que se almacenarán en la base de datos (metadatos que se almacén en el catálogo o diccionario de datos de la BD)
- Construir → almacenar los datos en algún medio de almacenamiento controlado por el SGDB. Incluye consultas para recuperar datos específicos, actualizar para reflejar los cambios
- Compartir → permite que varios usuarios y programas accedan a la base de datos de forma simultánea.

* Una aplicación accede a la base de datos enviando consultas o transacciones al SGDB. Una consulta habitualmente provoca la recuperación de algunos datos; una transacción puede provocar la lectura o la escritura de algunos datos en la base de datos.

¿Qué es un Sistema de Gestión de Bases de Datos?

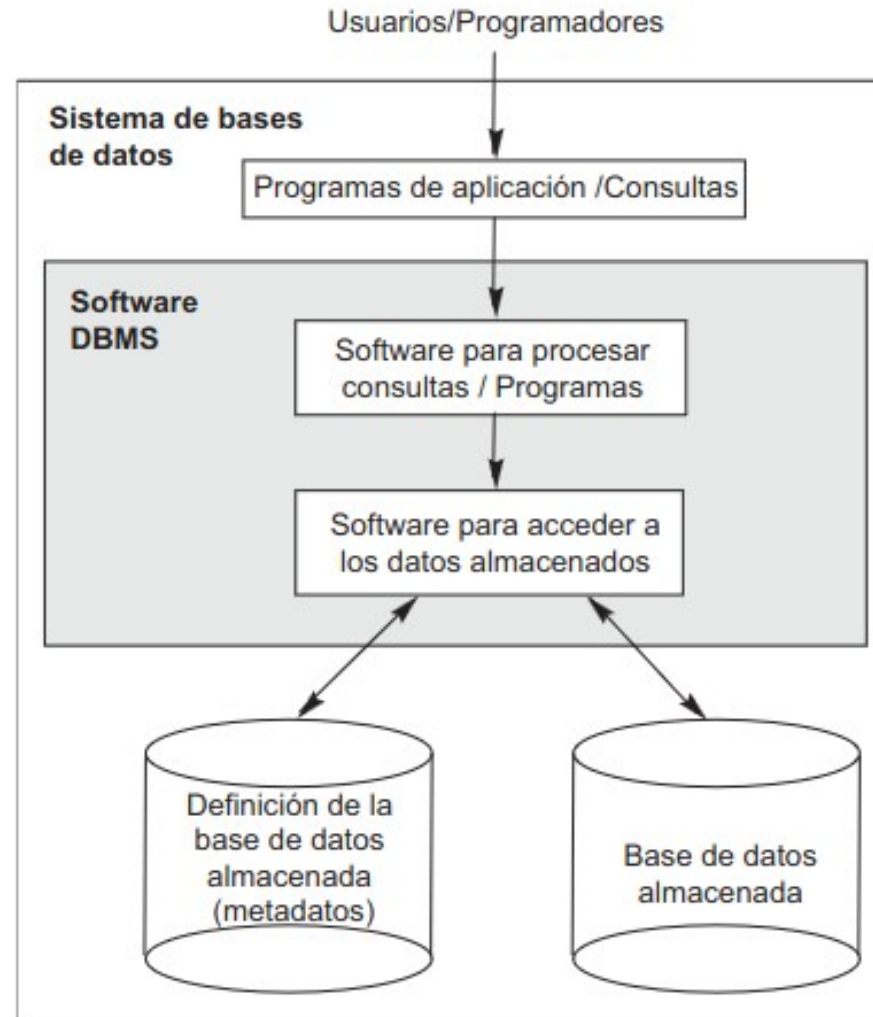
Otras características de los SGBD

- **Protección** → Resguarda el sistema contra el funcionamiento defectuoso del hardware o el software (caídas) y la protección de la seguridad contra el acceso no autorizado o malintencionado.
- **Mantenimiento** → el DBMS debe ser capaz de mantener el sistema de bases de datos permitiendo que el sistema evolucione según cambian los requisitos con el tiempo.

ORACLE®



Entorno de sistema de BD



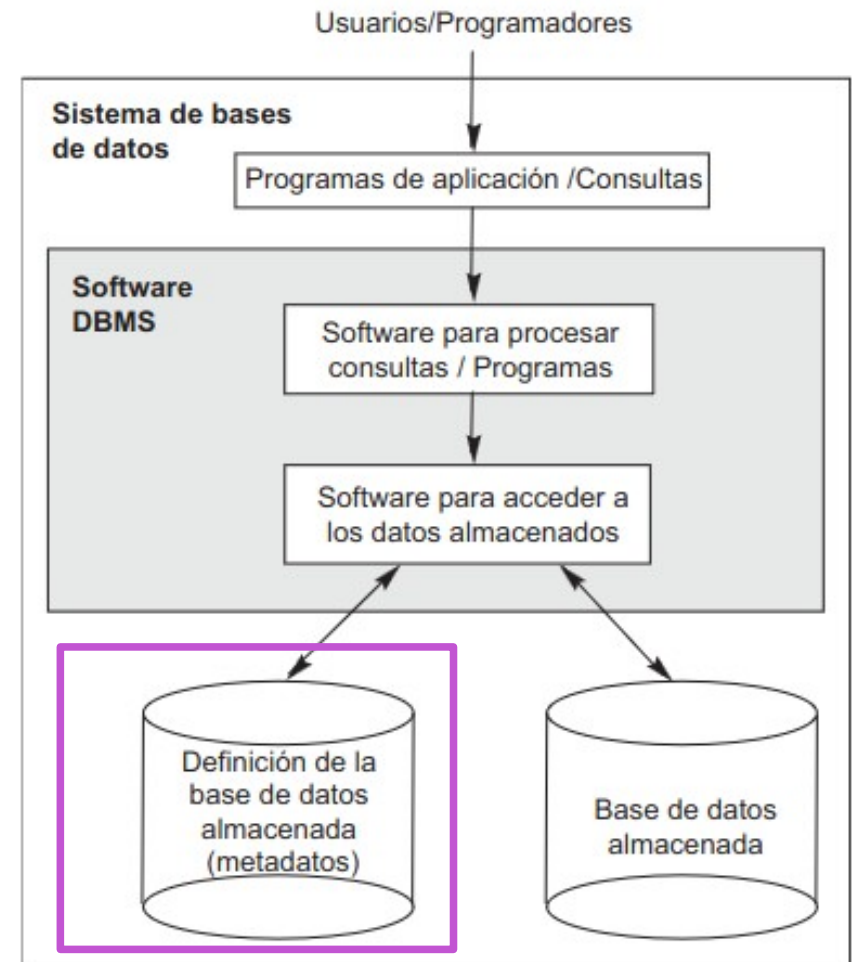
Metodología de trabajo de BD

Comparando la *metodología de archivos* para guardar información con las de *BD* podemos distinguir las siguientes características fundamentales:

- 1) Naturaleza **autodescriptiva** de un sistema de bases de datos.
- 2) Aislamiento entre programas y datos, y abstracción de datos.
- 3) Soporte de varias vistas de los datos.
- 4) Compartición de datos y procesamiento de transacciones multiusuario

1. Naturaleza autodescriptiva de un SGBD

El sistema de bases de datos no sólo contiene la propia base de datos, sino también una completa definición o descripción de la estructura de la base de datos y sus restricciones. Esta definición se almacena en el catálogo DBMS, que contiene información como la estructura de cada archivo, el tipo y el formato de almacenamiento de cada elemento de datos, y distintas restricciones de los datos.



2. Aislamiento entre programas y datos, y abstracción de datos.

Es la propiedad que describe la *estructura de los archivos de datos se almacena en el catálogo* *DBMS,* *independientemente de los programas de acceso.*

Independencia programa-datos.



independencia programa-operación



independencia programa-operación

Operar sobre los datos invocando operaciones por sus nombres y argumentos, independientemente de cómo estén implementadas las operaciones.

Abstracción de datos

(Para los SGBD es sinónimo de Modelo de Datos)

3. Soporte de varias vistas de los datos

Una base de datos tiene muchos usuarios, cada uno de los cuales puede necesitar una perspectiva o vista diferente de la base de datos.

- Una **vista** puede ser un subconjunto de la base de datos o puede contener datos virtuales derivados de los archivos de la base de datos pero que no están explícitamente almacenados.
- Un **DBMS multiusuario** cuyos usuarios tienen variedad de diferentes aplicaciones debe ofrecer **facilidades para definir varias vistas**

4. Compartición de datos y procesamiento de transacciones multiusuario

El DBMS debe incluir software de control de la concurrencia para que, al intentar actualizar los mismos datos varios usuarios a la vez, lo hagan de un modo controlado para que el resultado de la actualización sea correcto.

En aplicaciones de procesamiento de transacciones en línea es fundamental que software DBMS multiusuario garantice que las transacciones concurrentes operan correcta y eficazmente.

Para garantizar una **transacción** se deben cumplir 2 propiedades:

- **Aislamiento:** garantiza que parezca que cada transacción se ejecuta de forma aislada de otras transacciones, aunque puedan estar ejecutándose cientos de transacciones al mismo tiempo.
- **Atomicidad:** garantiza que se ejecuten o todas o ninguna de las operaciones de bases de datos de una transacción.

Actores que participan en la interacción con una BD

- **Administradores de BD - DBA:** responsables del acceso autorizado a la base de datos, de la coordinación y monitorización de su uso, de adquirir los recursos software y hardware necesarios y de responder ante problemas como las brechas de seguridad o de unos tiempos de respuesta pobres.
- **Diseñadores de las bases de datos:** son los responsables de identificar los datos que se almacenarán en la base de datos y de elegir las estructuras apropiadas para representar y almacenar esos datos. Deben comunicarse con los presuntos usuarios de la base de datos para conocer sus requisitos, a fin de crear un diseño que satisfaga sus necesidades.

Actores que participan en la interacción con una BD

- **Usuarios finales:** son las personas cuyos trabajos requieren acceso a la base de datos para realizar consultas, actualizaciones e informes. Entre ellos se encuentran usuarios sw enlatados, aquellos que acceden a través de lenguaje de consultas, científicos, ingenieros, analistas y usuarios finales que mantienen sus BD personales.
- **Analistas de sistemas y programadores de aplicaciones (ingenieros de software):**
 - Los analistas de sistemas determinan los requisitos de los usuarios finales.
 - Los programadores (desarrolladores o ingenieros de SW) de aplicaciones implementan esas especificaciones como sistemas; después, verifican, depuran, documentan y mantienen las soluciones.

Deben familiarizarse con todas las posibilidades proporcionadas por el SGBD para desempeñar sus tareas.

Conceptos Básicos de BDs

- **Independencia física:**

- Simplifica el desarrollo de las aplicaciones, que no han de preocuparse del almacenamiento de los datos.
- Permite realizar cambios en la estructura física para optimizar el uso de la BD sin tener que modificar las aplicaciones.

- **Independencia lógica:**

- Facilita la realización de cambios en las aplicaciones sin tener que modificar la base de datos.
- Cada aplicación organiza los datos en función de sus propias necesidades.

- **Lenguaje de consulta:** Facilitan al usuario el acceso a los datos.

- **Centralización:** Gestión automática de accesos concurrentes.

- **Consistencia de datos:** Mantenimiento de la integridad de los datos. P.ej. Datos duplicados, valores incorrectos...

- **Seguridad:** Control sobre el acceso a los datos particulares.

Beneficios adicionales de utilizar BD

- **Potencial para implementar estándares.** Facilitando la comunicación y cooperación de distintos sectores de usuarios (en empresas grandes) que se encuentran relacionados.
- **Tiempo de desarrollo de aplicación reducido.** Se estima que el tiempo de desarrollo utilizando un SGBD es de una sexta a una cuarta parte del necesario para un sistema de archivos tradicional.
- **Flexibilidad.** Los SGBD permiten ciertos tipos de cambios evolutivos en la estructura de la BD sin que ello afecte a los datos almacenados y a los sistemas de aplicación existentes.
- **Disponibilidad de la información actualizada.** Las actualizaciones en la BD están disponibles inmediatamente. Ésta es una característica imprescindible en aplicaciones de procesamiento transaccionales (ej: Cajeros, reserva de vuelos...)
- **Economías de escala.** Al consolidar la información de distintas aplicaciones en una misma organización evita el retrabajo, la compra de equipamiento individual por depto generando la reducción de los costes globales de funcionamiento y administración.

Evolución histórica y tipos de modelos de BDs

Los modelos más conocidos son:

- Jerárquico (IMS)
 - estructura jerárquica, de árbol
 - implementa punteros para relacionar los datos
- Red ó network (CODASYL DBTG)
 - estructura de grafo
 - implementa punteros
- Relacional (ORACLE, SyBase, DB2, SQL Server, Informix)
 - estructura de tablas: filas y columnas
 - no tiene punteros, se relaciona a través de atributos claves
- Orientada a objetos (OO)
 - Clases, herencia, polimorfismo
 - asociaciones, agregación, composición
- Objeto-Relacional (ORACLE, DB2, Informix)
 - estructura de tablas: filas y columnas
 - no tiene punteros, se relaciona a través de atributos claves

Evolución de las
Bases de Datos

1970

1980

1990

2000



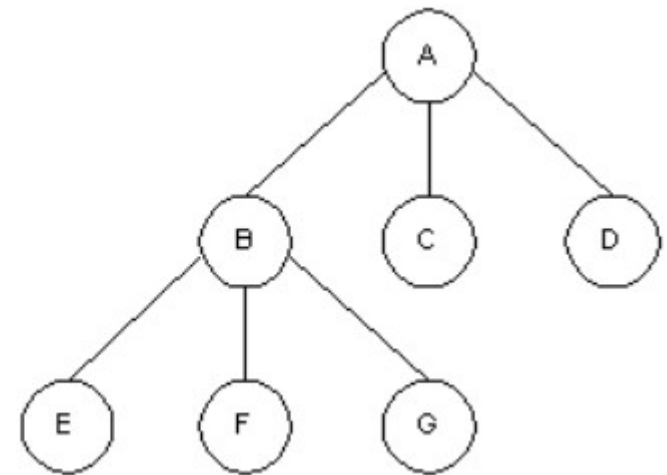
Bases de datos Jerárquica

Un DBMS jerárquico utiliza jerarquías o árboles para la representación lógica de los datos. Los archivos son organizados en jerarquías, y normalmente cada uno de ellos se corresponde con una de las entidades de la base de datos. Los árboles jerárquicos se representan de forma invertida, con la raíz hacia arriba y las hojas hacia abajo.

Nivel 1:
Raíz

Nivel 2

Nivel 3



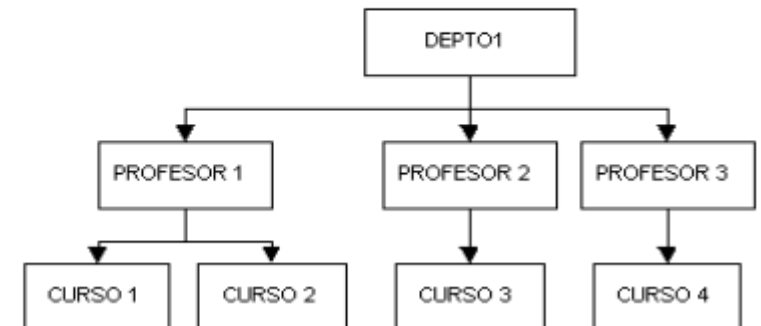
Bases de datos Jerárquica

Cada nodo del árbol representa un tipo de registro conceptual, es decir, una entidad. A su vez, cada registro o segmento está constituido por un número de campos que los describen – las propiedades o atributos de las entidades. Las relaciones entre entidades están representadas por las ramas. En el ejemplo cada departamento es una entidad que mantiene una relación de uno a muchos con los profesores, que a su vez mantienen una relación de uno a muchos con los cursos que imparten.

Estructura lógica



Ejemplo de base de datos

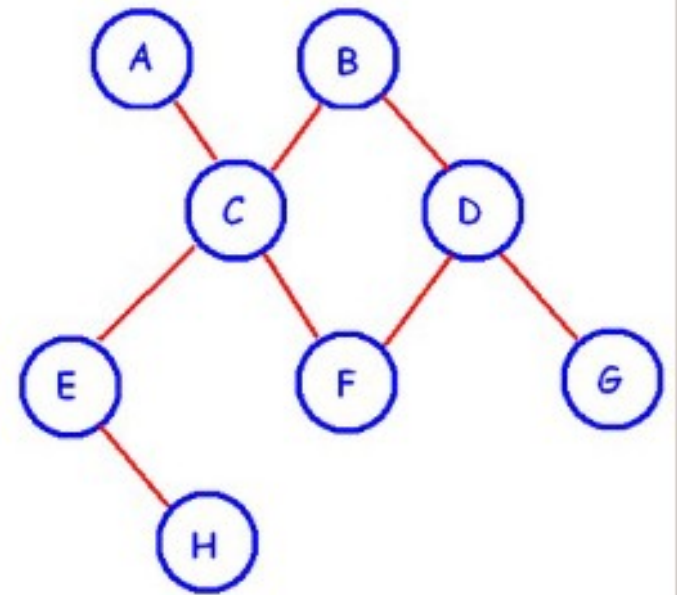


Bases de datos Jerárquica - Desventajas

- Al borrar un nodo padre, desaparecen también sus nodos subordinados.
- Sólo podrá añadirse un nodo hijo, si existe el nodo padre.
- Rigidez de su estructura: sólo un padre por hijo y ausencia de relaciones entre los nodos hijos.

Bases de datos en Red

Una estructura de datos en red, o estructura plex, es muy similar a una estructura jerárquica, de hecho no es más que un súper conjunto de ésta. Al igual que en la estructura jerárquica, cada nodo puede tener varios hijos pero, a diferencia de ésta, también puede tener varios padres. En el ejemplo se muestra una disposición plex. En esta representación, los nodos C y F tienen dos padres, mientras que los nodos D, E, G y H tienen sólo uno.



Bases de datos Relacionales

- Las bases de datos representan un conjunto de tablas relacionadas entre sí.
- Los conceptos básicos del modelo relacional son:
 - **Registro:** Es algo así como cada ficha de un fichero convencional.
 - **Tabla:** Es un conjunto de fichas de un mismo tipo.
- Este modelo es el más utilizado en la actualidad para reducir problemas de administración de datos dinámicamente.
- En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia.
- La información puede ser recuperada o almacenada por medio de consultas, que ofrecen una amplia flexibilidad y poder para administrar la información.

Lenguajes

Lenguaje de Definición de Datos (DDL: Data Definition Language)

Permite definir los datos a nivel conceptual, se genera un archivo especial denominado diccionario de datos (DATA DICTIONARY).

Por ej: Describir y dar nombre a los datos que se requieren para cada aplicación, junto a las reglas que garantizan su integridad y seguridad. CREATE TABLE // ALTER TABLE // DROP TABLE.

Lenguaje de Manipulación de Datos (DML: Data Manipulation Language)

Tiene sentencias que permiten recuperar la información como también ingresar, modificar y eliminar datos.

Por ej: Consultar, añadir, modificar o borrar datos de la base de datos. SELECT FROM // DELETE FROM // UPDATE TABLE