

1. Introducción



Como hemos visto anteriormente, los ficheros permiten organizar y memorizar conjuntos de datos del mismo tipo o naturaleza con una determinada estructura, siendo un medio para el almacenamiento de los datos o resultados de una aplicación específica. Pero si las aplicaciones, al ser diseñadas, deben depender directamente de sus ficheros o archivos, se pierde independencia y surgen serios inconvenientes: como información duplicada, incoherencia de datos, fallos de seguridad, etc.

Estos problemas debían ser solucionados, es cuando aparece el concepto de base de datos. Una base de datos permitirá reunir toda la información relacionada en un único sistema de almacenamiento, pudiendo cualquier aplicación utilizarla de manera independiente y ofreciendo una mejora en el tratamiento de la información, así como una evolución para el desarrollo de aplicaciones.

La gestión de las bases de datos ha experimentado gran cantidad de cambios, partiendo de aplicaciones especializadas hasta llegar a convertirse en el núcleo de los entornos informáticos modernos. Con la llegada de Internet en los noventa, el número de usuarios de bases de datos creció exponencialmente, y aunque muchos de ellos no sean conscientes de ello, el acceso a dichas bases forma parte de la vida cotidiana de muchos de nosotros.

Conocer los sistemas que gestionan las bases de datos, sus conceptos fundamentales, el diseño, lenguajes y la implementación de éstas, podemos considerarlo imprescindible para alguien que se está formando en el campo de la informática

2. Conceptos

A finales de los setenta, la aparición de nuevas tecnologías de manejo de datos a través de los sistemas de bases de datos supuso un considerable cambio. Los sistemas basados en ficheros separados dieron paso a la utilización de sistemas gestores de bases de datos, que son sistemas software centralizados o distribuidos que ofrecen facilidades para la definición de bases de datos, selección de estructuras de datos y búsqueda de forma interactiva o mediante lenguajes de programación.

Base de datos: Es una colección de datos relacionados lógicamente entre sí, con una definición y descripción comunes y que están estructurados de una determinada manera. Es un conjunto estructurado de datos que representa entidades y sus interrelaciones, almacenados con la mínima redundancia y posibilitando el acceso a ellos eficientemente por parte de varias aplicaciones y usuarios.

La base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina **metadatos**, se almacena en el **diccionario de datos o catálogo** y es lo que permite que exista **independencia de datos** lógica-física.

Una base de datos constará de los siguientes elementos:

- **Entidades:** objeto real o abstracto con características diferenciadoras de otros, del que se almacena información en la base de datos. En una base de datos de una clínica veterinaria, posibles entidades podrían ser: *ejemplar, doctor, consulta*, etc.
- **Atributos:** son los datos que se almacenan de la entidad. Cualquier propiedad o característica de una entidad puede ser atributo. Continuando con nuestro ejemplo, podrían ser atributos: *raza, color, nombre, número de identificación*, etc.
- **Registros:** donde se almacena la información de cada entidad. Es un conjunto de atributos que contienen los datos que pertenecen a una misma repetición de entidad. En nuestro ejemplo, un registro podría ser: *2123056, Sultán, Podenco, Gris, 23/03/2009*.
- **Campos:** donde se almacenan los atributos de cada registro. Teniendo en cuenta el ejemplo anterior, un campo podría ser el valor *Podenco*.

3. Ventajas de las bases de datos

- **Acceso múltiple:** diversos usuarios o aplicaciones podrán acceder a la base de datos, sin que existan problemas en el acceso o los datos.
- **Utilización múltiple:** cada uno de los usuarios o aplicaciones podrán disponer de una visión particular de la estructura de la base de datos, de tal manera que cada uno de ellos accederá sólo a la parte que realmente le corresponde.
- **Flexibilidad:** la forma de acceder a la información puede ser establecida de diferentes maneras, ofreciendo tiempos de respuesta muy reducidos.
- **Confidencialidad y seguridad:** el control del acceso a los datos podrá ser establecido para que unos usuarios o aplicaciones puedan acceder a unos datos y a otros no, impidiendo a los usuarios no autorizados la utilización de la base de datos.
- **Protección contra fallos:** en caso de errores en la información, existen mecanismos bien definidos que permiten la recuperación de los datos de forma fiable.
- **Independencia física:** un cambio de soporte físico de los datos (por ejemplo: el tipo de discos), no afectaría a la base de datos o a las aplicaciones que acceden a ellos.
- **Independencia lógica:** los cambios realizados en la base de datos no afectan a las aplicaciones que la usan.
- **Redundancia:** los datos se almacenan, por lo general, una única vez. Aunque si es necesario, podríamos repetir información de manera controlada.
- **Interfaz de alto nivel:** mediante la utilización de lenguajes de alto nivel puede utilizarse la base de datos de manera sencilla y cómoda.
- **Consulta directa:** existe una herramienta para poder acceder a los datos interactivamente.

4. Usos

Ya sabemos lo que es una base de datos y sus características principales, pero es necesario conocer quien las usa y para qué.

¿Quién utiliza las bases de datos?

Existen cuatro tipos de personas que pueden hacer uso de una base de datos: el **administrador**, los **diseñadores** de la base de datos, los **programadores** de aplicaciones y los **usuarios** finales.

Tipo	Funciones y características
El administrador	Es la persona encargada de la creación o implementación física de la base de datos. Es quien escoge los tipos de ficheros, los índices que hay que crear, la ubicación de éstos, etc. En general, es quien toma las decisiones relacionadas con el funcionamiento físico del almacenamiento de información. Siempre teniendo en cuenta las posibilidades del sistema de información con el que trabaje. Junto a estas tareas, el administrador establecerá la política de seguridad y de acceso para garantizar el menor número de problemas.
Los diseñadores	Son las personas encargadas de diseñar cómo será la base de datos. Llevarán a cabo la identificación de los datos, las relaciones entre ellos, sus restricciones, etc. Para ello han de conocer a fondo los datos y procesos a representar en la base de datos. Si estamos hablando de una empresa, será necesario que conozcan las reglas de negocio en la que esta se mueve. Para obtener un buen resultado, el diseñador de la base de datos debe implicar en el proceso a todos los usuarios de la base de datos, tan pronto como sea posible.
Los programadores de aplicaciones	Una vez diseñada y construida la base de datos, los programadores se encargarán de implementar los programas de aplicación que servirán a los usuarios finales. Estos programas de aplicación ofrecerán la posibilidad de realizar consultas de datos, inserción, actualización o eliminación de los mismos. Para desarrollar estos programas se utilizan lenguajes de tercera o cuarta generación.
Los usuarios finales	Son los clientes finales de la base de datos. Al diseñar, implementar y mantener la base de datos se busca cumplir los requisitos establecidos por el cliente para la gestión de su información.

¿Para qué se utilizan las bases de datos?

Enumerar todos y cada uno de los campos donde se utilizan las bases de datos es complejo, aunque seguro que quedarán muchos en el tintero, a continuación se recopilan algunos de los ámbitos donde se aplican.

- Banca: información de clientes, cuentas, transacciones, préstamos, etc.
- Líneas aéreas: información de clientes, horarios, vuelos, destinos, etc.
- Universidades: información de estudiantes, carreras, horarios, materias, etc.
- Transacciones de tarjeta de crédito: para comprar con tarjetas de crédito y la generación de los extractos mensuales.
- Telecomunicaciones: para guardar registros de llamadas realizadas, generar facturas mensuales, mantener el saldo de las tarjetas telefónicas de prepago y almacenar información sobre las redes.
- Medicina: información hospitalaria, biomedicina, genética, etc.
- Justicia y Seguridad: delincuentes, casos, sentencias, investigaciones, etc.
- Legislación: normativa, registros, etc.
- Organismos públicos: datos ciudadanos, certificados, etc.
- Sistemas de posicionamiento geográfico.
- Hostelería y turismo: reservas de hotel, vuelos, excursiones, etc.
- Ocio digital: juegos online, apuestas, etc.
- Cultura: gestión de bibliotecas, museos virtuales, etc.
- Etc.

5. Ubicación de la información

Utilizamos a diario las bases de datos, pero ¿Dónde se encuentra realmente almacenada la información?. Las bases de datos pueden tener un tamaño muy reducido (1 MegaByte o menos) o bien, ser muy voluminosas y complejas (del orden de Terabytes). Sin embargo todas las bases de datos normalmente se almacenan y localizan en discos duros y otros dispositivos de almacenamiento, a los que se accede a través de un ordenador. Una gran base de datos puede necesitar servidores en lugares diferentes, y viceversa, pequeñas bases de datos pueden existir como archivos en el disco duro de un único equipo.

A continuación, se exponen los **sistemas de almacenamiento de información más utilizados para el despliegue de bases de datos**, comenzaremos por aquellos en los que pueden alojarse bases de datos de tamaño pequeño y mediano, para después analizar los sistemas de alta disponibilidad de grandes servidores.

- **Discos SATA:** Es una interfaz de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento, como puede ser el disco duro, lectores y regrabadores de CD/DVD/BD, Unidades de Estado Sólido u otros dispositivos. SATA proporciona mayores velocidades, mejor aprovechamiento cuando hay varias unidades, mayor longitud del cable de transmisión de datos y capacidad para conectar unidades al instante, es decir, insertar el dispositivo sin tener que apagar el ordenador. La primera generación específica en transferencias de 150 Megabytes por segundo, también conocida por SATA 150 MB/s o Serial ATA-150. Actualmente se comercializan dispositivos SATA II, a 300 MB/s, también conocida como Serial ATA-300 y los SATA III con tasas de transferencias de hasta 600 MB/s.
- **Discos SCSI:** Son interfaces preparadas para discos duros de gran capacidad de almacenamiento y velocidad de rotación. Se presentan bajo tres especificaciones: SCSI Estándar (Standard SCSI), SCSI Rápido (Fast SCSI) y SCSI Ancho-Rápido (Fast-Wide SCSI). Su tiempo medio de acceso puede llegar a 7 milisegundos y su velocidad de transmisión secuencial de información puede alcanzar teóricamente los 5 MB/s en los discos SCSI Estándares, los 10 MBps en los discos SCSI Rápidos y los 20 MBps en los discos SCSI Anchos-Rápidos (SCSI-2). Un controlador SCSI puede manejar hasta 7 discos duros SCSI.
- **RAID:** acrónimo de *Redundant Array of Independent Disks* o matriz de discos independientes, es un contenedor de almacenamiento redundante. Se basa en el montaje en conjunto de dos o más discos duros, formando un bloque de trabajo, para obtener desde una ampliación de capacidad a mejoras en velocidad y seguridad de almacenamiento. Según las características que queramos primar, se establecen distintos sistemas de RAID.
- **Sistemas NAS:** Es el acrónimo de *Network Attached Storage* ó sistema de almacenamiento masivo en red. Estos sistemas de almacenamiento permiten compartir la capacidad de almacenamiento de un computador (Servidor) con ordenadores personales o servidores clientes a través de una red, haciendo uso de un sistema operativo optimizado para dar acceso a los datos a través de protocolos de comunicación específicos. Suelen ser dispositivos para almacenamiento masivo de datos con capacidades muy altas, de varios Terabytes, generalmente superiores a los discos duros externos y además se diferencian de estos al conectar por red.
- **Sistemas SAN:** Acrónimo de *Storage Area Network* o red de área de almacenamiento. Se trata de una red concebida para conectar servidores, matrices (arrays) de discos y librerías de soporte. La arquitectura de este tipo de sistemas permite que los recursos de almacenamiento estén disponibles para varios servidores en una red de área local o amplia. Debido a que la información almacenada no reside directamente en ninguno de los servidores de la red, se optimiza el poder de procesamiento para aplicaciones comerciales y la capacidad de almacenamiento se puede proporcionar en el servidor donde más se necesite.

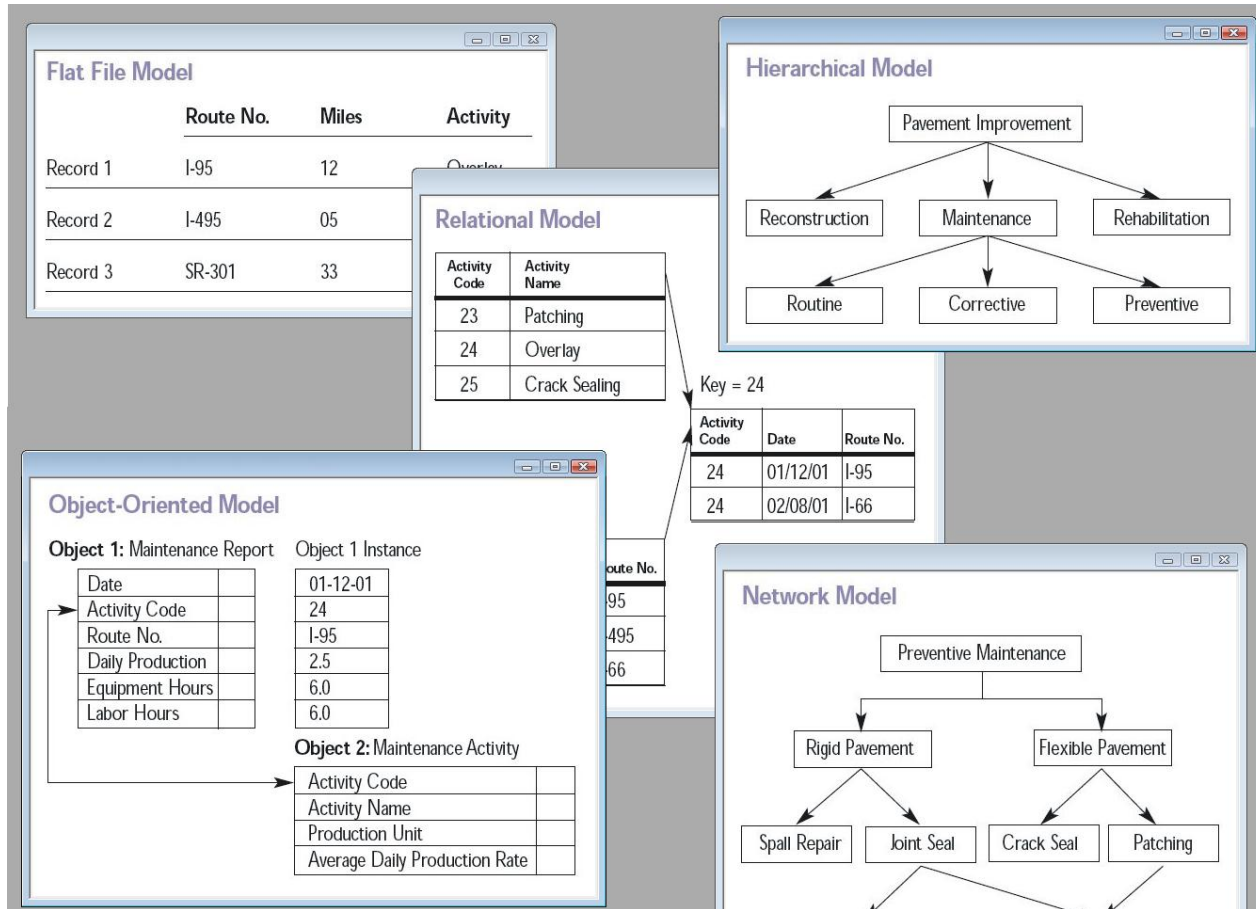
Para saber más

Puedes ampliar más información sobre algunos de los sistemas de almacenamiento vistos, además de tendencias y curiosidades en almacenamiento, a través de los siguientes enlaces:

[Sistemas RAID.](#)
[Sistemas NAS.](#)
[Sistemas SAN.](#)
[Bases de datos en la nube.](#)
[Centros de datos de Google](#)

6. Modelos de bases de datos

La clasificación tradicional de las bases de datos establece tres modelos de bases de datos: **jerárquico**, en **red** y **relacional**. En la actualidad el modelo de bases de datos más extendido es el **relacional**. Aunque, hay que tener en cuenta que dos de sus variantes (modelo de bases de datos **distribuidas** y **orientadas a objetos**) son las que se más se están utilizando en los últimos tiempos.



6.1. Modelo jerárquico

Cuando IBM creó su Sistema Administrador de Información o IMS, se establecieron las bases para que la gran mayoría de sistemas de gestión de información de los años setenta utilizaran el modelo jerárquico. También recibe el nombre de **modelo en árbol**, ya que utiliza una estructura en árbol invertido para la organización de los datos.

La información se organiza con un jerarquía en la que la relación entre las entidades de este modelo siempre es del tipo **padre/hijo**. De tal manera que existen nodos que contienen atributos o campos y que se relacionarán con sus nodos hijos, pudiendo tener cada nodo más de un hijo, pero un nodo siempre tendrá un sólo padre.

Los datos de este modelo se almacenan en estructuras lógicas llamadas segmentos. Los segmentos se relacionan entre sí utilizando arcos. La forma visual de este modelo es de árbol invertido, en la parte superior están los padres y en la inferior los hijos.

Hoy en día, debido a sus limitaciones, el modelo jerárquico está en desuso. En el siguiente gráfico puedes observar la estructura de almacenamiento del modelo jerárquico.

modelo jerárquico de bases de datos:

- **Nodo Raíz:**
 - **Empresa**
 - **Nodo Hijo Nivel 1:**
 - **Departamento de Ventas**
 - **Nodo Hijo Nivel 2:**
 - **Empleado A**
 - **Empleado B**
 - **Departamento de Marketing**
 - **Nodo Hijo Nivel 2:**
 - **Empleado C**
 - **Empleado D**
 - **Departamento de Recursos Humanos**
 - **Nodo Hijo Nivel 2:**
 - **Empleado E**
 - **Empleado F**

6.2. Modelo en red

El modelo de datos en red aparece a mediados de los sesenta como respuesta a limitaciones del modelo jerárquico en cuanto a representación de relaciones más complejas. Podemos considerar a IDS (Integrated Data Store) de Bachman como el primer sistema de base de datos en red. Tras él se intentó crear un estándar de modelo de red por parte de CODASYL, siendo

un modelo que tubo gran aceptación a principios de los setenta.

El modelo en red organiza la información en registros (también llamados nodos) y enlaces. En los registros se almacenan los datos, mientras que los enlaces permiten relacionar estos datos. Las bases de datos en red son *parecidas a las jerárquicas sólo que en ellas puede haber más de un padre*.

En este modelo se pueden representar perfectamente cualquier tipo de relación entre los datos, pero hace muy complicado su manejo. Al no tener que duplicar la información se ahorra espacio de almacenamiento.

El sistema de gestión de información basado en *el modelo en red más popular es el sistema IDMS*.

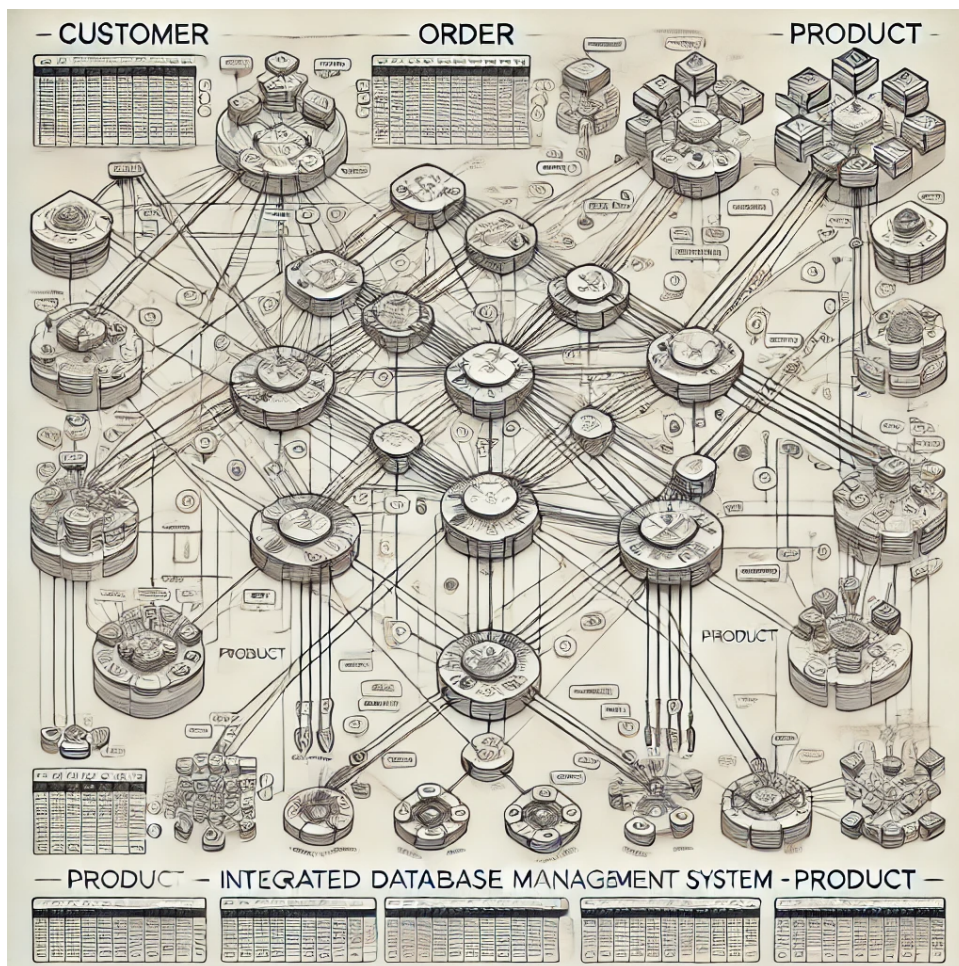
GPT 4 Explicación

El modelo IDMS (Integrated Database Management System) es un tipo de sistema de gestión de base de datos basado en el modelo de red. Fue desarrollado originalmente por Charles Bachman y sus colegas en los años 60 y se ha utilizado ampliamente en entornos empresariales. Este modelo permite una organización más flexible de los datos en comparación con el modelo jerárquico, mediante la utilización de estructuras de red que soportan relaciones muchos a muchos.

Características clave del modelo IDMS:

- **Estructura de red:** En lugar de una estructura de árbol, IDMS utiliza una estructura de red donde un registro (nodo) puede tener múltiples padres y múltiples hijos. Esto permite representar relaciones complejas entre los datos de forma más natural y eficiente.
- **Enlaces explícitos:** Los registros están conectados a través de enlaces o punteros, lo que facilita el acceso y la manipulación de los datos. Esto permite a los usuarios acceder a los datos desde cualquier punto de entrada en la red, sin necesidad de seguir una ruta jerárquica.
- **Integridad de datos:** IDMS ofrece robustas capacidades de integridad de datos, incluyendo el manejo de integridad referencial y transaccional, lo que es crucial para aplicaciones empresariales que requieren consistencia y fiabilidad en el manejo de la información.
- **Alto rendimiento para transacciones complejas:** Es especialmente útil en entornos donde las transacciones son complejas y requieren la manipulación de grandes volúmenes de datos interrelacionados.

El modelo IDMS ha sido fundamental en el desarrollo de los sistemas de bases de datos y continúa siendo relevante en ciertos entornos especializados donde las estructuras de datos complejas son la norma.



6.3. Modelo relacional

Este modelo es posterior a los dos anteriores y fue desarrollado por Codd en 1970. Hoy en día las bases de datos relacionales son las más utilizadas.

En el modelo relacional la base de datos es percibida por el usuario como un conjunto de tablas. Esta percepción es sólo a nivel lógico, ya que a nivel físico puede estar implementada mediante distintas estructuras de almacenamiento.

El modelo relacional utiliza **tablas bidimensionales** (relaciones) para la representación lógica de los datos y las relaciones entre ellos. Cada relación (tabla) posee un nombre que es único y contiene un conjunto de columnas.

Se llamará **registro, entidad o tupla** a cada fila de la tabla y **campo o atributo** a cada columna de la tabla.

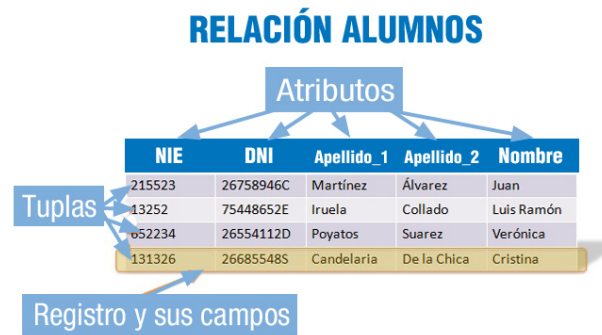
A los conjuntos de valores que puede tomar un determinado atributo, se le denomina **dominio**.

Una **clave** será un atributo o conjunto de atributos que identifique de forma única a una tupla.

Las tablas deben cumplir una serie de requisitos:

- Todos los registros son del **mismo tipo**.
- La tabla sólo puede tener **un tipo** de registro.
- **No** existen campos o atributos **repetidos**.
- **No** existen registros **duplicados**.
- **No** existe **orden** en el almacenamiento de los registros.
- Cada registro o tupla es identificada por una **clave** que puede estar **formada por uno o varios campos o atributos**.

A continuación puedes observar cómo es una relación con sus tuplas y atributos en el modelo relacional.



El lenguaje habitual para construir las consultas a bases de datos relacionales es **SQL, Structured Query Language** o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se conoce como normalización de una base de datos.

6.4. Modelo orientado a objetos

El **modelo orientado a objetos Wikipedia** define una base de datos en términos de **objetos**, sus propiedades y sus operaciones. Los objetos con la misma estructura y comportamiento pertenecen a una **clase**, y las clases se organizan en jerarquías. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados **métodos**. Algunos sistemas existentes en el mercado, basados en el modelo relacional, han sufrido evoluciones incorporando conceptos orientados a objetos. A estos modelos se les conoce como sistemas **objeto-relacionales**.

El objetivo del modelo orientado a objetos es cubrir las limitaciones del modelo relacional. Gracias a este modelo se incorporan mejoras como la herencia entre tablas, los tipos definidos por el usuario, disparadores almacenables en la base de datos (triggers), soporte multimedia, etc.

Los conceptos más importantes del paradigma de objetos que el modelo orientado a objetos incorpora son:

- **Encapsulación** - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- **Herencia** - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- **Polimorfismo** - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

Desde la aparición de la programación orientada a objetos (POO u OOP) se empezó a pensar en bases de datos adaptadas a estos lenguajes. Este modelo es considerado como el fundamento de las bases de datos de tercera generación, siendo consideradas las bases de datos en red como la primera y las bases de datos relacionales como la segunda generación. Aunque no han reemplazado a las bases de datos relacionales, si son el tipo de base de datos que más está creciendo en los últimos años.

6.5. Otros modelos

Además de los modelos clásicos vistos hasta el momento, vamos a detallar a continuación las particularidades de otros modelos de bases de datos existentes y que, en algunos casos, son una evolución de los clásicos.

1. **Modelo Objeto-Relacional**. Las bases de datos pertenecientes a este modelo, son un híbrido entre las bases del modelo relacional y el orientado a objetos. El mayor inconveniente de las bases de datos orientadas a objetos radica en los costes de la conversión de las bases de datos relacionales a bases de datos orientadas a objetos.

En una base de datos **objeto-relacional (BDOR)** siempre se busca obtener lo mejor del modelo relacional, incorporando las mejoras ofrecidas por la orientación a objetos. En este modelo se siguen almacenando tuplas, aunque la **estructura de las tuplas no está restringida** sino que las **relaciones pueden ser definidas en función de otras**, que es lo que denominamos **herencia directa**.

El **estándar** en el que se basa este modelo es **SQL99**. Este estándar ofrece la posibilidad de añadir a las bases de datos relacionales procedimientos almacenados de usuario, triggers, tipos definidos por el usuario, consultas recursivas, bases de datos OLAP, tipos LOB, ...

Otra característica a destacar es la capacidad para incorporar funciones que tengan un código en algún lenguaje de programación como por ejemplo: SQL, Java, C, etc.

La gran mayoría de las bases de datos relacionales clásicas de gran tamaño, como Oracle, SQL Server, etc., son objeto-relacionales.

2. **Modelo de bases de datos orientadas a documentos**. En este modelo el principal objeto de gestión es el documento que contiene datos semi estructurados que podrán estar almacenados en algún formato, por ejemplo XML. Como ejemplo de este tipo de bases de datos puedes encontrar **MongoDB** y **CouchDB**.
3. **Bases de datos multidimensionales**. Son bases de datos **ideadas para desarrollar aplicaciones muy concretas**. Básicamente almacena sus datos con varias dimensiones, es decir que en vez de un valor, encontramos varios dependiendo de los ejes definidos o una base de datos de estructura basada en dimensiones orientada a **consultas complejas y alto rendimiento**. En una base de datos multidimensional, la información se representa como matrices multidimensionales, cuadros de múltiples entradas o funciones de varias variables sobre conjuntos finitos. **Cada una de estas matrices se denomina cubo**. Eso facilita el manejo de grandes cantidades de datos dentro de empresas, dándole a esto una amplia aplicación dentro de varias áreas y diferentes campos del conocimiento humano.
4. **Bases de datos transaccionales**. Son bases de datos caracterizadas por su velocidad para gestionar el intercambio de información, **se utilizan sobre todo en sistemas bancarios, análisis de calidad y datos de producción industrial**. Son bases de datos **muy fiables**, ya que en ellas cada una de las operaciones de inserción, actualización o borrado se realizan

completamente o se descartan.

7. Sistemas Gestores de Bases de Datos

Para poder tratar la información contenida en las bases de datos se utilizan los sistemas gestores de bases de datos o **SGBD** (español), también llamados **DBMS (DataBase Management System)**, que ofrecen un conjunto de programas que permiten acceder y gestionar dichos datos.

El objetivo fundamental de los SGBD es proporcionar eficiencia y seguridad a la hora de recuperar o insertar información en las bases de datos. Estos sistemas están diseñados para la manipulación de grandes bloques de información.

Sistema Gestor de Base de Datos: Conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios no informáticos, como a los analistas programadores, o al administrador, los medios necesarios para describir y manipular los datos contenidos en la base de datos, manteniendo su integridad, confidencialidad y seguridad.

El **SGBD** permite a los usuarios la creación y el **mantenimiento** de una base de datos, facilitando la **definición**, **construcción** y **manipulación** de la información contenida en éstas.

- Definir una base de datos consistirá en especificar los tipos de datos, las estructuras y las restricciones que los datos han de cumplir a la hora de almacenarse en dicha base.
- Por otro lado, la construcción de la base será el proceso de almacenamiento de datos concretos en algún medio o soporte de almacenamiento que esté supervisado por el SGBD.
- Finalmente, la manipulación de la base de datos incluirá la posibilidad de realización de consultas para recuperar información específica, la actualización de los datos y la generación de informes a partir de su contenido.

Las ventajas del uso de SGBD son:

- Proporcionan al usuario una **visión abstracta** de los datos, **ocultando** parte de **la complejidad** relacionada con cómo se almacenan y mantienen los datos.
- Ofrecen **Independencia física**, es decir, la **visión** que tiene de la información el usuario, y la **manipulación** de los datos almacenados en la Base de Datos, es **independiente** de cómo estén almacenados físicamente.
- **Disminuyen la redundancia y la inconsistencia** de datos.
- **Aseguran la integridad** de los datos.
- **Facilitan el acceso** a los datos, aportando **rapidez** y **evitando la pérdida** de datos.
- **Aumentan la seguridad y privacidad** de los datos.
- **Mejoran la eficiencia**.
- **Permiten compartir datos y accesos concurrentes**.
- **Facilitan el intercambio** de datos **entre distintos sistemas**.
- **Incorporan** mecanismos de **copias de seguridad y recuperación** para restablecer la información en caso de fallos en el sistema.

El SGBD interactúa con otros elementos software existentes en el sistema, concretamente con el sistema operativo (SO). Los datos almacenados de forma estructurada en la base de datos son utilizados indistintamente por otras aplicaciones, será el SGBD quien ofrecerá una serie de facilidades a éstas para el acceso y manipulación de la información, basándose en las funciones y métodos propios del sistema operativo.

7.1. Funciones

Un SGBD desarrolla tres funciones fundamentales como son las de descripción, manipulación y utilización de los datos. A continuación se detallan cada una de ellas.

- **Función de descripción o definición:** Permite al diseñador de la base de datos crear las estructuras apropiadas para integrar adecuadamente los datos. Esta función es la que permite definir las tres estructuras de la base de datos: Estructura interna, Estructura conceptual y Estructura externa. (Estos conceptos se verán más adelante en el epígrafe sobre arquitectura del SGBD). Esta función se realiza mediante el **lenguaje de descripción de datos o DDL**. Mediante ese lenguaje: se definen las estructuras de datos, se definen las relaciones entre los datos y se definen las reglas (restricciones) que han de cumplir los datos.
- **Función de manipulación:** permite a los usuarios de la base buscar, añadir, suprimir o modificar los datos de la misma, siempre de acuerdo con las especificaciones y las normas de seguridad dictadas por el administrador. Se llevará a cabo por medio de un **lenguaje de manipulación de datos (DML)** que facilita los instrumentos necesarios para la realización de estas tareas.
- **Función de control:** permite al administrador de la base de datos establecer mecanismos de protección de las diferentes visiones de los datos asociadas a cada usuario, proporcionando elementos de creación y modificación de dichos usuarios. Adicionalmente, incorpora sistemas para la creación de copias de seguridad, carga de ficheros, auditoría, protección de ataques, configuración del sistema, etc. El lenguaje que implementa esta función es el **lenguaje de control de datos o DCL**.

¿Y a través de qué lenguaje podremos desarrollar estas funciones sobre la base de datos? Lo haremos utilizando el **Lenguaje Estructurado de Consultas (SQL: Structured Query Language)**. Este lenguaje proporciona sentencias para realizar operaciones de DDL, DML y DCL. SQL fue publicado por el ANSI en 1986 (American National Standard Institute) y ha ido evolucionando a lo largo del tiempo. Además, los SGBD suelen proporcionar otras herramientas que complementan a estos lenguajes como generadores de formularios, informes, interfaces gráficas, generadores de aplicaciones, etc.

7.2. Componentes

Una vez descritas las funciones que un SGBD debe llevar a cabo, imaginarás que un SGBD es un paquete de software complejo que ha de proporcionar servicios relacionados con el almacenamiento y la explotación de los datos de forma eficiente. Para ello, cuenta con una serie de componentes que se detallan a continuación:

1. **Lenguajes de la base de datos.** Cualquier sistema gestor de base de datos ofrece la posibilidad de utilizar lenguajes e interfaces adecuadas para sus diferentes tipos de usuarios. A través de los lenguajes se pueden especificar los datos que componen la BD, su estructura, relaciones, reglas de integridad, control de acceso, características físicas y vistas externas de los usuarios. Los lenguajes del SGBD son: Lenguaje de Definición de los Datos (**DDL**), Lenguaje de Manejo de Datos (**DML**) y Lenguaje de Control de Datos (**DCL**).
2. **El diccionario de datos.** Descripción de los datos almacenados. Se trata de información útil para los programadores de aplicaciones. Es el lugar donde se deposita la información sobre la totalidad de los datos que forman la base de datos. Contiene las características lógicas de las estructuras que almacenan los datos, su nombre, descripción, contenido y organización. En una base de datos relacional, el diccionario de datos aportará información sobre:
 - Estructura lógica y física de la BD.
 - Definición de tablas, vistas, índices, disparadores, procedimientos, funciones, etc.
 - Cantidad de espacio asignado y utilizado por los elementos de la BD.
 - Descripción de las restricciones de integridad.
 - Información sobre los permisos asociados a cada perfil de usuario.
 - Auditoría de acceso a los datos, utilización, etc.
3. **El gestor de la base de datos.** Es la parte de software encargada de garantizar el correcto, seguro, íntegro y eficiente acceso y almacenamiento de los datos. Este componente es el encargado de proporcionar una interfaz entre los datos almacenados y los programas de aplicación que los manejan. Es un intermediario entre el usuario y los datos. Es el encargado de garantizar la privacidad, seguridad e integridad de los datos, controlando los accesos concurrentes e interactuando con el sistema operativo.
4. **Usuarios de la base de datos.** En los SGBD existen diferentes perfiles de usuario, cada uno de ellos con una serie de permisos sobre los objetos de la BD. Generalmente existirán:

- El **administrador de la base de datos** o Database Administrator (DBA), que será la persona o conjunto de ellas encargadas de la función de administración de la base de datos. Tiene el control centralizado de la base de datos y es el responsable de su buen funcionamiento. Es el encargado de autorizar el acceso a la base de datos, de coordinar y vigilar su utilización y de adquirir los recursos software y hardware que sean necesarios.
 - Los **usuarios de la base de datos**, que serán diferentes usuarios de la BD con diferentes necesidades sobre los datos, así como diferentes accesos y privilegios. Podemos establecer la siguiente clasificación:
 - Diseñadores.
 - Operadores y personal de mantenimiento.
 - Analistas y programadores de aplicaciones.
 - Usuarios finales: ocasionales, simples, avanzados y autónomos.
5. **Herramientas de la base de datos.** Son un conjunto de aplicaciones que permiten a los administradores la gestión de la base de datos, de los usuarios y permisos, generadores de formularios, informes, interfaces gráficas, generadores de aplicaciones, etc.

7.3. Arquitectura

Un SGBD cuenta con una arquitectura a través de la que se simplifica a los diferentes usuarios de la base de datos su labor. El objetivo fundamental es separar los programas de aplicación de la base de datos física.

Encontrar un estándar para esta arquitectura no es una tarea sencilla, aunque los tres estándares que más importancia han cobrado en el campo de las bases de datos son **ANSI/SPARC/X3**, **CODASYL** y **ODMG** (*éste sólo para las bases de datos orientadas a objetos*). Tanto ANSI (EEUU), como ISO (Resto del mundo), son el referente en cuanto a estandarización de bases de datos, conformando un único modelo de bases de datos.

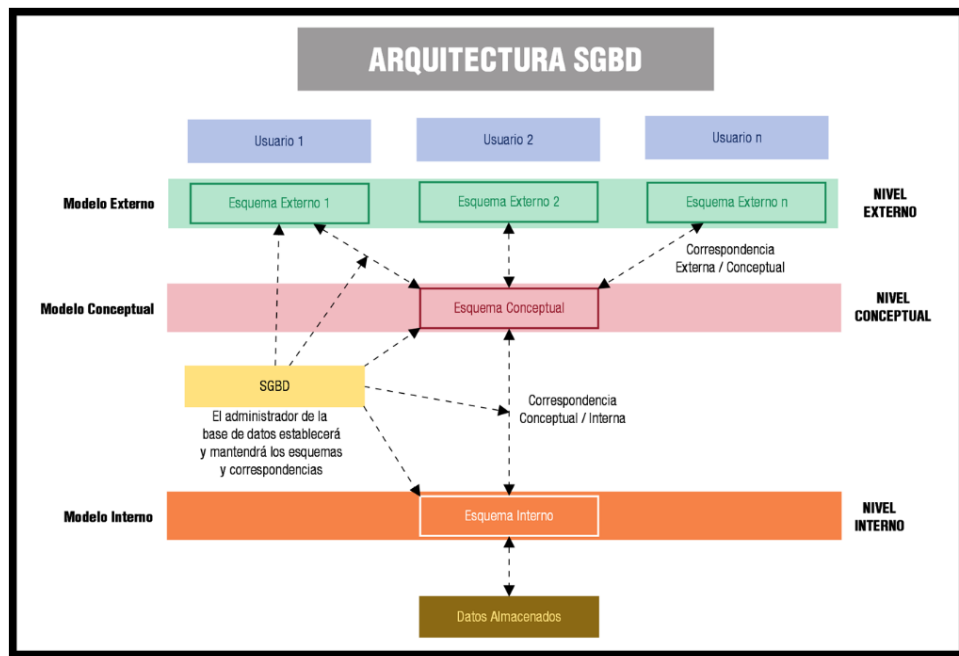
La arquitectura propuesta proporciona tres niveles de abstracción: **nivel interno o físico**, **nivel lógico o conceptual** y **nivel externo o de visión del usuario**. A continuación se detallan las características de cada uno de ellos:

- **Nivel interno o físico:** En este nivel se describe la estructura física de la base de datos a través de un esquema interno encargado de detallar el sistema de almacenamiento de la base de datos y sus métodos de acceso. Es el nivel más cercano al almacenamiento físico. A través del esquema físico se indican, entre otros, los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, las unidades de almacenamiento, etc.
- **Nivel lógico o conceptual:** En este nivel se describe la estructura completa de la base de datos a través de un esquema que detalla las entidades, atributos, relaciones, operaciones de los usuarios y restricciones. Los detalles relacionados con las estructuras de almacenamiento se ocultan, permitiendo realizar una abstracción a más alto nivel.
- **Nivel externo o de visión del usuario:** En este nivel se describen las diferentes vistas que los usuarios percibirán de la base de datos. Cada tipo de usuario o grupo de ellos verá sólo la parte de la base de datos que le interesa, ocultando el resto.

Para una base de datos, sólo existirá un **único esquema interno**, un **único esquema conceptual** y podrán existir **varios esquemas externos** definidos para uno o varios usuarios.

Gracias a esta arquitectura se consigue la **independencia de datos** a dos niveles:

- **Independencia lógica:** Podemos modificar el esquema conceptual sin alterar los esquemas externos ni los programas de aplicación.
- **Independencia física:** Podemos modificar el esquema interno sin necesidad de modificar el conceptual o el externo. Es decir, se puede cambiar el sistema de almacenamiento, reorganizar los ficheros, añadir nuevos, etc., sin que esto afecte al resto de esquemas.



7.4. Tipos de SGBD

¿Qué tipos de SGBD existen? Para responder a esta pregunta podemos realizar la siguiente clasificación, atendiendo a diferentes criterios:

1. El primer criterio que se suele utilizar es **por el modelo lógico en que se basan**. Actualmente, el modelo lógico que más se utiliza es el **relacional**. Los modelos en red y jerárquico han quedado obsoletos. Otro de los modelos que más extensión está teniendo es el modelo **orientado a objetos**. Por tanto, en esta primera clasificación tendremos:
 - Modelo Jerárquico.
 - Modelo de Red.
 - Modelo Relacional.
 - Modelo Orientado a Objetos.

(Para recordar los modelos de bases de datos vistos, sitúate en el epígrafe 4 de esta Unidad de Trabajo y analiza su contenido.)

2. El segundo criterio de clasificación se centra en el **número de usuarios** a los que da servicio el sistema:
 - **Monousuario:** sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales.
 - **Multiusuario:** entre los que se encuentran la mayor parte de los SGBD, atienden a varios usuarios al mismo tiempo.
3. El tercer criterio se basa en el **número de sitios en los que está distribuida la base de datos:**
 - **Centralizados:** sus datos se almacenan en un solo computador. Los SGBD centralizados pueden atender a varios usuarios, pero el SGBD y la base de datos en sí residen por completo en una sola máquina.
 - ****Distribuidos (Homogéneos, Heterogéneos):**** la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red. Los sistemas homogéneos utilizan el mismo SGBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a varias bases de datos autónomas preexistentes almacenadas en sistemas distribuidos heterogéneos. Esto da lugar a los SGBD federados o sistemas multibase de datos en los que los SGBD participantes tienen cierto grado de autonomía local.
4. **El cuarto criterio toma como referencia el coste.** Existiendo paquetes gratuitos y otros con elevado coste como Oracle. (**Precios de Oracle**)
5. El quinto, y último, criterio establece su clasificación **según el propósito:**
 - **Propósito General:** pueden ser utilizados para el tratamiento de cualquier tipo de base de datos y aplicación.
 - **Propósito Específico:** Cuando el rendimiento es fundamental, se puede diseñar y construir un software de propósito especial para una aplicación específica, y este sistema no sirve para otras aplicaciones. Muchos sistemas de reservas de líneas aéreas son de propósito especial y pertenecen a la categoría de **sistemas de procesamiento de transacciones en línea**, que deben atender un gran número de transacciones concurrentes sin imponer excesivos retrasos.

8. Sistemas Comerciales vs Sistemas Libres

SGBD	Descripción	URL
ORACLE	Reconocido como uno de los mejores a nivel mundial. Es multiplataforma, fiable y seguro. Es Cliente/Servidor. Basado en el modelo de datos Relacional. De gran potencia, aunque con un precio elevado hace que sólo se vea en empresas muy grandes y multinacionales. Ofrece una versión gratuita Oracle Database 11g Express Edition .	Oracle
MYSQL	Sistema muy extendido que se ofrece bajo dos tipos de licencia, comercial o libre. Para aquellas empresas que deseen incorporarlo en productos privativos, deben comprar una licencia específica. Es Relacional, Multihilo, Multiusuario y Multiplataforma. Su gran velocidad lo hace ideal para consulta de bases de datos y plataformas web.	MySQL
DB2	Multiplataforma, el motor de base de datos relacional integra XML de manera nativa, lo que IBM ha llamado pureXML, que permite almacenar documentos completos para realizar operaciones y búsquedas de manera jerárquica dentro de éste, e integrarlo con búsquedas relacionales.	DB2
INFORMIX	Otra opción de IBM para el mundo empresarial que necesita un DBMS sencillo y confiable. Es un gestor de base de datos relacional basado en SQL. Multiplataforma. Consume menos recursos que Oracle, con utilidades muy avanzadas respecto a conectividad y funciones relacionadas con tecnologías de Internet/Intranet, XML, etc.	Informix
Microsoft SQL SERVER	Sistema Gestor de Base de Datos producido por Microsoft. Es relacional, sólo funciona bajo Microsoft Windows, utiliza arquitectura Cliente/Servidor. Constituye la alternativa a otros potentes SGBD como son Oracle, PostgreSQL o MySQL.	Microsoft SQL Server

Otros SGBD comerciales importantes son: DBASE, ACCESS e INTERBASE.

SGBD	Descripción	URL
MySQL	Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Distribuido bajo dos tipos de licencias, comercial y libre. Multiplataforma, posee varios motores de almacenamiento, accesible a través de múltiples lenguajes de programación y muy ligado a aplicaciones web.	MySQL
PostgreSQL	Sistema Relacional Orientado a Objetos. Considerado como la base de datos de código abierto más avanzada del mundo. Desarrollado por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Es multiplataforma y accesible desde múltiples lenguajes de programación.	PostgreSQL
Firebird	Sistema Gestor de Base de Datos relacional, multiplataforma, con bajo consumo de recursos, excelente gestión de la concurrencia, alto rendimiento y potente soporte para diferentes lenguajes.	Firebird
Apache Derby	Sistema Gestor escrito en Java, de reducido tamaño, con soporte multilinguaje, multiplataforma, altamente portable, puede funcionar embebido o en modo cliente/servidor.	Apache Derby
SQLite	Sistema relacional, basado en una biblioteca escrita en C que interactúa directamente con los programas, reduce los tiempos de acceso siendo más rápido que MySQL o PostgreSQL, es multiplataforma y con soporte para varios lenguajes de programación.	SQLite

9. Bases de datos centralizadas

Si nos preguntamos cómo es la arquitectura de un sistema de base de datos, hemos de saber que todo depende del sistema informático que la sustenta. Tradicionalmente, la arquitectura centralizada fue la que se utilizó inicialmente, aunque hoy en día es de las menos utilizadas.

Sistema de base de datos centralizado: Es aquella estructura en la que el SGBD está implantado en una sola plataforma u ordenador desde donde se gestiona directamente, de modo centralizado, la totalidad de los recursos. Es la arquitectura de los centros de proceso de datos tradicionales. Se basa en tecnologías sencillas, muy experimentadas y de gran robustez.

Los sistemas de los años sesenta y setenta eran totalmente centralizados, como corresponde a los sistemas operativos de aquellos años, y al hardware para el que estaban hechos: un gran ordenador para toda la empresa y una red de terminales sin inteligencia ni memoria.

Las principales características de las bases de datos centralizadas son:

- Se almacena completamente en una ubicación central, es decir, todos los componentes del sistema residen en un solo computador o sitio.
- No posee múltiples elementos de procesamiento ni mecanismos de intercomunicación como las bases de datos distribuidas.
- Los componentes de las bases de datos centralizadas son: los datos, el software de gestión de bases de datos y los dispositivos de almacenamiento secundario asociados.
- Son sistemas en los que su seguridad puede verse comprometida más fácilmente.

Ventajas e inconvenientes de las bases de datos centralizadas.

En la siguiente tabla se representan las ventajas e inconvenientes destacables de esta arquitectura de bases de datos.

Ventajas	Inconvenientes
Se evita la redundancia debido a la posibilidad de inconsistencias y al desperdicio de espacio.	Un mainframe en comparación de un sistema distribuido no tiene mayor poder de cómputo.
Se evita la inconsistencia. Ya que si un hecho específico se representa por una sola entrada, la no-concordancia de datos no puede ocurrir.	Cuando un sistema de bases de datos centralizado falla, se pierde toda disponibilidad de procesamiento y sobre todo de información confiada al sistema.
La seguridad se centraliza.	En caso de un desastre o catástrofe, la recuperación es difícil de sincronizar.
Puede conservarse la integridad.	Las cargas de trabajo no se pueden difundir entre varias computadoras, ya que los trabajos siempre se ejecutarán en la misma máquina.
El procesamiento de los datos ofrece un mejor rendimiento.	Los departamentos de sistemas retienen el control de toda la organización.
Mantenimiento más barato. Mejor uso de los recursos y menores recursos humanos.	Los sistemas centralizados requieren un mantenimiento central de datos.

10. Bases de datos distribuídas

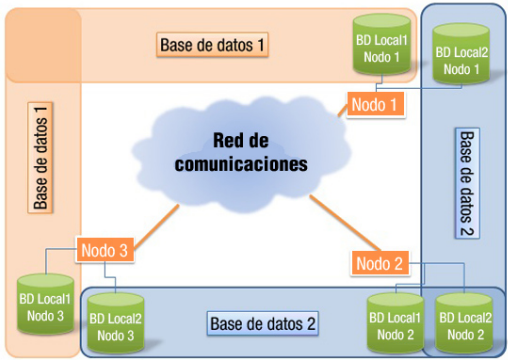
La necesidad de integrar información de varias fuentes y la evolución de las tecnologías de comunicaciones, han producido cambios muy importantes en los sistemas de bases de datos. La respuesta a estas nuevas necesidades y evoluciones se materializa en los sistemas de bases de datos distribuidas.

Base de datos distribuida (BDD): es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran distribuidas entre diferentes nodos interconectados por una red de comunicaciones.

Sistema de bases de datos distribuida (SBDD): es un sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones, de tal forma que, un usuario en cualquier sitio puede acceder los datos en cualquier parte de la red exactamente como si los datos estuvieran almacenados en su sitio propio.

Sistema gestor de bases de datos distribuida (SGBDD): es aquel que se encarga del manejo de la BDD y proporciona un mecanismo de acceso que hace que la distribución sea transparente a los usuarios. El término transparente significa que la aplicación trabajaría, desde un punto de vista lógico, como si un solo SGBD ejecutado en una sola máquina, administrara esos datos.

BASES DE DATOS RELACIONALES DISTRIBUIDAS, HOMOGÉNEAS Y ALTAMENTE INTEGRADAS



Un SGBDD desarrollará su trabajo a través de un conjunto de sitios o nodos, que poseen un sistema de procesamiento de datos completo con una base de datos local, un sistema de gestor de bases de datos e interconectados entre sí. Si estos nodos están dispersos geográficamente se internocerarán a través de una red de área amplia o WAN, pero si se encuentran en edificios relativamente cercanos, pueden estar interconectados por una red de área local o LAN. Este tipo de sistemas es utilizado en: organizaciones con estructura descentralizada, industrias de manufactura con múltiples sedes (automoción), aplicaciones militares, líneas aéreas, cadenas hoteleras, servicios bancarios, etc.

En la siguiente tabla se representan las ventajas e inconvenientes destacables de las BDD:

Ventajas e inconvenientes de las bases de datos distribuidas.

Ventajas	Inconvenientes
El acceso y procesamiento de los datos es más rápido ya que varios nodos comparten carga de trabajo.	La probabilidad de violaciones de seguridad es creciente si no se toman las precauciones debidas.
Desde una ubicación puede accederse a información alojada en diferentes lugares.	Existe una complejidad añadida que es necesaria para garantizar la coordinación apropiada entre los nodos.
Los costes son inferiores a los de las bases centralizadas.	La inversión inicial es menor, pero el mantenimiento y control puede resultar costoso.
Existe cierta tolerancia a fallos. Mediante la replicación, si un nodo deja de funcionar el sistema completo no deja de funcionar.	Dado que los datos pueden estar replicados, el control de concurrencia y los mecanismos de recuperación son mucho más complejos que en un sistema centralizado.
El enfoque distribuido de las bases de datos se adapta más naturalmente a la estructura de las organizaciones. Permiten la incorporación de nodos de forma flexible y fácil.	El intercambio de mensajes y el cómputo adicional necesario para conseguir la coordinación entre los distintos nodos constituyen una forma de sobrecarga que no surge en los sistemas centralizados.
Aunque los nodos están interconectados, tienen independencia local.	Dada la complejidad del procesamiento entre nodos es difícil asegurar la corrección de los algoritmos, el funcionamiento correcto durante un fallo o la recuperación.

10.1. Fragmentación

Sabemos que en los sistemas de bases de datos distribuidas la información se encuentra repartida en varios lugares. La forma de extraer los datos consultados puede realizarse mediante la fragmentación de distintas tablas pertenecientes a distintas bases de datos que se encuentran en diferentes servidores. El problema de fragmentación se refiere al particionamiento de la información para distribuir cada parte a los diferentes sitios de la red.

Pero hay que tener en cuenta el **grado de fragmentación** que se aplicará, ya que éste es un factor determinante a la hora de la ejecución de consultas. Si no existe fragmentación, se tomarán las relaciones o tablas como la unidad de fragmentación. Pero también puede fragmentarse a nivel de tupla (fila o registro) o a nivel de atributo (columna o campo) de una tabla. No será adecuado un grado de fragmentación nulo, ni tampoco un grado de fragmentación demasiado alto. El grado de fragmentación deberá estar equilibrado y dependerá de las

particularidades de las aplicaciones que utilicen dicha base de datos. Concretando, el objetivo de la fragmentación es encontrar un nivel de particionamiento adecuado en el rango que va desde tuplas o atributos hasta relaciones completas.

Tema extra

11.- Primeros pasos en Oracle Database 18c Express Edition (XE)

¿Qué es Oracle Database 18c Express Edition**?***

Es la última versión del SGBD Oracle Express Edition (XE) con todas la potencia de Oracle. Gratuita, fácil de descargar y utilizar, recomendado para instituciones educativas.

Como indica Oracle en su página oficial:

Tanto si es desarrollador, administrador de base de datos, científico de datos o educador, como si sencillamente tiene curiosidad por las bases de datos, Oracle Database 18c Express Edition (XE) es ideal para comenzar. Es la misma potente Oracle Database en la que confían las empresas de todo el mundo, empaquetada para una descarga simple, fácil de usar y una experiencia con todas las funciones. Dispondrá de una Oracle Database para usar en cualquier entorno, además de la capacidad de incrustar y redistribuir, todo gratuitamente.

Oracle

¿Y si tengo instalada una versión distinta de Oracle Database?

Si tienes una versión anterior y te manejas bien con ella no es necesario para el desarrollo del módulo que instales la versión 18c XE. Pero si decides hacerlo tendrás que desinstalar previamente la anterior.

¿Por dónde empezamos?

El primer paso que debemos dar es descargar el software necesario desde la página oficial de Oracle. A través del siguiente enlace podrás acceder a la zona de descarga de Oracle Database 18c Express Edition, regístrate, escoge el que se ajuste a tus necesidades según tu sistema operativo y descárgalo en tu ordenador. Si no tienes soltura con Linux te recomiendo que instales la versión para windows [OracleXE184_Win64.zip](#)

[Zona de descarga de Oracle Database 18c Express Edition.](#)

¿Cómo se realiza la instalación?

Para llevar a cabo la instalación del software descargado, dependiendo de tu sistema operativo, debes registrarte y seguir los pasos desde el siguiente enlace de la página oficial de Oracle (está en inglés).

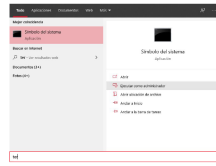
[Pasos para la instalación de Oracle Database 18c XE](#)

o desde este PDF traducido al español [Instalacion de Oracle Database 18c XE.pdf](#) (pdf - 269 KB)

En este enlace [PDF_Acceso](#) (pdf - 92 KB) tienes los pasos para acceder al usuario administrador desde **SQLPLUS**. [visualizar el usuario](#) y el contenedor activo, y [crear un usuario](#).

Acesso al usuario administrador desde SQLPLUS para crear un usuario.

1.- Abrimos la terminal del símbolo de sistema (ventana de comandos de windows) ejecutando como administrador.



2.- Nos conectamos en SQLPLUS como administradores con la sentencia
sqlplus sys as sysdba
(tendremos que indicar la contraseña que pusimos en la instalación)



3.- Al conectarnos con sys nos conectamos a un contenedor de base de datos (CDB) llamada ROOT.
Para mostrar el usuario y la base de datos actual utilizamos el comando show
show user con_name

