

UT_1 ALMACENAMIENTO DE LA INFORMACIÓN

1.1 FICHEROS

1.2 BASES DE DATOS

1.3 SISTEMAS GESTORES DE BASES DE DATOS

1.4 SISTEMAS GESTORES DE BBDD COMERCIALES Y LIBRES

1.5 BD CENTRALIZADAS Y BD DISTRIBUIDAS

1.1 FICHEROS

En la década de los setenta, los procesos básicos que se llevaban a cabo en una empresa se centraban en cuestiones relacionadas con contabilidad y facturación. Las necesidades de almacenamiento y gestión de información podían satisfacerse utilizando un número relativamente reducido de archivos en papel agrupados y ordenados, los típicos ficheros clásicos.

Al realizar la informatización de la información, se pasó de tener los datos en papel a poder acceder a ellos de manera mucho más rápida a través del ordenador. La informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba manualmente, por eso se sigue hablando de ficheros, formularios, carpetas, directorios,...

La información ha de ser trasladada del papel a formato digital, se guarda para su posterior recuperación, consulta y procesamiento. El elemento que permitió almacenar los datos de manera permanente en dispositivos de memoria masiva fue el fichero o archivo.

Fichero o archivo: conjunto de información relacionada, tratada como un todo y organizada de forma estructurada. Los ficheros están formados por **registros lógicos** que contienen datos relativos a un mismo elemento u objeto. A su vez, los registros están divididos en **campos** que contienen cada una de las informaciones elementales que forman un registro.

Los datos se guardarán en el fichero de forma que se puedan añadir, suprimir, actualizar o consultar en cualquier momento.

La información que contiene un fichero se llevará desde el dispositivo de almacenamiento externo a la memoria para ser tratada. Se conoce como **registro físico** o bloque a la cantidad de información que es transferida entre el soporte y la memoria principal en una sola operación de lectura/escritura.

Normalmente se transfieren varios registros del fichero en cada operación de I/e, por tanto, un bloque puede contener varios registros lógicos y al número de registros que entran en un bloque se conoce como **factor de bloqueo**.

Tipos de ficheros

La información guardada en los ficheros es una ristra de bits (ceros y unos) y es necesaria su **interpretación** para dar sentido a su contenido. Por ejemplo, para almacenar una imagen en un ordenador se puede utilizar un fichero binario bip que almacena en un vector de datos con los colores que tiene cada pixel que tiene la imagen. Las dimensiones de dicha imagen y su paleta de colores también hay que guardarla en el fichero.

Todos estos datos se guardan siguiendo un formato y el sistema operativo o la utilidad que trate los gráficos, debe conocer este formato para poder extraer los pixel y mostrarlos en la pantalla en la forma y dimensiones correctas. ¿Qué sucede si se abre el gráfico con una utilidad como el bloc de notas? El resultado será ilegible e incomprensible.

Tradicionalmente los ficheros se clasifican atendiendo a varios criterios:

- Según su contenido: ficheros de texto o binarios
- Según su organización: secuencial, directa o indexada
- Según su utilidad: maestros, históricos y de movimientos.
- Según el periodo de permanencia: permanentes y temporales.

El **contenido** de un fichero puede ser tratado como texto o como datos binarios, es decir, los bits almacenados en un fichero pueden ser traducidos por el S. O. a caracteres alfabéticos y números que entiende el ser humano, o pueden ser tratados como componentes de estructuras de datos más complejas, como ficheros que guardan vídeo, sonido, imágenes, etc.

La **organización** de un fichero indica la forma en que se accederá a sus datos. Se pueden clasificar en:

- fichero secuencial, su contenido está dispuesto siguiendo una secuencia ordenada, unos datos a continuación de otros. Para acceder a un dato concreto, será necesario pasar por todos los anteriores.
- fichero directo, se puede acceder a un dato sin pasar por los anteriores.
- fichero indexado, se accede a los datos consultando un índice, es decir, una estructura de datos que permite acceder a la información rápidamente simulando el índice de un libro.

La **utilidad** de un fichero será el uso que se va a hacer de él. Se dividen en:

- fichero maestro si contiene información relevante para la organización (los clientes de una empresa).
- Fichero de movimientos, guarda las operaciones que se quieren hacer sobre los datos, altas, bajas y modificaciones y será necesario enfrentarlo con el fichero maestro para actualizar éste.
- Fichero histórico, guarda datos que no son necesarios para el proceso diario

El **periodo de permanencia** será el tiempo que la información se guarda en el fichero. Se clasifican en:

- Permanentes, que contienen información relevante para una aplicación y a su vez se subdividen en:
 - o maestros,
 - o históricos, contienen datos que antes fueron actuales(antiguos clientes de la empresa) y
 - o constantes, contienen datos fijos para una aplicación que no suelen ser modificados (códigos postales).
- Temporales, que guardan información útil para parte de la aplicación no para toda ella y son generados a partir de permanentes. Su tiempo de existencia es corto y se subdividen en:
 - o intermedios, almacenan resultados de una aplicación que serán utilizados por otra
 - o de maniobra, almacenan datos de una aplicación que no pueden ser mantenidos en memoria principal por falta de espacio
 - o ficheros de resultados, guardan datos que van a ser transferidos a un dispositivo de salida.

Desde el punto de vista del S.O. los ficheros se tratan según su contenido (texto o datos binarios) y según su tipo (imágenes, ejecutables, video, etc.)

Ficheros de texto

También se conocen como ficheros planos o ASCII (American Standard Code for Information Interchange). El código ASCII es un estándar que asigna un valor numérico a cada carácter, así se pueden representar los documentos llamados de Texto Plano, es decir, los que son directamente legibles por las personas. En este código se utiliza un byte para representar cada carácter ($2^8=256$ combinaciones, es decir, 256 caracteres numerados de 0 a 255). La extensión del fichero de texto servirá para identificar el tipo de información que contiene, por ejemplo:

- ficheros de configuración de S.O. o de una aplicación, extensiones .ini, .inf, .conf.
- ficheros de código fuente, contienen programas informáticos, .sql, .java, .c
- ficheros de páginas web que son interpretados por el navegador, .html, .css, .xml
- formatos enriquecidos, textos que contienen códigos de control para mostrar un texto más elegante, .rtf, .ps, .tex.

Ficheros binarios

Estos ficheros necesitan un formato para ser interpretados. Algunos tipos de formatos son:

- de imagen: .jpg, .gif, .bmp, .png, etc.
- de video: .mov, .avi, .mpg
- comprimidos o empaquetados: .zip, .Z, .tar
- ejecutables o compilados: .exe, .com, .cgi
- procesadores de texto: .doc, .odt

1.2 BASES DE DATOS

Los ficheros permiten organizar y memorizar conjuntos de datos del mismo tipo con una determinada estructura. Cuando se diseñan las aplicaciones, si éstas dependen directamente de los ficheros, se pierde independencia y surgen problemas como información duplicada, incoherencia de datos, fallos de seguridad, etc.

Para solucionar estos problemas aparece el concepto de **base de datos**. Una BD permite reunir toda la información relacionada en un único sistema de almacenamiento y cualquier aplicación podrá utilizarla de manera independiente.

La gestión de BD ha experimentado gran cantidad de cambios. Con la llegada de internet, el número de usuarios de BD creció exponencialmente, y aunque muchos de ellos no sean conscientes de ello, el acceso a dichas BD forma parte de la vida cotidiana.

Conceptos

Una BD consta de los siguientes elementos:

Dato: es el registro sobre conceptos, transacciones o sucesos. La información implica que los datos estén procesados para que resulten útiles para el receptor. Procesar un dato supone situarlo en un contexto determinado. Es decir, 1996 es un dato, si indicamos que se trata de la fecha de nacimiento de una persona ya estaremos hablando de información.

Tipo de dato: indica la naturaleza del dato. Podemos tener datos numéricos que son aquellos con los que se pueden realizar operaciones aritméticas y los datos alfanuméricos que contienen caracteres alfabéticos y dígitos numéricos. Estos datos simples se pueden combinar para formar tipos de datos complejos. Por ejemplo, podemos definir el tipo de dato Fecha que a su vez estaría formado por tres datos numéricos: día, mes y año.

Entidad, objeto real o abstracto con características diferenciadoras de otros, del que se almacena información. Supongamos una BD de una clínica veterinaria, posibles entidades pueden ser: mascota, veterinario, consulta, etc.

Atributos: datos que se almacenan de la entidad. Cualquier propiedad o característica de una entidad puede ser un atributo. En el caso de la entidad Mascota pueden 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 el ejemplo, serían los datos de una mascota concreta: 123456, Sultán, podenco, gris, 23/03/2011.

Campo: es un identificador para toda una familia de datos. Cada campo pertenece a un tipo de datos. Por ejemplo, el campo “FechaNacimiento” representa las fechas de nacimiento para las personas y pertenece al tipo de datos Fecha.

Campo clave: es un campo que identifica de forma única a cada registro. Ejemplo, el NIF es único por cada persona, será por tanto el campo clave de los que contiene el registro persona.

Base de datos: 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 representan entidades y sus interrelaciones, almacenados con la mínima redundancia y posibilitando el acceso a ellos eficientemente por parte de varias aplicaciones y usuarios.

La BD no solo contiene datos de la organización, también almacena una descripción de dichos datos, a esta descripción se conoce como metadatos y se almacena en el diccionario de datos o catálogo y es lo que permite que exista independencia de datos lógica-física.

Uso de las bases de datos

Dentro de una organización existen distintas personas que pueden utilizar una base de datos y con distinto propósito. Estas personas son:

- El **administrador**, es quien crea e implementa físicamente la BD. Toma las decisiones relacionadas con el almacenamiento de la información. También establece las políticas de seguridad y de acceso para garantizar el menor número de problemas.
- Los **diseñadores**, diseñan cómo será la BD. Llevan a cabo la identificación de los datos, las relaciones entre ellos, sus restricciones, etc. Deben conocer a fondo los datos y las reglas de negocio en la que

la empresa se mueve. Para obtener buen resultado debe implicar en el proceso a todos los usuarios de la BD.

- Los **programadores de aplicaciones**, una vez diseñada y construida la BD, éstos se encargan de implementar los programas de aplicación que servirán a los usuarios finales. Estos programas ofrecerán la posibilidad de realizar consultas de datos, inserción, actualización o eliminación de los mismos.
- Los **usuarios finales**, son los clientes finales de la BD. Al diseñar, implementar y mantener la BD se busca cumplir los requisitos establecidos por el cliente para la gestión de la información.

Ejemplos de uso de las BD:

- Banca: información de clientes, cuentas, transacciones, préstamos, etc.
- Líneas aéreas, clientes, horarios, vuelos, destinos, etc.
- Universidades: estudiantes, carreras, horarios, profesores, etc.
- Telecomunicaciones: registro de llamadas realizadas, generación de facturas mensuales, etc.
- Medicina: información hospitalaria, pacientes, médicos, historiales clínicos, etc.
- Justicia: casos, delincuentes, sentencias, investigaciones, etc.
- Organismos públicos: datos ciudadanos, certificados, etc.
- Sistemas de posicionamiento geográfico
- Hostelería y turismo: hoteles, reservas, vuelos, excursiones, etc.
- Ocio digital: apuestas, juegos on line, etc.
- Cultura: gestión de bibliotecas, museos, etc.

Ubicación de la información

¿Dónde se encuentra realmente almacenada la información? Las bases de datos pueden tener tamaño muy reducido o bien, ser muy voluminosas y complejas (del orden de Terabytes). Normalmente se almacenan en discos duros y otros dispositivos de almacenamiento externo a los que se accede desde un ordenador. Una gran BD puede necesitar servidores en lugares diferentes, y viceversa, pequeñas bases de datos pueden existir como archivos en el disco duro de un solo equipo.

Los sistemas de almacenamiento de información más utilizados para el despliegue de bases de datos son los siguientes:

- Discos SATA, es una interfaz de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento como pueden ser disco duro, lectores y grabadores de CD/DVD. La velocidad de transferencia puede ser: 150, 300 o 600 Megabytes.
- Discos SCSI, son interfaces preparadas para discos duros de gran capacidad de almacenamiento y velocidad de rotación.
- RAID, es una matriz de discos independientes. Se basa en el montaje en conjunto de dos o más discos duros formando un bloque de trabajo. Se permite la ampliación de la velocidad y mejoras en la velocidad y seguridad de almacenamiento.
- Sistemas NAS, sistema de almacenamiento masivo en red que permite compartir la capacidad de almacenamiento de un ordenador servidor con ordenadores personales o servidores clientes a través de una red.
- Sistemas SAN, red de área de almacenamiento. Se trata de una red concebida para conectar servidores y matrices de discos. Los recursos estarán disponibles para varios servidores en una red.

Modelos de bases de datos

Tradicionalmente se establecen tres modelos de BD: jerárquico, en red y relacional. De ellos el más extendido es el relacional aunque actualmente existen dos variantes que son las que más se están utilizando, el modelo de bases de datos distribuidas y orientadas a objetos.

Modelo jerárquico

Empezó a utilizarse en los años setenta cuando IBM creó su Sistema Administrador de Información (IMS) y sentó las bases para que la mayoría de sistemas de gestión de la información lo utilizaran. También se conoce como modelo en árbol al utilizar una estructura en árbol invertido para la organización de los datos.

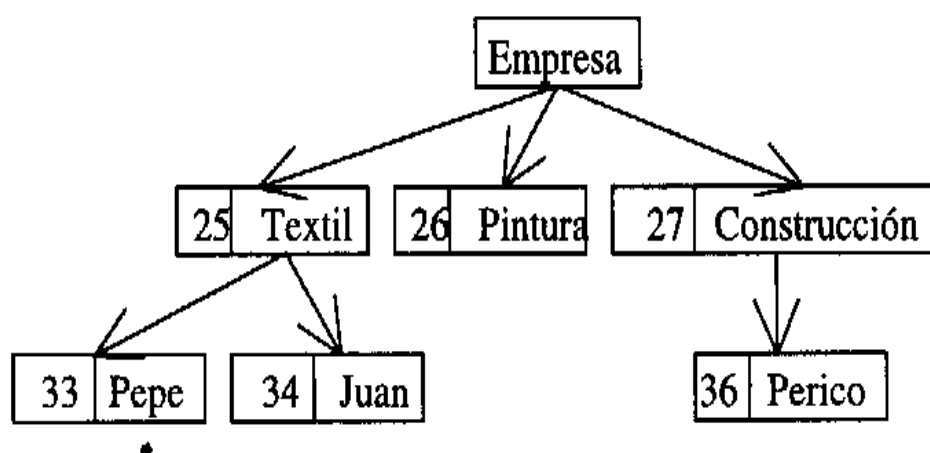
La información se organiza con una jerarquía en la que la relación entre las entidades siempre es del tipo padre/hijo. Existen nodos que contienen atributos o campos y que se relacionan con sus nodos hijos, pudiendo cada uno tener más de un hijo, pero un nodo siempre tendrá un nodo padre. El nivel más alto se denomina raíz.

Los datos se almacenan en estructuras lógicas llamadas segmentos que se relacionan entre sí utilizando arcos. Visualmente es un árbol invertido con los nodos padre en la parte superior y los hijos en la inferior.

Actualmente este modelo está en desuso debido a sus limitaciones, por ejemplo:

- Sólo permite relaciones 1:N
- Un segmento hijo no puede tener más de un padre
- No se permite más de una relación entre dos segmentos.
- Para acceder a cualquier nodo es necesario siempre comenzar por la raíz.
- El recorrido del árbol ha de hacerse en un orden determinado.

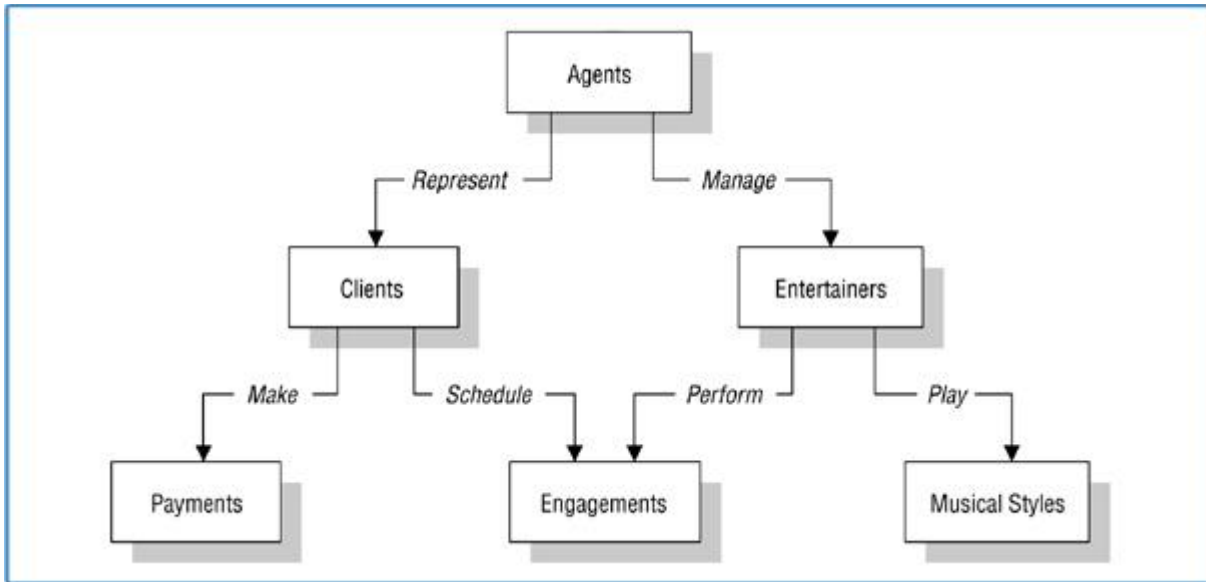
Ejemplo:



Modelo en red

Aparece este modelo a mediados de los sesenta como respuesta a las limitaciones del modelo jerárquico.

La información también se organiza en registros o nodos y enlaces. En los registros se almacenan los datos y los enlaces permiten relacionar dichos datos. Es un modelo muy parecido al jerárquico con la diferencia que un nodo puede tener más de un padre, por tanto se pueden representar relaciones N:M.



(Los contratos relacionan los clientes y los artistas)

Modelo relacional

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

En este modelo la BD se representa 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. Las tablas son bidimensionales y se conocen como relaciones (de ahí el nombre del modelo) y permiten representar tanto los datos como sus relaciones. Cada tabla tiene un nombre único y está dividida en filas y columnas.

Cada fila de la tabla se conoce como entidad, registro o tupla y cada columna como campo o atributo. Al conjunto de valores que puede tomar un atributo se le llama dominio.

Cada tabla tendrá un atributo llamado clave que identificara de forma unívoca cada fila, por tanto, nunca hay dos tuplas iguales.

Las tablas deben cumplir una serie de requisitos:

- Todos los registros son del mismo tipo
- 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.

Ejemplo de tablas:

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsain
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

Tabla *cliente*

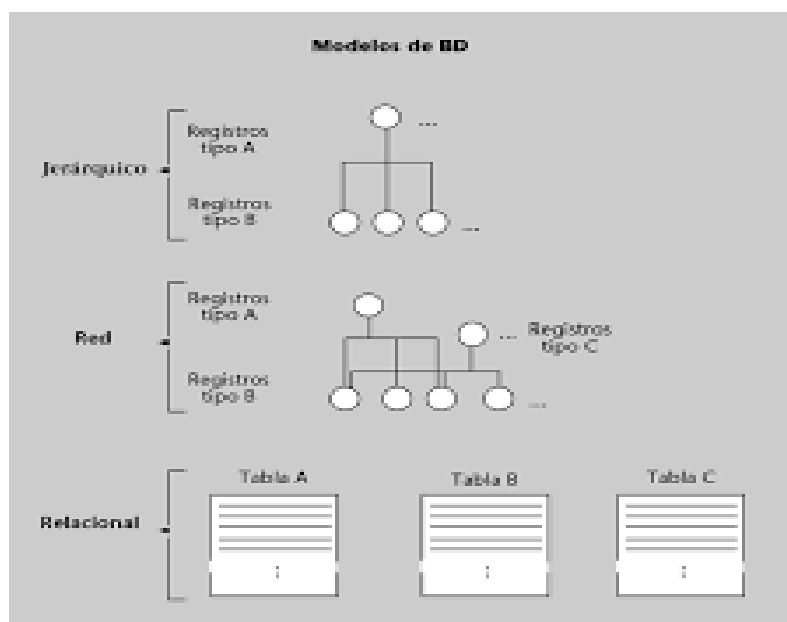
<i>número-cuenta</i>	<i>saldo</i>	<i>id-cliente</i>	<i>número-cuenta</i>
C-101	500	19.283.746	C-101
C-215	700	19.283.746	C-201
C-102	400	01.928.374	C-215
C-305	350	67.789.901	C-102
C-201	900	18.273.609	C-305
C-217	750	32.112.312	C-217
C-222	700	33.666.999	C-222
		01.928.374	C-201

Tabla *cuenta*

Tabla de relación *cliente-cuenta*

El lenguaje habitual para realizar consulta sobre una BD relacional es SQL, Structured Query Language, o lenguaje estructurado de consultas.

Un resumen de los tres modelos anteriores:



Modelo orientado a objetos

Este modelo define una BD en término 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 llamados métodos. Algunos sistemas basados en el modelo relacional han evolucionado e incorporado conceptos orientados a objetos. A estos modelos se les conoce como sistemas objeto-relacionales.

El modelo orientado a objetos es cubrir las limitaciones del modelo relacional e incorporan mejoras como la herencia entre tablas, los tipos definidos por el usuario, disparadores almacenados en la BD (triggers), etc.

Los conceptos de la POO (Programación orientada a objetos) que incorpora el modelo son:

- Encapsulación, propiedad que permite ocultar la información al resto de los objetos y así impedir accesos incorrectos.
- Herencia, propiedad mediante la que 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.

Este modelo es considerado como la tercera generación de las bases de datos siendo el modelo en red la primera y el modelo relacional la segunda.

Modelo Objeto-Relacional

Las BD pertenecientes a este modelo son un híbrido entre las del modelo relacional y el orientado a objetos. En ellas se busca obtener lo mejor del modelo relacional, incorporando las mejoras que ofrece la orientación a objetos. Se siguen almacenando tuplas, pero su estructura puede reflejar herencia directa, es decir, las relaciones pueden ser definidas en función de otras.

La mayoría de las BD relacionales clásicas de gran tamaño (Oracle, SQL Server) son objeto-relacionales e incorporan funciones que tengan código en algún lenguaje de programación como Java, C, etc.

1.3 SISTEMAS GESTORES DE BASES DE DATOS

Un Sistema Gestor de Base de Datos (SGBD) es un 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 BD manteniendo su integridad, confidencialidad y seguridad.

El SGBD debe permitir al usuario la definición, la creación y la manipulación de la BD.

Las ventajas del uso SGBD son:

- Proporcionan al usuario una visión abstracta de los datos ocultando parte de la complejidad de cómo se almacenan y mantienen los datos.

- Ofrecen independencia física, es decir, la visión y manipulación de los datos es independiente de cómo están almacenados físicamente.
- Disminuyen la redundancia de los datos.
- Aseguran la integridad de los datos.
- Facilitan el acceso a los datos, aportando rapidez y evitando la pérdida de 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 del sistema.

Funciones de un SGBD

Las funciones fundamentales del SGBD son:

1. **Función de descripción o definición**, permite al diseñador de la BD crear las estructuras de datos asociadas a los tres niveles que forman la arquitectura de una BD: *interno*, *conceptual* y *externo*. Para ello utiliza un lenguaje de descripción de datos o **DDL** que le permite definir las estructuras de datos, las relaciones entre ellos o las restricciones que deben cumplir. Se especifican las características de los datos a cada uno de los tres niveles:
 - A nivel interno, indicar el espacio en disco reservado para la BD, la longitud de los campos, etc.
 - A nivel conceptual, se proporcionan herramientas para la definición de entidades y su identificación, sus atributos, sus relaciones, las restricciones de integridad.
 - A nivel externo, se definen las vistas de los distintos usuarios
2. **Función de manipulación**, permite a los usuarios buscar, añadir, suprimir o modificar los datos de la BD de acuerdo con las especificaciones y normas de seguridad dictadas por el administrador. Para ello se utiliza un lenguaje de manipulación de datos o **DML**. También permite definir vistas externas comunes a todos los usuarios o vistas parciales para cada uno de ellos.
3. **Función de control**, permite al administrador establecer mecanismos de protección de las diferentes visiones de los datos asociadas a cada usuario y proporciona elementos de creación y modificación de dichos usuarios. También incorpora sistemas para la creación de copias de ficheros, carga de ficheros, auditoría, protección de ataques, etc. El lenguaje utilizado para estas operaciones es el lenguaje de control de datos o **DCL**.

El lenguaje SQL proporciona sentencias para realizar operaciones DDL, DML y DCL. Además los SGBD suelen proporcionar otras herramientas que complementan a estos lenguajes como generadores de formularios, informes, interfaces gráficas, etc.

Componentes del SGBD

Un SGBD es un paquete de software complejo que debe proporcionar servicios relacionados con el almacenamiento y la explotación de los datos de forma eficiente. Para ello cuenta con los siguientes componentes:

1. **Lenguajes de la BD**, así tendremos DDL, DML y DCL que permitirán especificar los datos, su estructura, sus relaciones, las reglas de integridad, control de acceso, características físicas y vistas externas de los usuarios.
2. **El diccionario de datos (DD)** que contiene la descripción de los datos almacenados, aquí se guarda información sobre todos los datos que forman la BD, por ejemplo:
 - Estructura lógica y física de la BD.
 - Definición de tablas, vistas, índices, disparadores, procedimientos, 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.
 - Auditoria de acceso a los datos, utilización, etc.
3. **El gestor de la BD**. Es un intermediario entre el usuario y los datos. Es la parte de software que proporciona una interfaz entre los datos almacenados y las aplicaciones que los manejan. Además es el encargado de garantizar la privacidad, seguridad e integridad de los datos, controlando los accesos concurrentes e interactuando con el S.O.
4. **Usuarios de la BD**. En el sistema existen distintos perfiles de usuario:
 - Administrador de la BD o DBA, tiene el control y es el responsable del buen funcionamiento de la BD. Da permisos de acceso, coordina y vigila su utilización y se encarga de adquirir los recursos software y hardware necesarios.
 - Usuarios de la BD, cada uno tendrá unas necesidades concretas sobre los datos, así como diferentes accesos y privilegios. Aquí se distinguen: diseñadores, operadores y personal de mantenimiento, analistas y programadores de aplicaciones, usuarios finales: ocasionales, simples, avanzados y autónomos.
5. **Herramientas de la BD**, conjunto de aplicaciones que permiten a los administradores la gestión de la BD, de los usuarios y permisos, generadores de formularios, informes, interfaces gráficas, generadores de aplicaciones, etc.

Arquitectura del SGBD

Todo SGBD cuenta con una arquitectura que permite separar los programas de aplicación de la BD física. Los estándares para esta arquitectura más importantes son: ANSI/SPARC/X3, CODASYL y OMDG (sólo para BD orientadas a objetos). En 1975 el comité ANSI/SPARC propuso una arquitectura compuesta por los siguientes niveles de abstracción:

- **Nivel interno o físico**, es el nivel más bajo de abstracción y se describe la estructura física de la BD mediante un esquema interno o físico encargado de detallar el almacenamiento físico de la BD. En este esquema se indica por ejemplo los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, sus campos, etc.
- **Nivel lógico o conceptual**, aquí se describe mediante el esquema conceptual la estructura completa de la BD (todos los datos que van a intervenir en el sistema) a través de un esquema conceptual que detalla las entidades, atributos, relaciones, operaciones de los usuarios y restricciones. Se obtiene a partir de los requerimientos de los usuarios potenciales del sistema a implantar independientemente de la organización o dónde se guardarán físicamente. Contiene los datos elementales, los compuestos, las reglas que rigen el funcionamiento de la empresa, etc.

- **Nivel externo o de visión del usuario**, se describen las diferentes vistas que los usuarios percibirán de la BD. Cada tipo de usuario verá sólo la parte de la BD que le interesa y el resto quedará oculto. Es el conjunto de visiones individuales de la BD y cada una se denomina subesquema o vista y son independientes del esquema conceptual. Un subesquema puede ser compartido por varios usuarios y cada usuario tendrá la posibilidad de acceder a distintos subesquemas.

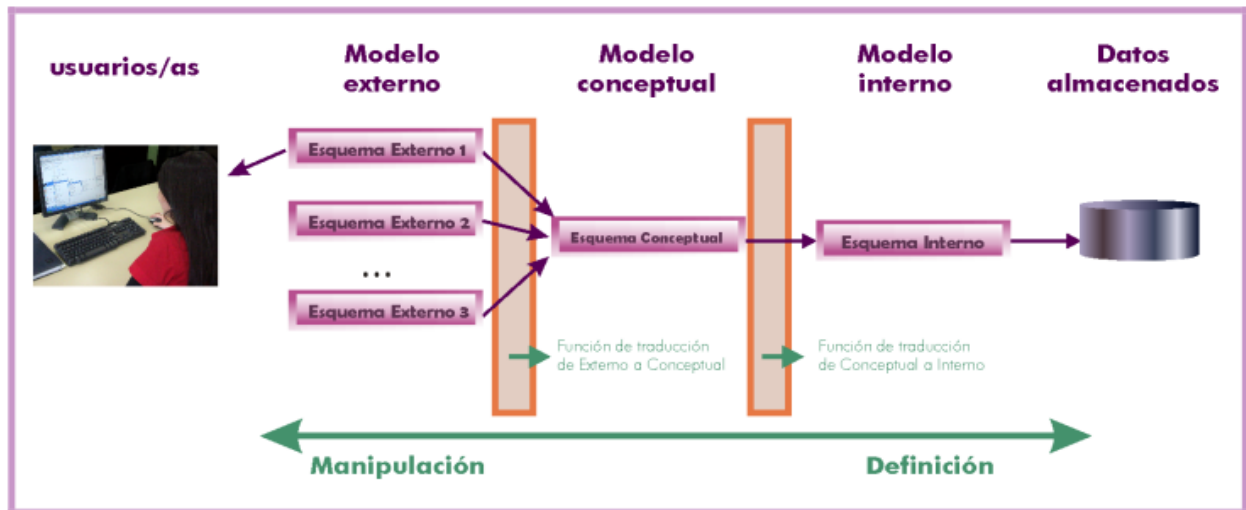


Ilustración 11. Niveles en el modelo ANSI

Ejemplo: información manejada por la organización de la liga nacional de baloncesto.

- esquema externo
 - subesquema 1: jugadores con una altura determinada.
 - Subesquema 2: partidos ganados por un equipo.
 - Subesquema 3: equipos con al menos tres jugadores de 2,05.
- esquema conceptual
 - equipos: información de los equipos
 - jugadores: información de los jugadores
 - partidos: información de los partidos
 - REGLAS: un jugador sólo pertenece a un equipo, un partido al día, etc
- esquema interno
 - archivos: equipos, jugadores y partidos
 - organización: indexada
 - clave de cada archivo
 - longitud de los registros de cada archivo
 - nombre de los campos
 - tipo de datos de cada campo.

Tipos de SGBD

Los SGBD se pueden clasificar atendiendo a distintos criterios y así tenemos los siguientes tipos:

- Atendiendo al modelo de datos en el que se basan pueden ser:
 - Modelo jerárquico.
 - Modelo en red

- **Modelo relacional**
- **Modelo orientado a objetos.**
- Atendiendo **al número de usuarios** a los que da servicio el sistema:
 - **Monousuario**, sólo admiten un usuario a la vez y el principal uso son los ordenadores personales.
 - **Multiusuario**, son la mayoría de los SGBD y atienden a varios usuarios al mismo tiempo.
- Atendiendo **al número de sitios** en los que está distribuida la BD:
 - **Centralizados**, los datos están en un solo ordenador. Pueden atender a varios usuarios, pero tanto el SGBD como la BD están en una sola máquina.
 - **Distribuidos**, la BD y el SGBD pueden estar distribuidos en varios sitios conectados por una red.
- Atendiendo **al propósito**:
 - **Propósito general**, se pueden usar para tratar cualquier tipo de base de datos y aplicación.
 - **Propósito específico**, si el rendimiento es fundamental, se puede diseñar un software de propósito especial exclusivamente para una aplicación específica. Por ejemplo, sistemas de reservar 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.

1.4 SISTEMAS GESTORES DE BBDD COMERCIALES Y LIBRES

Existen muchos SGBD comerciales en el mercado. A la hora de elegir uno u otro habrá que tener en cuenta el volumen de carga que debe soportar la BD, el S.O. usado como soporte, presupuesto, etc. Los más importantes y extendidos hasta hoy día son:

- **Oracle**, es multiplataforma, confiable y seguro. Es Cliente/Servidor y está basado en el modelo Relacional. Es muy potente y de precio elevado por lo que se usa fundamentalmente en empresas muy grandes y multinacionales. Ofrece versiones gratuitas como Oracle Database 11g Express Edition.
- **MySql**, se ofrece con dos tipos de licencia, comercial o libre. Es relacional, multihilo, multiusuario y multiplataforma. Su gran velocidad de respuesta lo hace ideal para consulta de bases de datos y plataformas web.
- **DB2**, es multiplataforma y el motor de BD relacional integra XML, esto permite almacenar documentos completos para realizar operaciones y búsquedas de manera jerárquica dentro de éste.
- **INFORMIX**, es una opción sencilla y fiable que ofrece IBM para el mundo empresarial. Es un gestor de BD relacional multiplataforma. Consume menos recursos que Oracle y dispone de utilidades muy avanzadas respecto a conectividad y funciones relacionadas con tecnologías de Internet, XML, etc.
- **Microsoft SQL Server**, es relacional y solo funciona bajo Windows, utiliza arquitectura cliente/servidor. Es la alternativa a Oracle, PostgreSQL y MySQL
- **SYBASE**, dispone de tres versiones para ajustarse a las necesidades reales de la empresa. Es un sistema relacional, de alto rendimiento, con soporte a grandes volúmenes de datos, transacciones y usuarios. Y es de bajo coste.

Otros SGBD comerciales importantes son : DBASE, ACCESS, INTERBASE y FOXPRO.

La alternativa a los **SGBD** comerciales serían los **libres** o de código abierto, también llamados Open Source. Los **más utilizados** actualmente son:

- **MySQL**, es relacional, multiplataforma, multiusuario. Posee varios motores de almacenamiento, accesible a través de múltiples lenguajes de programación y muy ligado a aplicaciones web.
- **PostgreSQL**, sistema relacional orientado a objetos. Se considera como la BD de código abierto más avanzada del mundo. Es multiplataforma y accesible desde múltiples lenguajes de programación.
- **Firebird**, es relacional, multiplataforma, con bajo consumo de recursos, excelente gestión de concurrencia, alto rendimiento y potente soporte para diferentes lenguajes.
- **Apache Derby**, está escrito en JAVA, de reducido tamaño, con soporte multilenguaje, multiplataforma, portable, puede funcionar embebido o en modo cliente/servidor.
- **SQLite**, relacional y basado en una biblioteca escrita en C. Es más rápido que MySQL o PostgreSQL y es multiplataforma.

1.5 Bases de datos centralizadas y distribuidas

Un **sistema de base de datos centralizado** es una **estructura en la que el SGBD está implantado en una sola plataforma u ordenador** desde **donde se gestionan** directamente **la totalidad de los recursos**. El rango de estos sistemas comprende desde los sistemas de BD monousuario ejecutándose en ordenadores personales hasta los sistemas de alto rendimiento.

Los **componentes de la BD centralizada son**: **los datos**, **el SW** del SGBD y los **dispositivos de almacenamiento secundario** asociados, como discos de almacenamiento y cintas para las copias de seguridad.

Ventajas:

- **Se evita la redundancia y se ahorra espacio de almacenamiento.**
- **Se evita la inconsistencia**, si un dato sólo se almacena una vez siempre existe concordancia.
- **Se centraliza la seguridad con un control central** que establece el Administrador, se asegurará que el único medio de acceder a los datos sea a través de los canales establecidos
- **El mantenimiento es más barato.** Mejor uso de los recursos y menor número de recursos humanos.

Inconvenientes:

- **Cuando falla el sistema, la recuperación de la información es difícil.**
- **Las cargas de trabajo no se pueden repartir entre varias máquinas.**
- **El mantenimiento ha de ser centralizado.**

Una BD distribuida es un conjunto de múltiples BD lógicamente **relacionadas que se encuentran distribuidas en distintos nodos** interconectados por una red de comunicaciones.

Un SGBDD es el que se encarga del manejo de la BDD y proporciona un mecanismo de acceso que hace que la distribución sea transparente a los usuarios, es decir, como si fuese un SGBD en modo local el que administrara los datos.

El SGBDD desarrollará su trabajo a través de un conjunto de sitios o nodos que poseen un sistema de procesamiento de datos completo con una BD local y un SGBD interconectados. Si los nodos están dispersos

geográficamente, se interconectarán a través de una red WAN y si se encuentran en edificios relativamente cercanos, utilizarán una red LAN. Este tipo de sistema se suele utilizar en: cadenas hoteleras, servicios bancarios, aplicaciones militares, etc.

Ventajas:

- El acceso y procesamiento de los datos es más rápido ya que varios nodos comparten carga de trabajo.
- Desde una ubicación puede accederse a información alojada en distintos lugares.
- Los costes son inferiores a los de las BD centralizadas.
- Existe cierta tolerancia a fallos. Mediante la replicación, si un nodo deja de funcionar, el sistema completo no falla.
- Este enfoque distribuido se adapta mejor a la estructura de las organizaciones y permiten la incorporación de nuevos nodos de forma flexible y ágil.
- Aunque los nodos estén conectados, tienen independencia local.

Inconvenientes:

- Existen mayores riesgos en cuanto a la seguridad del sistema.
- La inversión inicial es menor, pero el mantenimiento y control puede resultar costoso.
- Como pueden estar los datos replicados, el control de concurrencia y los mecanismos de recuperación deben ser mucho más complejos que en un sistema centralizado.
- Para conseguir la coordinación entre los distintos nodos, el sistema será sobrecargado.

Figura 1

