BASES DE DATOS UD-1

ALMACENAMIENTO DE LA INFORMACIÓN (II)

BASES DE DATOS

ÍNDICE

[BASES DE DATOS 2](#_Toc116110488)

[A. BASE DATOS Y SISTEMA GESTOR DE BASE DE DATOS 2](#_Toc116110489)

[B. DEFINICIÓN DE BD 2](#_Toc116110490)

[C. MODELOS DE BASE DE DATOS 3](#_Toc116110491)

[D. EJEMPLO NOSQL: BIG TABLE 6](#_Toc116110492)

# BASE DE DATOS Y SISTEMA GESTOR DE BASE DE DATOS

Una **base de datos (BD)** es un conjunto de datos que tienen relación lógica entre sí, los cuales están en un sitio físico.

Para interactuar con los datos, vamos a usar un programa denominado **sistema gestor de base de datos (SGBD)**.

Los usuarios hablan con ese sistema a través de las aplicaciones que realiza peticiones al SGBD, las cuales se pasan a la BD solicitándole unos datos los cuales recibe de vuelta y entrega al usuario en pantalla.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamenteLa arquitectura básica de la BD es la siguiente:

El usuario o programador solicita información al sistema BD, esta información es procesada por el gestor y, a continuación, solicita los datos a la BD, la cual se los devuelve y, a continuación, el gestor se los muestra al usuario.

# DEFINICIÓN DE BD

Es una colección de datos relacionados lógicas entre sí. Características:

* + **Estructura determinada**. Existen varias estructuras según el modelo de BD.
  + **Definición y descripción comunes**. En algún lugar de la BD hay un lugar donde hay definiciones y descripciones comunes.
  + Almacenamiento con la **mínima** **redundancia**. La BD intenta optimizar el acceso y modificación de datos.
  + Posibilidad de **acceso** **eficiente** por parte de usuarios y aplicaciones.

La base de datos no contiene sólo a los datos, sino una descripción de estos (tipo de datos). Esta descripción es lo que se denomina **metadatos** y se almacena en el **diccionario de datos** o **catálogo**.

# MODELOS DE BASE DE DATOS

Existen diferentes tipos de BD que se pueden utilizar. Según cómo estructura los datos hablamos de **modelo lógico de base de datos**. El modelo **no** determina cómo se visualizan los datos (de ellos se encargará la aplicación), simplemente guarda los datos internamente.

* + **Modelo jerárquico o en árbol**. Ya no se utiliza.
    - Los datos se estructuran en nodos o segmentos que contienen atributos o campos.
    - Los nodos se relacionan entre sí mediante jerarquías padre/hijo.
    - La forma visual es de árbol invertido.
    - Problema: Cada nodo/segmento sólo puede tener un padre.

Una captura de pantalla de una computadora

Descripción generada automáticamente

* + **Modelo en red**.
    - Una captura de pantalla de una computadora

      Descripción generada automáticamenteCada nodo puede tener varios padres.
  + **Modelo relacional**. La BD en este modelo es percibida por el usuario como un conjunto de tablas bidimensionales.

En una tabla o relación tenemos que:

* + - Cada fila se llama **registro**, **entidad** o **tupla**.
    - Cada columna se llama **campo** o **atributo**.
    - Al conjunto de valores que puede tomar un atributo, se le llama **dominio**.
    - Una **clave** será un atributo o conjunto de atributos que identifique de forma única a una tupla. Requiere **normalización** de la información.

Se emplea el lenguaje **SQL**.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* + **Modelo multidimensional**. Se usa en aplicaciones o entornos específicos que requieren consultas complejas y de alto rendimiento. La información se representa en tablas multidimensionales (hipercubos), cada dimensión o eje representa un campo o métrica.
  + Interfaz de usuario gráfica, Aplicación

    Descripción generada automáticamente**Modelo orientado a objetos**. Desarrollada a partir de la programación orientada a objetos (POO). Define objetos con atributos y métodos, los objetos con la misma estructura y comportamiento pertenecen a una clase y las clases se organizan en jerarquías.

Elimina limitaciones del modelo relacional e incorporan mejoras como la herencia entre tablas, los tipos definidos por el usuario e integra datos multimedia de forma natural.

* + **Modelo objeto-relacional**. Es una extensión de la base de datos relacional tradicional, a la cual se le proporcionan características de la POO.

Los registros pueden ser a la vez tablas. Posibilita guardar objetos más complejos en una sola tabla con referencias a otras.

* + **Modelo NoSQL (No relacional)**. Se utiliza mucho cuando hay un gran volumen de datos y se requiere que el tiempo de respuesta sea corto. Suelen ser bases de datos distribuidas.

|  |  |
| --- | --- |
| RELACIONAL | NO RELACIONAL |
| Datos estructurados empleando tablas y registros | Datos no estructurados o semiestructurados |
| Datos normalizados y del mismo tipo | Datos heterogéneos |
| Lenguaje SQL | Lenguajes propios |

Existen varios modelos para almacenar y acceder a los datos:

* + - **Almacén de documentos (*document* *storage*).** Cada dato es un documento basado en JSON o XML.
    - **Almacén de clave-valor (*key-value storage*)**. Cada dato es una pareja clave-valor. Por ejemplo, “dni” – “11111A” (“clave” – “valor”).
    - **Almacén de columna ancha (*wide-column storage*).** Existe una especie de tabla, pero cada fila tiene un número indeterminado de columnas.
    - **Almacén de grafos (*graph storage*).** Cada dato es un grafo que tiene relación con otros.

# EJEMPLO NOSQL: BIG TABLE

Es una BD propietaria de Google para guardar datos de sus aplicaciones. La información de Google se genera a una gran velocidad y está representada por datos muy heterogéneos.

En una BD de tipo relacional sería imposible dar una respuesta rápida y obligaría a establecer un diccionario de datos previamente, lo cual también es imposible.

Está montada sobre **Google File System (GFS)** en la que los ficheros están distribuidos en diferentes máquinas. La base de datos no realiza réplicas de los datos, lo realiza GFS.

Esto ha dado lugar al **Cloud Big Table** que puede ser contratado por las empresas.

Tiene la capacidad de guardar y acceder a enormes cantidades de datos a gran velocidad. Además, es **escalable**.

Es de tipo ***Wide-Column Storage***.

No se realizan **chequeos de coherencia** de los datos.