UD1

DISEÑO CONCEPTUAL DE BASES DE DATOS

1.1 Sistemas de almacenamiento de la información

MP_0484

Bases de Datos

Introducción

Vamos a partir de que uno de los elementos más valiosos que tenemos en un sistema informático son los datos: hay que protegerlos, usarlos, almacenarlos y tratarlos como corresponde. Desde su origen, esto ha sido una necesidad constante y por supuesto las formas y tecnologías para darle solución han sido variadas. Hoy en día, usamos las bases de datos y los sistemas gestores de base de datos (SGBD) para almacenarlos. Trataremos estos tanto desde el punto de vista de las funciones que implementan como la arquitectura y los tipos de SGBD que hay en la actualidad.

Este tema es el primero del módulo donde se presentan los conceptos fundamentales. Facilita la comprensión de las unidades posteriores y proporciona una visión general de las redes de computadoras, lo que permitirá un estudio más detallado en los siguientes temas.

Sistemas lógicos de almacenamiento

Vamos a analizar los sistemas de almacenamiento de datos desde el punto de vista de cómo una aplicación almacena los datos que gestiona. No vamos a confundir esto con el almacenamiento de los datos por parte del sistema operativo.

Almacenamiento lógico y almacenamiento físico

El almacenamiento lógico y físico de los datos son dos conceptos fundamentales en la gestión de la información. Aunque están estrechamente relacionados, representan diferentes aspectos del manejo de datos en un sistema informático.

- **Almacenamiento lógico**: se refiere a la forma en que los datos son percibidos y organizados desde la perspectiva del usuario o del sistema. Se centra en la estructura, el formato y la manera en que se accede y se manipula la información.
- **Almacenamiento físico**: se refiere a cómo se almacenan realmente los datos en los dispositivos de almacenamiento, como discos duros, SSD, etc. Se centra en detalles técnicos, bloque del disco, sector, sistemas de archivos, fragmentación.

Sistemas lógicos de almacenamiento

En general podemos decir que los datos son uno de los bienes preciados de las empresas. Si hablamos del sistema informático de una empresa, ahí sí que podemos afirmar que es el bien más preciado: una pérdida de datos puede ser una situación crítica.

En todas las empresas se tratan los datos, se crean, se actualizan, se consultan y se transforman en otros. Toda esta información, todos estos datos se deben guardar en algún lugar, para ello tenemos diferentes sistemas lógicos de almacenamiento de datos. Las técnicas empleadas para almacenar datos son sumamente importantes para la velocidad de acceso y recuperación de estos. Estos sistemas pueden dividirse principalmente en dos categorías: sistemas de ficheros y sistemas de bases de datos.

Sistemas de ficheros

Los sistemas de ficheros son un conjunto de programas que prestan un servicio a los usuarios finales. Cada programa define y maneja sus propios datos. El origen de estos sistemas se remonta a la sustitución de los archivadores manuales con el objetivo de proporcionar un acceso más eficiente a los datos. Es un modelo descentralizado en el que cada sección, departamento, almacena y gestiona sus propios datos. Es obvio decir que es un sistema que está actualmente en desuso. Analizando los problemas que se encontraban por el uso de estos, es fácil entender la necesidad y las ventajas que aporta el uso de las bases de datos como sistema de almacenamiento lógico.

- **Duplicación de datos**: los sistemas de ficheros para su correcto funcionamiento necesitan ficheros duplicados. Esto genera una duplicidad con el consiguiente desaprovechamiento del espacio de almacenamiento y, lo que es más importante, tener datos duplicados nos puede llevar que no coincida el contenido (inconsistencia). Se pueden llegar a producir cosas a lo largo del tiempo, como tener datos de los empleados en dos ficheros diferentes con dos direcciones o teléfonos diferentes.
- **Dependencia de estructura del fichero**: la estructura de los ficheros en donde se almacenan los datos hace que se tenga que tratar el acceso a estos de una u otra forma: es diferente acceder de forma secuencial a un fichero que de forma indexada; por lo tanto, el programador debe tener en cuenta la estructura del fichero para realizar el programa. El problema se genera cuando un cambio en la estructura del fichero conlleva a modificaciones significativas en la programación de la aplicación.
- **Separación y aislamiento de datos**: al tener los datos descentralizados, separados o incluso duplicados en varios sitios, el programador debe acceder a todos y sincronizar el procesamiento de los ficheros implicados para asegurar que se tratan correctamente.
- **Ficheros con formatos incompatibles**: las estructuras de los ficheros pueden presentar una fuerte dependencia del lenguaje de programación utilizado. Esto puede derivar en incompatibilidades entre distintos ficheros. Por lo tanto, dificultará el procesamiento en modo conjunto.

Sistemas de bases de datos

De forma sencilla se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son almacenados y explotados por el sistema de información de una empresa. Las bases de datos son sistemas orientados a los datos que se deben entender como una estructura lógica diseñada para almacenar, organizar y gestionar grandes volúmenes de datos de manera eficiente. Algunas de las características más destacadas son:

- Los datos se organizan en una estructura definida y con independencia respecto de los programas.
- Menos redundancia: los datos se relacionan entre sí, no se repiten.
- Integridad de los datos: implica que no haya incoherencias entre estos y se puedan evitar pérdidas o actualizaciones indebidas.
- Mayor seguridad en los datos, al permitir limitar el acceso a los usuarios.
- Acceso a los datos de forma eficiente. La organización de los datos a través del sistema gestor de base de datos permitirá un resultado más optimo en el rendimiento.
- El acceso concurrente a los datos por múltiples usuarios es posible sin que haya incongruencias, pudiendo especificar niveles de acceso para los usuarios.
- Datos más documentados, gracias a los metadatos que permiten describir la información de la base de datos.
- Proporcionan un soporte para transacciones, lo que asegura que las operaciones se realicen de manera voluntaria y, una vez validadas, los datos se mantengan en estado válido incluso en caso de fallos.

Aplicaciones actuales de las bases de datos

Aplicaciones web y móviles

- **Redes sociales**: plataformas como Facebook, X, Instagram, etc., utilizan bases de datos para almacenar perfiles de usuarios, publicaciones, conexiones sociales, etc.
- **Comercio electrónico**: sitios de comercio electrónico como Amazon, eBay, Alibaba, etc., gestionan inventarios, historiales de pedidos, perfiles de clientes y otra información relacionada con las transacciones.
- **Aplicaciones de mensajería**: aplicaciones de mensajería instantánea como WhatsApp, Telegram, Signal, etc., almacenan conversaciones, contactos y otros datos de usuarios.
- **Aplicaciones de viajes**: plataformas como Airbnb, Booking.com, Uber, etc., utilizan bases de datos para gestionar listados de alojamientos, reservas, historiales de viajes, etc.

Sistemas Empresariales

- **Gestión de relaciones con clientes (CRM)**: software CRM como Salesforce, HubSpot, Microsoft Dynamics, etc., almacena información sobre clientes, oportunidades de ventas, interacciones y actividades.
- **Sistemas de gestión de recursos humanos (HRM)**: plataformas de HRM como Workday, BambooHR, ADP, etc., gestionan datos de empleados, nóminas, beneficios, evaluaciones de desempeño, etc.
- **Sistemas de planificación de recursos empresariales (ERP)**: software ERP como SAP, Oracle ERP, Microsoft Dynamics 365, etc., integran y gestionan datos de diferentes áreas de una organización, como finanzas, logística, producción, etc.

Aplicaciones científicas y de investigación

- **Bases de datos biológicas**: bases de datos como GenBank, Protein Data Bank (PDB), etc., almacenan secuencias genéticas, estructuras de proteínas y otros datos biológicos para investigación.
- **Bases de datos científicas**: bases de datos académicas como PubMed, IEEE Xplore, Google Scholar, etc., almacenan artículos, papers, patentes y otra información científica.
- **Bases de datos de investigación**: bases de datos para investigación en áreas como la física de partículas (CERN), astronomía (NASA), ciencias sociales, etc.

Sistemas de internet de las cosas (IoT)

- **Monitorización y control de dispositivos**: plataformas de IoT como AWS IoT, Google Cloud IoT, etc., almacenan datos generados por dispositivos conectados, como sensores, medidores, cámaras, etc.
- **Análisis de datos de sensores**: utilizan bases de datos para almacenar y analizar datos de sensores en tiempo real, con aplicaciones en la monitorización del clima, la gestión de la energía, la agricultura de precisión, etc.

Aplicaciones de entretenimiento y medios

- **Streaming de contenidos**: plataformas de streaming como Netflix, Spotify, YouTube, etc., almacenan catálogos de películas, música, vídeos y otro contenido multimedia, así como datos de preferencias de los usuarios.
- **Juegos en línea**: Fortnite, League of Legends, etc., utilizan bases de datos para gestionar perfiles de jugadores, estadísticas, logros, rankings, entre otros.

Arquitectura de los sistemas de información

En cualquier sistema de información de base de datos se puede considerar que se observan los datos desde tres puntos de vista (figura 1):

- **Esquema interno – vista interna**: representa la forma en la que están almacenados los datos. Esta visión solo la usa el administrador de la base de datos, la necesita para poder gestionarla más eficientemente.
- **Esquema lógico-conceptual**: se trata de un esquema teórico de los datos, en el que figuran organizados en estructuras reconocibles del mundo real y también aparece la forma en la que se relacionan. Este es el paso que permite modelar un problema real al sistema de almacenamiento. Es el primer paso por realizar para crear una base de datos; lo realizan los diseñadores o analistas. Algunos de los modelos conceptuales usados son modelo entidad-relación, el modelo RM/T, los modelos semánticos, normalización (solo para BD relacionales).
- **Esquema externo – vista externa – mundo real**: el esquema externo es lo que los usuarios ven, habitualmente a través de las aplicaciones que se han diseñado para ellos. La aplicación oculta lo que hay por detrás y muestra a los usuarios solo la parte que les corresponde por ser dicho usuario y con ciertos permisos.

Figura 1. Arquitectura básica de los sistemas de gestión de información basados en BBDD

Sistemas gestores de bases de datos

Los sistemas gestores de bases de datos (SGBD) pueden definirse como un paquete integral de software que se ejecuta en un sistema servidor centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario (figura 2)

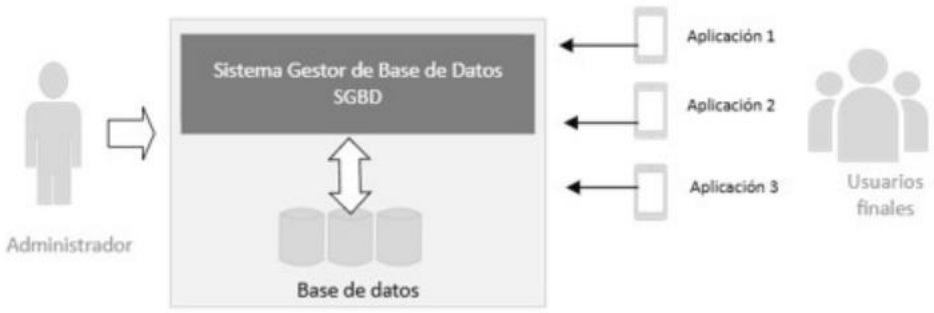

Figura 2. Estructura básica de un SGBD

Los SGBD son un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los diferentes usuarios realizar sus tareas habitualmente con los datos, garantizando, además, la seguridad de estos. Junto con el SGBD, aparece la figura del usuario administrador, que será el que conociendo el SGBD, crea la base de datos en función de las necesidades del usuario y la mantiene; encargándose también de dar los accesos adecuados a otros usuarios, realizar las copias de seguridad, entre otras funciones.

El usuario administrador incluso puede haber participado en el diseño e implementación del esquema conceptual o lógico. No debe confundirse entre SGBD y base de datos. La base de datos es la que almacena los datos, tendremos muchas bases de datos o no en función de nuestras necesidades, para la gestión del personal de la empresa, para los productos del almacén.

Para operar con esas bases de datos se utiliza el SGBD, que solo habrá uno. El administrador podrá interactuar con este para modificar la base de datos, crear nuevas, realizar copias de seguridad, es decir, las funciones de administración de las bases de datos. Adicionalmente, los usuarios utilizarán aplicaciones que les muestran una parte de la base de datos que representa su mundo real, solo lo que puede ver o lo que puede hacer. Las principales funciones que debe cumplir un SGBD se relacionan con:

- Crear y mantener bases de datos.
- Controlar los accesos al sistema.
- Manipular datos de acuerdo con las necesidades de la organización y sus usuarios.
- Velar por el cumplimiento de las normas de tratamiento de los datos.
- Evitar redundancias e inconsistencias en la base de datos.
- Mantener la integridad de los datos.

Podríamos decir que es similar a un sistema operativo que se ocupa del control de acceso a los datos. Este control se basa en una serie de subsistemas encargados de gestionar cada servicio.

Algunos de estos subsistemas son:

- **Sistema de gestión de la memoria**: se encarga de organizar la memoria asignada a cada tarea del SGBD, velando por que no falte memoria para el correcto y eficaz funcionamiento del SGBD sin que afecte negativamente al sistema operativo de la máquina.
- **Sistema gestor de entrada y salida**: se ocupa del adecuado acceso a los datos y la puesta a disposición de estos.
- **Procesador de lenguajes**: se encarga de interpretar las órdenes lanzadas por los usuarios a la BD.
- **Control de procesos**: se encarga de gestionar los programas en ejecución necesarios para el correcto funcionamiento de la base de datos.
- **Control de la red**: se encarga de controlar las conexiones a la base de datos desde la red y evitar problemas en casos de desconexión.
- **Control de transacciones**: permite gestionar las transacciones (conjunto de operaciones de manipulación de datos que se pueden validar o anular).

Modelos de bases de datos

Bases de datos jerárquicas

Este modelo fue utilizado por los primeros SGBD, desde que IBM lo definió para su IMS (Information Management System, Sistema Administrador de Información) en 1970. La información se organiza con estructura jerárquica en elementos llamados nodos. La relación entre ellos es en forma de árbol, donde cada nodo (o registro) está conectado a un padre y puede tener varios nodos hijo.

Podemos encontrar una serie de nodos que contienen atributos. Es importante señalar que, en esta estructura, un nodo puede tener entre ninguno y varios hijos, pero solo puede tener un padre. Un caso especial es el del nodo llamado raíz, este es el punto de entrada a la estructura y es el único que no tiene padre.

En este modelo, los datos se almacenan de forma lógica utilizando estructuras llamadas segmentos. Estos segmentos se relacionan entre sí mediante arcos. La representación gráfica de este modelo es la de un árbol invertido, de forma que los padres están arriba y los hijos abajo (figura 3).

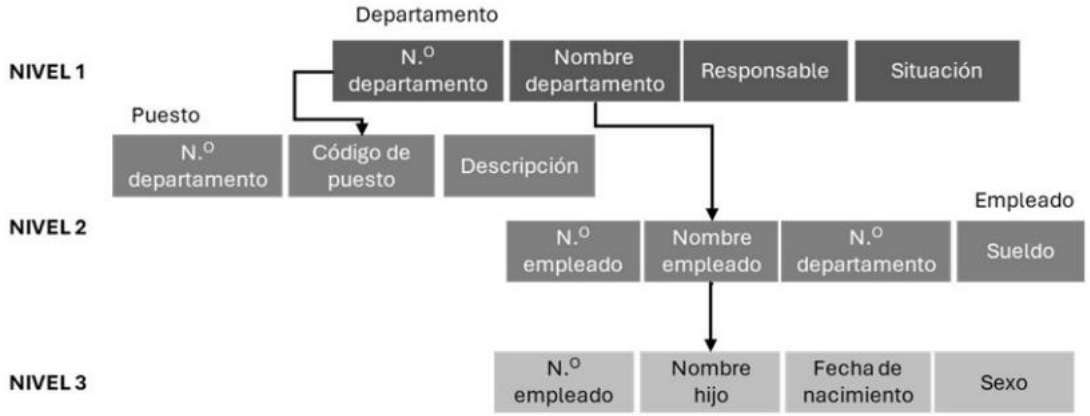

Figura 3. Modelo jerárquico

Este esquema está obsoleto y en desuso, ya que no es válido para modelar la mayoría de las situaciones reales que se deberían poder implementar en una base de datos, solamente permite relaciones de 1 a n de padre a hijos.

Bases de datos en red

A pesar de haber tenido históricamente una gran aceptación, apenas se utiliza en la actualidad. Uno de sus máximos exponentes fue Codasyl, que a principios de los 70, se convirtió en el modelo en red más utilizado. Este modelo organiza la información en registros (o nodos) y conjuntos (o enlaces). Los datos se almacenan en los registros. Las relaciones se representan mediante los conjuntos. Guarda ciertos parecidos al jerárquico, pero en el modelo en red puede haber más de un padre. Esto nos permite representar cualquier tipo de relación entre los datos. La principal desventaja de este modelo es lo complicado de su manejo, ya que es un complejo sistema de punteros, tal y como se muestra en la figura 4. Hoy en día es también un sistema obsoleto.

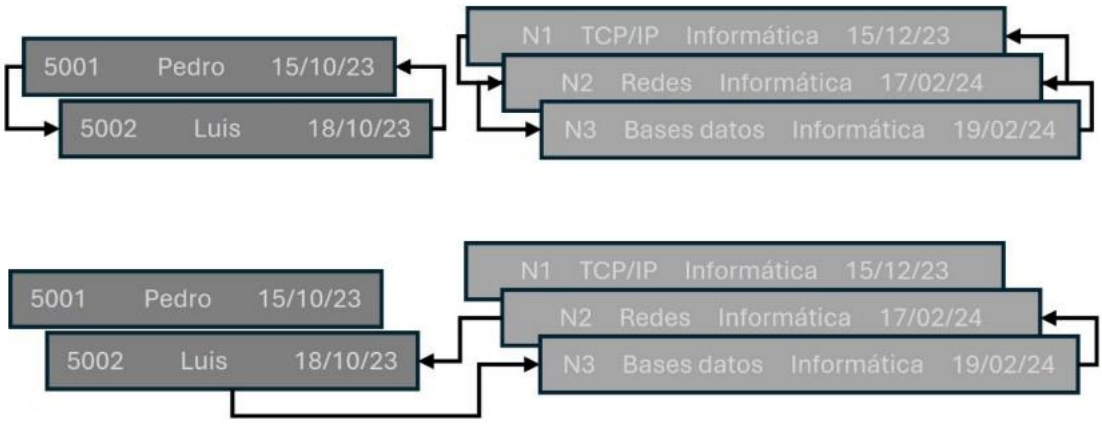

Figura 4. Ejemplo de base de datos en red

Bases de datos relacionales

En 1970, el Dr. Edgar Frank Codd, de los laboratorios de investigación de IBM, publicó un artículo en el que presentó el modelo relacional. En su artículo, también abordó las desventajas de los sistemas previos, como el jerárquico y el de red.

A raíz de esto, se empezaron a desarrollar numerosos sistemas relacionales, apareciendo los primeros a finales de los setenta y principios de los ochenta. Este desarrollo llevó a dos grandes avances:

- El desarrollo de un lenguaje de consultas estructurado llamado **structured query language (SQL)**, que se ha convertido en el estándar para los sistemas relacionales.
- La producción de varios SGBD relacionales durante los años ochenta, que algunos con las actualizaciones pertinentes siguen vigentes hoy en día.

En el modelo relacional los datos son almacenados en relaciones (tablas), de ahí su nombre, que están estructuradas en tuplas (filas) y atributos (columnas). Algunos ejemplos de SGBD relaciones que se usan en la actualidad son los siguientes:

- **MySQL**: utilizado en una variedad de aplicaciones web y empresariales. Muy popular en plataformas de código abierto y comerciales.
- **PostgreSQL**: conocido por su conformidad con el estándar SQL y su extensibilidad. Ideal para aplicaciones que requieren funciones avanzadas.
- **Microsoft SQL Server**: ampliamente utilizado en entornos empresariales y corporativos, con integración profunda en el ecosistema de Microsoft.
- **Oracle Database**: utilizado por grandes empresas para aplicaciones críticas que requieren alta disponibilidad, rendimiento y escalabilidad.

Base de datos orientadas a objetos

La aparición de la programación orientada a objetos (POO u OOP) trajo consigo la necesidad de bases de datos adaptadas a este nuevo paradigma y los lenguajes que lo implementaban. Este estilo de programación permite encapsular, mediante estructuras compuestas, datos (atributos) y procedimientos (métodos).

Esta misma filosofía se aplica en las bases de datos orientadas a objetos, lo que facilita mucho la integración con aplicaciones que utilizan este paradigma de programación.

Bases de datos NoSQL

Las bases de datos NoSQL, o Not Only SQL, son sistemas de gestión de bases de datos que proporcionan un mecanismo para almacenar y recuperar datos que no se modelan en las tablas tradicionales de bases de datos relacionales. Estas bases de datos están diseñadas para manejar grandes volúmenes de datos distribuidos, ofreciendo una mayor flexibilidad en términos de estructura de datos y escalabilidad horizontal. Algunas de las características destacables de las bases de datos NoSQL serían las siguientes:

- **Flexibilidad del esquema**: permiten almacenar datos sin un esquema fijo, lo que facilita el manejo de datos no estructurados y semiestructurados.
- **Escalabilidad horizontal**: diseñadas para escalar de manera horizontal distribuyendo la carga en múltiples servidores.
- **Alto rendimiento**: optimizadas para operaciones de lectura y escritura rápidas, adecuadas para aplicaciones que manejan grandes volúmenes de datos en tiempo real.
- **Alta disponibilidad y tolerancia a fallos**: implementan mecanismos de replicación y particionamiento para asegurar la disponibilidad de datos y tolerancia a fallos.
- **Modelos de consistencia flexibles**: ofrecen distintos niveles de consistencia, desde eventual hasta fuerte, permitiendo a los desarrolladores elegir el equilibrio adecuado entre consistencia y rendimiento.

Bases de datos distribuidas

Las bases de datos distribuidas son sistemas en los que los datos no se almacenan en un único lugar, sino que están repartidos en varios nodos o ubicaciones geográficas diferentes. Estos nodos pueden estar en la misma red local o dispersos globalmente. Esto genera algunas ventajas:

- **Alta disponibilidad**: los datos replicados en varios nodos aseguran esta ventaja.
- **Escalabilidad**: fácil de escalar añadiendo más nodos a la red.
- **Rendimiento**: mejor rendimiento debido a la distribución de la carga de trabajo.

Por el contrario, plantea ciertos desafíos:

- **Consistencia**: mantener la consistencia de los datos en todos los nodos es complejo.
- **Gestión y mantenimiento**: administrar una base de datos distribuida es más complicado que manejar una base de datos centralizada.
- **Latencia**: la latencia de red puede afectar el tiempo de respuesta de las consultas.