

BASES DE DATOS



1 Introducción

Una base de datos es una colección o conjunto de datos almacenados, relacionados de forma lógica entre sí y organizados con cierta estructura.

El término base de datos se suele abreviar como BD, en algunas ocasiones también BBDD, siguiendo las normas de los plurales en español, y DB que proviene del inglés *Data Base*.

Ejemplos de usos frecuentes de BBDD:

Administrativas: toda empresa necesita almacenar información sobre sus clientes, pedidos, facturas, productos...

Contables: para gestionar pagos, declaraciones de hacienda...

Motores de búsquedas: contienen información sobre todos los documentos de Internet. P.e: Google o Altavista

Científicas: recolección de datos climáticos y medioambientales, químicos, geológicos, etc. para simulaciones y predicciones.

Configuraciones: como el registro de Windows o de un portal Joomla!

Bibliotecas: contienen información bibliográfica.

Virus: los antivirus guardan información sobre software potencialmente malicioso.

Otros: militares, videojuegos, deportes, censos, etc.

Según la organización que tengan los datos distinguimos diferentes modelos de bases de datos: el jerárquico, en red, el relacional,... El modelo de bases de datos más extendido es el **relacional** y es el que trataremos en este módulo.

Es importante diferenciar entre el concepto de **Base de Datos** y el de **Sistema Gestor de Bases de Datos**, ya que es habitual confundirlos y sin embargo son cosas muy distintas.

El Sistema Gestor de Bases de Datos (SGBD o DBMS en inglés *Data Base Management System*) es el conjunto de programas o aplicación que nos permite definir, crear y mantener bases de datos, proporcionando un acceso controlado a las mismas. Es una herramienta que sirve de *interfaz* entre el usuario y las bases de datos.



2 Sistemas de Información Orientados al Proceso: Archivos

Anteriormente a los SGBD (años setenta) la información necesaria para la informatización de algún tipo de proceso se trataba y se gestionaba utilizando sistemas de gestión basados en archivos.

DEF: Un **archivo** (también denominad *fichero*) de datos es conjunto de datos que tratan sobre un mismo tema y que está almacenado en algún tipo de soporte fijo (el disco duro de un ordenador, por ejemplo).

DEF: Un **registro** es una estructura de datos homogéneos referentes a una misma entidad (por ejemplo una persona).

DEF: Un **campo** es cada una de las unidades de información de un registro

La definición de los datos se encuentra codificada dentro de los programas de aplicación y el control del acceso y la manipulación de los datos se hacen desde los mismos programas de aplicación.

Si tenemos una aplicación para una tarea determinada, esta está formada por un conjunto de programas donde cada uno realiza una serie de operaciones. Al utilizar un sistema de gestión basado en archivos:

- Cada programa debe contener la estructura y la definición completa de los datos del archivo
- Si la estructura de algún dato de un archivo cambia, hay que modificarlo en cada programa.
- Lo mismo si necesitamos añadir o quitar algún campo del archivo, independientemente de que el programa en cuestión vaya a utilizar o no este campo.

INCONVENIENTES

- Redundancia de datos: Aparecen datos repetidos pues es difícil que una aplicación utilice datos de otra aplicación, los mismos datos están en varios archivos. Por ejemplo, los datos de un alumno en el archivo de matrícula, en el de la biblioteca del centro, en el de calificaciones, etc.

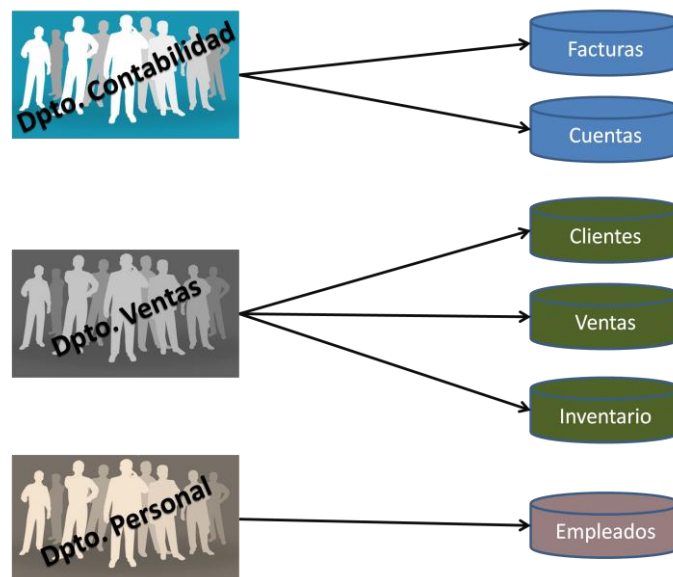
La redundancia causa un aumento del tiempo de acceso, una aumento de los costes de almacenamiento y lo peor es motivo de la *inconsistencia de los datos* pues el mismo dato puede contener información distinta en distintos archivos.

- Inconsistencia de datos: cuando los datos almacenados son incorrectos en algún archivo. Si hay redundancia de datos es casi seguro que llegará a haber inconsistencia de datos puesto que cada copia puede ir cambiando en el tiempo. Un ejemplo muy habitual es el cambio de domicilio de una persona. Si se modifica en un archivo y en otro no, se estará utilizando en muchas ocasiones una información incorrecta.

- Dependencia física de los datos pues el lugar en dónde están almacenados los archivos, también se encuentra codificada en los programas de aplicación. Así si lo cambiamos de carpeta o de unidad de disco los programas dejarán de poder acceder a él. Será necesario identificar, modificar y probar todos los programas que manipulen ese archivo.

- Dependencia lógica de los datos: pues la estructura de los archivos está codificada en los programas y un cambio en ella nos obliga a identificar, modificar y probar todos los programas que manipulen el archivo modificado, incluso aunque no usen el campo afectado.
- Separación y aislamiento de los datos pues al estar repartidos en varios archivos, y tener diferentes formatos, es difícil escribir nuevos programas que aseguren la manipulación de los datos correctos pues ¿cuál es el archivo correcto?
- Proliferación de programas lo que dificulta el acceso a los datos. Cada vez que se necesite una consulta no prevista en el diseño original, hay que desarrollar un nuevo programa, con lo que el acceso a los datos se hace, pues, de forma no eficiente.
- Dependencia de la estructura del archivo respecto al lenguaje de programación. Si necesitamos hacer un programa en un lenguaje de programación distinto al que lo creó, es muy probable que el formato del archivo sea incompatible con este nuevo lenguaje.
- Dificultad para el acceso concurrente o simultáneo de varios usuarios. En un sistema de gestión de archivos pueden dar por resultado datos inconsistentes.
- Problemas en la seguridad de los datos pues el control de accesos y permisos a distintos tipos de usuarios resulta muy complejo.

Sistema de Gestión de Datos basado en Archivos



Todos estos problemas en el tratamiento de grandes volúmenes de información llevan a la aparición de las **Bases de Datos** y de los **Sistemas Gestores de Bases de Datos**, surgidos de la idea de separar los datos de los programas que los manipulan pudiendo así modificarse la estructura de los datos de los archivos sin tener que modificar los programas con los que trabajan. Había que estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan.

TIPOS DE ARCHIVOS

Según su contenido:

- Fichero de **texto** (o fichero plano o ascii): el sistema operativo traduce el contenido del fichero a caracteres alfabéticos y números que entiende el ser humano. P.e: documento de Microsoft Word.
- Fichero **binario**: almacena estructuras de datos complejas. P.e: sonido, vídeo, etc.

Según su utilidad:

- Ficheros **maestros**: almacenan información principal. P.e: clientes.
- Ficheros de **movimiento**: almacenan variaciones de un fichero maestro. P.e: bajas de clientes
- Ficheros **históricos**: almacenan datos que muestran los cambios experimentados por un dato. Por ejemplo las compras de un cliente durante el año.

Según su organización interna: la organización de un fichero dicta la forma en la que se puede acceder a sus datos.

- **Secuencial**: se debe recorrer el fichero desde principio a fin hasta localizar el dato buscado. Similar a la selección de imágenes en una cinta de vídeo.
- **Directa**: permiten acceder a un dato concreto sin tener que visitar los anteriores. Similar a la selección de canciones en un CD.
- **Indexada**: contiene un índice que indica en qué lugar del fichero se encuentra cada dato. Similar a la consulta del índice de un libro.

3 Sistemas de Información Orientados a los Datos: las Bases de Datos

DEF: Una Base de Datos es una *“Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más, de la mejor manera posible. Los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraerlos datos almacenados”* (Martin, 1975).

Las Bases de Datos presentan algunos inconvenientes:

- Instalación costosa.
- Necesidad de personal especializado.
- Implantación larga y difícil.
- Falta de rentabilidad a corto plazo.
- Excesiva dependencia del fabricante.

Evolución de las BBDD

- Década 1950: Sistemas basados en ficheros

- Se inventan las cintas magnéticas que sólo podían leerse secuencialmente y almacenaban ficheros secuenciales.

- Década 1960: BBDD jerárquicas y en red

- Se generaliza el uso de discos magnéticos que permiten el acceso directo a cualquier parte de un fichero.
- Las BBDD jerárquicas y en red estructuran la información como listas enlazadas y árboles.

- Década 1970: BBDD relacionales

- Edgar Frank Codd publica el artículo “Un modelo relacional de datos para grandes bancos de datos compartidos” que supone la definición del modelo relacional de datos basado en la lógica de predicados y la teoría de conjuntos.
- Hoy en día siguen siendo las BBDD más utilizadas.

- Década 1980: SQL (Structured Query Language)

- IBM lanza su motor de bases de datos DB2.
- Años más tarde crea un potente lenguaje de consultas para manipular información de BBDD relacionales.

- Mediados 1990: BBDD paralelas

- IBM lanza una versión de DB2 que reparte una BD enorme entre varios servidores comunicados por líneas de alta velocidad.

- Finales 1990: BBDD orientadas a objetos

- IBM y Oracle incorporan a sus BBDD la posibilidad de manipular objetos (datos + comportamiento)

- Aparición de Internet: BBDD distribuidas

- La cantidad de información crece a gran velocidad.
- Las BBDD distribuidas multiplican el nº de ordenadores que controlan la BD. Necesitan intercambiar información y actualizaciones de la BD a través de la red.
- Aparece SW de ayuda a la decisión y las BBDD multidimensionales.

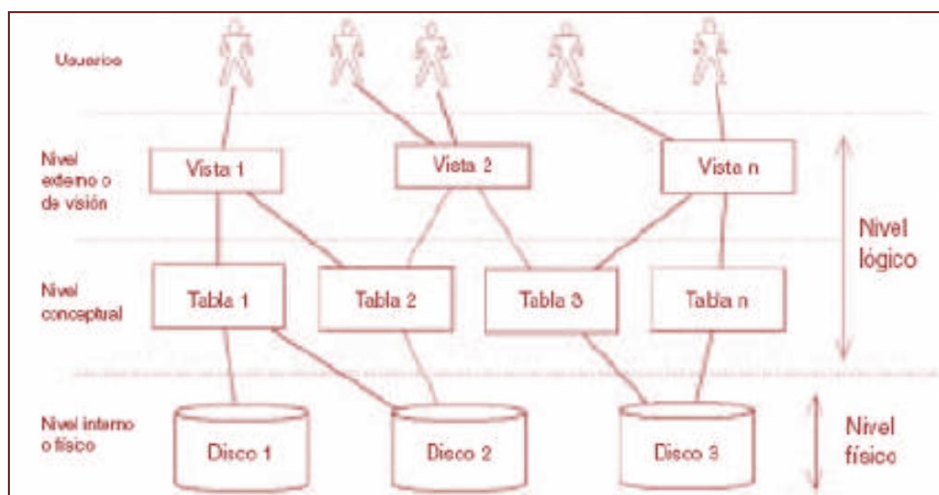
En 1975, el comité de estandarización ANSI-SPARC (*American National Standard Institute – Standards Planning and Requirements Committee*) propuso la arquitectura de tres niveles para los SGBD cuyo objetivo principal era el de separar los programas de aplicación de la BD física. En esta arquitectura el esquema de una BD se define en tres niveles de abstracción distintos:

Nivel interno o físico: Es la representación de cómo la información es almacenada en los dispositivos de almacenamiento. Describe las estructuras, dispositivos, volúmenes, ficheros, organización del archivo, los métodos de acceso a los registros, los tipos de registros, la cantidad de almacenamiento a reservar, etc.

Nivel conceptual: Es la visión o representación del problema tal y como éste se presenta en el mundo real. En esta observación, en el análisis del problema, se determinan los objetos o entidades que intervienen en el mismo, las propiedades o características de estas entidades y las relaciones o dependencias que existen entre ellos.

Se ocultan los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones.

Nivel externo o de visión: Es la visión de los datos que tienen los usuarios finales. En este nivel se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un usuario o grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos.



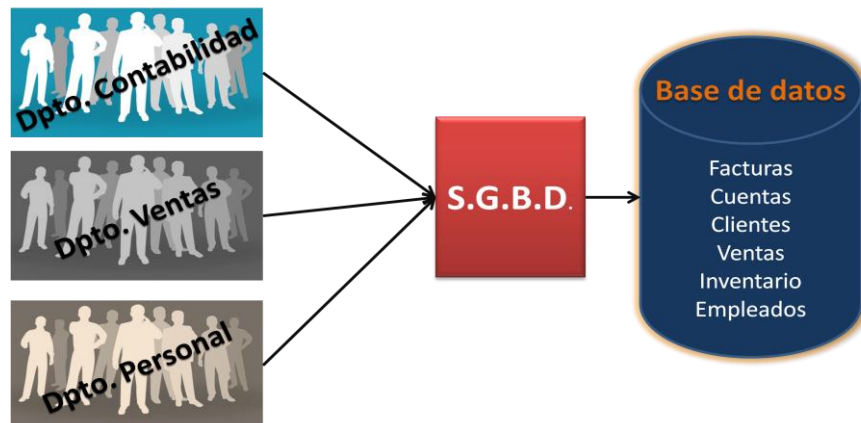
En realidad los únicos datos que existen están a nivel físico almacenados en discos u otros dispositivos. Los SGBD basados en esta arquitectura permiten que cada grupo de usuarios haga referencia a su propio esquema externo.

Un ejemplo: Un usuario del Dpto. de Ventas puede ver las fechas con formato *dd-mm-yy* que es más adecuado para la relación con los clientes. Sin embargo, en el Dpto. de Contabilidad se visualizan las fechas con formato *yyyy-mm-dd* que es un formato de fecha internacional.

4 Sistemas Gestores de Bases de Datos

DEF: un Sistema Gestor de Bases de Datos o SGBD es una aplicación o conjunto de programas que permite la gestión de Bases de Datos. Suministra a usuarios, a analistas, a programadores y al administrador de la BD, los medios necesarios para describir, recuperar y manipular los datos, manteniendo su integridad, confidencialidad y seguridad.

Sistema de Gestión de Bases de Datos



4.1 Funciones

- Definición de los datos a los distintos niveles de abstracción (físico, lógico y externo)
- Manipulación de los datos en la base de datos. Es decir, la inserción, modificación, borrado y acceso o consulta a los mismos.
- Mantenimiento de la integridad de la base de datos. Integridad en cuanto a los datos en sí, sus valores y las relaciones entre ellos. Para ello hay que garantizar que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna.

DEF: Una **transacción** es un conjunto de acciones que cambian el contenido de la Base de Datos.

Por ejemplo

Una transacción en el sistema de información informatizado de una empresa inmobiliaria podría ser dar de alta a un empleado o eliminar un inmueble. Una transacción un poco más complicada sería eliminar un empleado y reasignar sus inmuebles a otro empleado. En este caso hay que realizar varios cambios sobre la BD. Si la transacción fallara durante su realización, por ejemplo porque fallase el hardware, la BD quedaría en un estado inconsistente. Algunos de los cambios se habrían hecho y otros no. Por lo tanto, los cambios realizados deberían ser deshechos para devolver la BD a un estado consistente.

- Control de la privacidad y seguridad de los datos en la base de datos.
- Debe garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales. Debe controlar que sólo los usuarios autorizados accedan a la base de datos.
- Los SGBD ofrecen mecanismos para implantar restricciones de integridad en la base de datos. Estas restricciones van a proteger la base de datos contra daños accidentales. Los valores de los datos que se almacenan deben satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad, que especificará el administrador y los diseñadores de la base de

- datos. El sistema gestor de base de datos puede determinar si se produce una violación de la restricción.
- Debe asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la base de datos de forma concurrente.
- Proporciona herramientas y mecanismos para la planificación y realización de copias de seguridad y restauración para restablecer la información en caso de fallos en el sistema.
- Y, en definitiva, los medios necesarios para el establecimiento de todas aquellas características exigibles a una base de datos como herramientas para la importación/exportación de datos o herramientas de desarrollo.

4.2 Componentes.

- **Diccionario De Datos**

También denominado catálogo o repositorio. Contiene información que describe los datos y objetos de la base de datos y sus propiedades. Toda esta información que se almacena recibe el nombre genérico de *metadatos*, ya que se trata de datos que describen otros datos. Normalmente un diccionario de datos contiene, entre otras cosas:

- Información sobre la estructura lógica y física de la BD.
- Información sobre el espacio asignado y utilizado por los objetos. Nombre, tipo y tamaño de los datos.
- Relaciones entre los datos.
- Restricciones de integridad sobre los datos.
- Usuarios autorizados a acceder a los objetos de la base de datos.
- Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.

- **Lenguajes**

Los lenguajes de un SGBD se clasifican en:

- Lenguaje de Definición de Datos o LDD (**DDL**: Data Definition Language). Es el lenguaje para definir y describir los objetos de la BD, su estructura, relaciones y restricciones.
- Lenguaje de Control de Datos o LCD (**DCL**: Data Control Language) para el control y seguridad de los datos (privilegios, modos de acceso, etc.)
- Lenguaje de Manipulación de Datos o LMD (**DML**: Data Management Language): se utiliza para actualizar (insertar, modificar y borrar) y consultar (leer) los datos de la base de datos.

Dependiendo del modelo de datos y del SGBD los lenguajes pueden ser:

***Procedimentales** (o Procedurales) que describen paso a paso el conjunto de instrucciones que deben ejecutarse para solucionar un problema.

***No Procedimentales** (o Declarativos) describen el problema que se quiere solucionar, pero no las instrucciones necesarias para solucionarlo.

Con los lenguajes declarativos se dice “qué” se quiere obtener, no “cómo”. Por ejemplo, *obtener los nombres de todos los empleados que tengan más de 30 años*.

Las BD relacionales utilizan lenguajes declarativos como SQL (*Structured Query Language*) o QBE (*Query By Example*).

4.3 Usuarios y permisos.

Por lo general hay cuatro tipos de usuarios en un SGBD: Administrador, Diseñador, Programador, Usuario

-Administrador del SGBD

Es el encargado del diseño físico de la base de datos y de su implementación (llevarla a la práctica), realiza el control de la seguridad y de la concurrencia, mantiene el sistema para que siempre se encuentre operativo y se encarga de que los usuarios y las aplicaciones obtengan buenas prestaciones.

Realiza por tanto tareas como:

- Instalar el sistema gestor de base de datos en el sistema informático.
- Crear las bases de datos que se vayan a gestionar.
- Crear y mantener el esquema de la base de datos.
- Crear y mantener las cuentas de usuario de la base de datos.
- Arrancar y parar el sistema gestor de base de datos, y cargar las bases de datos con las que se ha de trabajar.
- Colaborar con el administrador del sistema operativo en las tareas de ubicación, dimensionado y control de los archivos y espacios de disco ocupados por el sistema gestor de base de datos.
- Colaborar en las tareas de formación de usuarios.
- Establecer estándares de uso, políticas de acceso y protocolos de trabajo diario para los usuarios de la base de datos.
- Suministrar la información necesaria sobre la base de datos a los equipos de análisis y programación de aplicaciones.
- Efectuar tareas de explotación como:
 - o Vigilar el trabajo diario colaborando en la información y resolución de dudas de los usuarios de la base de datos.
 - o Controlar en tiempo real los accesos, tasas de uso, anomalías...
 - o Si llega el caso, reorganizar la base de datos.
 - o Efectuar las copias de seguridad periódicas de la base de datos.
 - o Restaurar la base de datos después de un incidente material a partir de las copias de seguridad...
 - o Ajustar y optimizar la base de datos mediante el ajuste de sus parámetros, y con ayuda de las herramientas de monitorización y de las estadísticas del sistema.

El administrador debe conocer muy bien el SGBD que se esté utilizando, así como el equipo informático sobre el que esté funcionando. Tiene una gran responsabilidad ya que posee el máximo nivel de privilegios. Hay que procurar que en la administración de una base de datos haya el menor número de administradores posible.

-Diseñador del SGBD

Realiza el diseño lógico de la BD, debiendo identificar los datos, las relaciones entre ellos y sus restricciones.

La persona que diseña la BD debe tener un profundo conocimiento de los datos de la empresa y también debe conocer sus reglas de negocio. Las reglas de negocio describen las características principales de los datos tal y como los ve la empresa. Para obtener un buen resultado, el diseñador de la BD debe implicar en el desarrollo del modelo de datos a todos los usuarios de la BD, tan pronto como sea posible.

El diseño lógico de la BD es independiente del SGBD concreto que se vaya a utilizar, es independiente de los programas de aplicación, de los lenguajes de programación y de cualquier otra consideración física.

-Programador del SGBD

Se encargan de implementar los programas de aplicación que servirán a los usuarios finales. Estos programas de aplicación son los que permiten consultar datos, insertarlos, modificarlos y eliminarlos.

-Usuarios del SGBD

Interactúan con la base de datos, por lo general a través de aplicaciones e interfaces. Son clientes de la BD que hacen uso de ella sin conocer en absoluto su funcionamiento y organización.

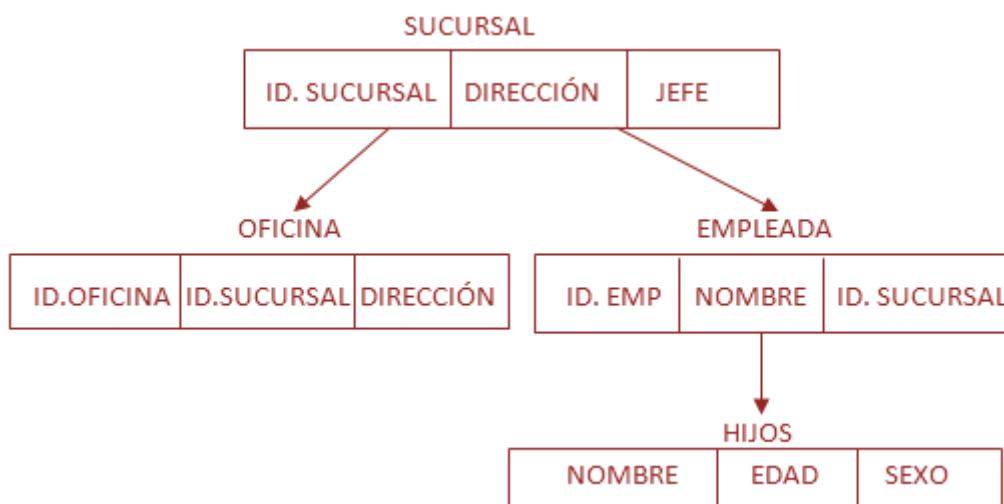
4.4 Tipos de SGBD:

- **SEGÚN EL MODELO LÓGICO**

Modelo Jerárquico.- Fue utilizado por los primeros SGBD. Se le llama también modelo en árbol debido a que utiliza una estructura en árbol para organizar los datos.

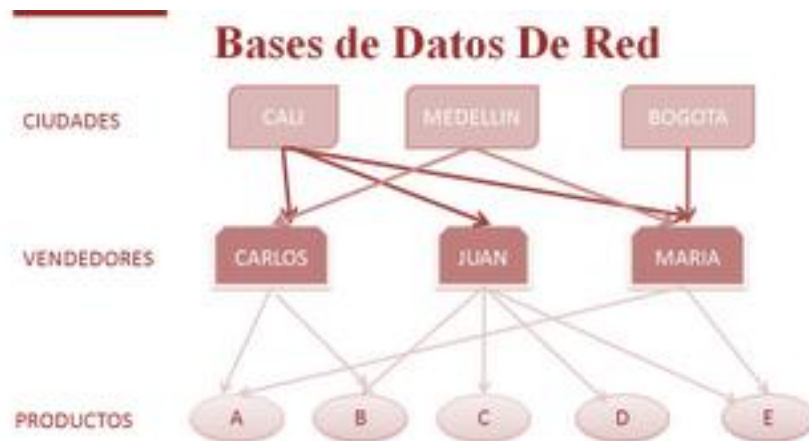
La información se organiza con una relación entre las entidades de este modelo de tipo jerárquico, es decir, siempre de tipo padre/hijo. De esta forma hay una serie de nodos que contendrán atributos y que se relacionarán con nodos hijos de forma que puede haber más de un hijo para el mismo padre pero un hijo sólo puede tener un padre.

Este esquema está en desuso ya que no es válido para modelar la mayoría de problemas de bases de datos.



Modelo de Red.- Es un modelo que ha tenido una gran aceptación aunque apenas se utiliza actualmente. 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 cada hijo puede tener más de un padre.

En este modelo se pueden representar perfectamente cualquier tipo de relación entre los datos, pero hace muy complicado su manejo.



Modelo Relacional.- En este modelo los datos se organizan en tablas cuyos datos se relacionan. Es el modelo más popular y que estudiaremos más ampliamente.

Nombre	Calle	Ciudad	Número
Lowery	Mapple	Queens	900
Shiver	North	Bronx	556
Shiver	North	Bronx	647
Hodges	Sidehill	Brooklyn	801
Hodges	Sidehill	Brooklyn	647

Número	Saldo
900	55
556	100.000
647	105.366
801	10.533

Modelo orientado a 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. La programación orientada a objetos permite cohesionar datos y procedimientos, haciendo que se diseñen estructuras que poseen datos (atributos) en las que se definen los procedimientos (operaciones) que pueden realizar con los datos. En las bases orientadas a objetos se utiliza esta misma idea.

A través de este concepto se intenta que estas bases de datos consigan solucionar las limitaciones de las relacionales. Por ejemplo el problema de la herencia (el hecho de que no se puedan realizar relaciones de herencia entre las tablas), tipos definidos por el usuario, disparadores (triggers) almacenables en la base de datos, soporte multimedia...

Se supone que son las bases de datos de tercera generación (la primera fue las bases de datos jerárquicas y en red y la segunda las relacionales), lo que significa que el futuro parece estar a favor de estas bases de datos. Pero siguen sin

reemplazar a las relacionales, aunque son el tipo de base de datos que más está creciendo en los últimos años.



Modelo Objeto-Relacional.- Es una extensión de la base de datos relacional tradicional, a la cual se le proporcionan características de la programación orientada a objetos (POO). El problema de las bases de datos orientadas a objetos es que requieren reinvertir capital y esfuerzos de nuevo para convertir las bases de datos relacionales en bases de datos orientadas a objetos. En las bases de datos objeto-relacional se intenta conseguir una compatibilidad relacional dando la posibilidad de integrar mejoras de la orientación a objetos.

Estas bases de datos se basan en el estándar SQL 99. En ese estándar se añade a las bases relacionales la posibilidad de almacenar procedimientos de usuario, disparadores (triggers), tipos definidos por el usuario, consultas recursivas, ...

Las últimas versiones de la mayoría de las clásicas grandes bases de datos relacionales (Oracle, SQL Server, Informix, ...) son objeto-relacional.

- **SEGÚN EL NÚMERO DE USUARIOS**

Monousuario sólo un usuario puede acceder simultáneamente a los datos

Multiusuario varios usuarios pueden acceder concurrentemente a los datos

- **SEGÚN LA FORMA DE ALMACENAR LOS DATOS**

Centralizadas un solo servidor almacena los datos

Distribuidas múltiples servidores de almacenamiento

- **SEGÚN EL ÁMBITO DE APLICACIÓN**

De Propósito **General**

De Propósito **Específico**.

- **SEGÚN SU COSTE**

Comerciales: son de pago como Oracle, SQLServer

Libres: gratuitos como Postgree, MySQL

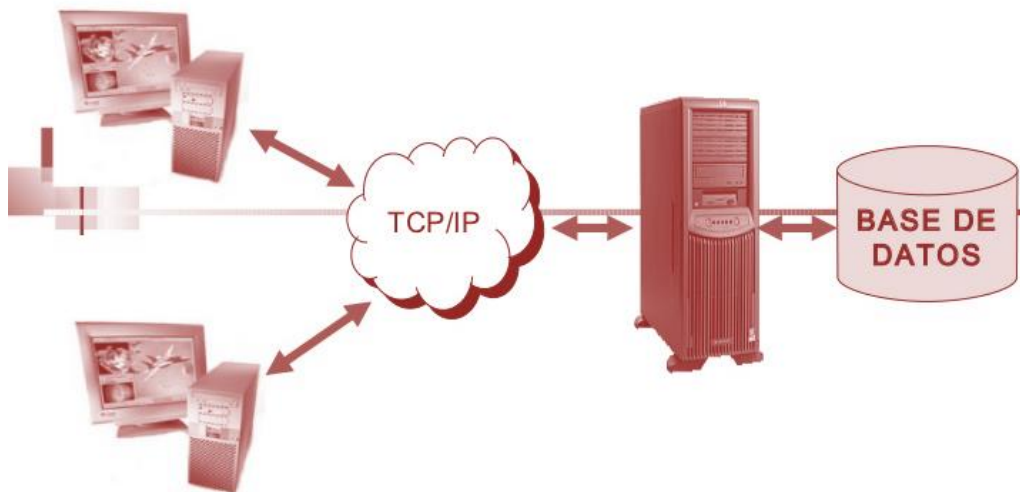
5 Arquitectura Cliente/Servidor

La Arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

Las Arquitecturas múltiples clientes / múltiples servidores son más flexibles, ya que la base de datos es distribuida en varios servidores. Cada cliente tiene un servidor directo al cual hace sus peticiones. La comunicación entre los servidores ejecuta las transacciones y peticiones de los usuarios y esta es transparente para ellos.

En la Arquitecturas de igual a igual un SGBD verdadero no distingue entre cliente y servidor.

Idealmente cada máquina puede desempeñar la funcionalidad de cliente y de servidor. La base de datos es físicamente distribuida en diferentes lugares fragmentado y replicando los datos. La fragmentación es deseable ya que hace posible el poner los datos cerca de los usuarios que necesitan estos datos, de esta forma se reduce potencialmente el costo de la transmisión y se reduce el tamaño de las relaciones involucradas en las consultas de los usuarios.



Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

6 Sistemas gestores de bases de datos comerciales y libres

Los SGBD son fundamentales en las empresas por eso se debe elegir el que mejor se adecue a sus necesidades.

-COMERCIALES

Ventajas

- Es el SGBD relacional más usado a nivel mundial
- Puede ejecutarse en todas las plataformas
- Dispone de un lenguaje de diseño muy completo: el PL/SQL
- Permite el uso de particiones que mejoran la eficiencia y la replicación (duplicación de las bases de datos de un servidor –maestro- en otro –esclavo-). También admite la administración de bases de datos distribuidas.
- Es la más orientada a Internet

Desventajas

- Elevado coste de adquisición
- Elevado coste de formación
- Necesidad de ajustes

The Oracle logo, consisting of the word "ORACLE" in a bold, red, sans-serif font.

Ventajas

- Seguridad al permitir administrar permisos a todo: permisos a nivel de servidor, seguridad en tablas, permitir o no lectura, escritura y ejecución,...
- Utiliza una extensión llamada Transact SQL que soporta la modificación y eliminación de bases de datos, tablas, atributos, índices, etc.
- Es capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea
- Soporta procedimientos almacenados
- Un buen entorno gráfico

Desventajas

- Utiliza una gran cantidad de memoria RAM para la instalación y utilización del software
- La relación calidad-precio está por debajo de ORACLE

The Microsoft SQL Server logo, featuring a stylized red and grey tower icon to the left of the text "Microsoft SQL Server" in a black, sans-serif font.

**Ventajas**

- Sencillo, bueno para principiantes con asistentes muy potentes

Desventajas

- Se necesita tener siempre instalado Access para que funcione
- No es recomendable para bses de datos de gran calibre

-LIBRES**Ventajas**

- Código abierto y gratuito
- Alta concurrencia, pues permite que mientras un proceso escribe en una tabla, otros puedan acceder a la misma sin necesidad de bloqueo.
- Amplia variedad de tipos de datos nativos
- Herencia de tablas
- Integridad transaccional

Desventajas

- Consume más recursos por lo que requiere mayores características de hardware
- Fácil de vulnerar sin la protección adecuada
- Sentencias poco intuitivas

Ventajas

- Multihilo y multiusuario
- Multiplataforma
- Soporte para subconsultas, procedimientos almacenados y control de transacciones
- Rápido, fiable y fácil de usar

**Desventajas**

- Los privilegios de una tabla no se eliminan automáticamente al borrarla.
- El soporte para disparadores es básico, por lo que hay ciertas limitaciones en qué se puede hacer con ellos
- No tiene un panel de control gráfico y detallado. No es intuitivo

Índice de contenidos

1 Introducción	2
2 Sistemas de Información Orientados al Proceso: Archivos	3
3 Sistemas de Información Orientados a los Datos: las Bases de Datos	5
4 Sistemas Gestores de Bases de Datos	7
4.1 Funciones	8
4.2 Componentes.	9
4.3 Usuarios y permisos.	10
4.4 Tipos de SGBD:	11
5 Arquitectura Cliente/Servidor	14
6 Sistemas gestores de bases de datos comerciales y libres	15