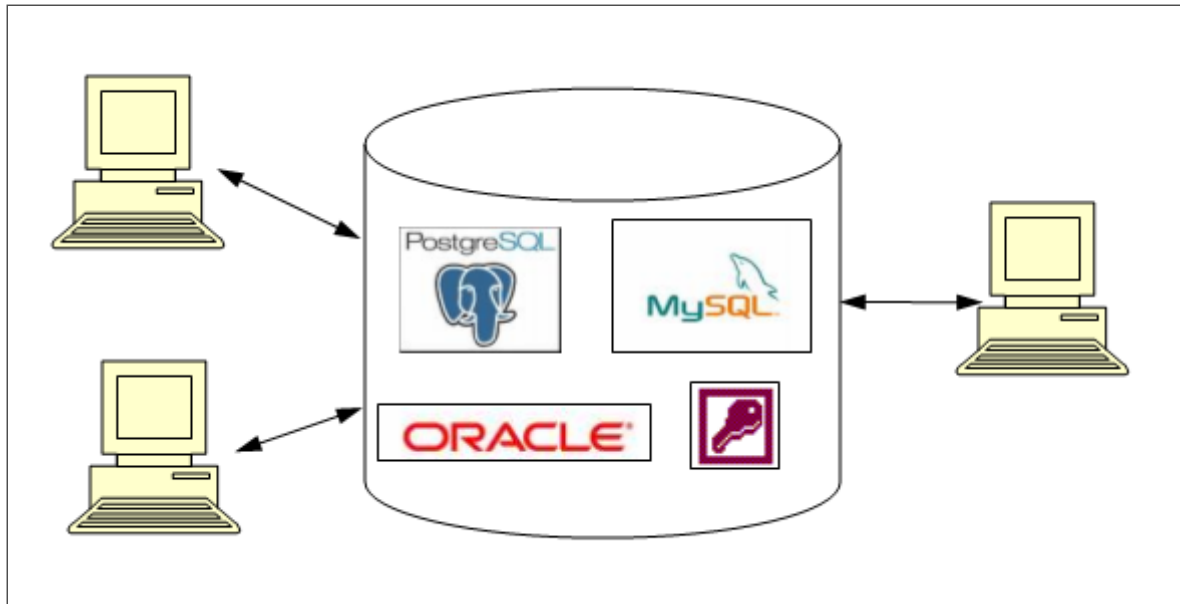


Tema 1: Introducción





objetivos

Este es el tema de introducción, y por tanto de primer contacto con la materia.

Los objetivos son:

- Ver la necesidad de mejorar los ficheros tradicionales
 - Tener claro los conceptos de Base de Datos y de Sistema Gestor de Base de Datos.
 - Ver las ventajas y también los inconvenientes
 - Entender el concepto de Modelo de Datos y conocer los distintos tipos.
 - Conocer la estandarización de la arquitectura en 3 niveles.
 - Saber de la existencia de distintos lenguajes
-

1. Archivos tradicionales

Los ficheros tradicionales, que son muy útiles para aplicaciones pequeñas y concretas, empiezan a dar problemas cuando el sistema de información va creciendo. Vamos a ver un ejemplo que ilustre la explicación.

Supongamos una empresa pequeña que ha informatizado la información de sus empleados. En un principio sólo quería una información general de ellos, para poder tener unos listados, enviar cartas a todos, etc. Así se podría plantear un fichero donde guardar la información de los empleados: nombre, apellidos, DNI, dirección, teléfono y fecha de nacimiento. *Todos* los programas que utilizan el archivo (en un principio muy pocos) deben tener la declaración de los campos del archivo. Por ejemplo, si son en lenguaje C, la definición deberá ser como la siguiente:

```
...
Struct templo
{char dni [10];
  char nombre [30];
  char direccion [30];
  char teléfono [10];
  char data_n [10]
} V_empleat; FILE * pFEmpleats; ...
```

Y un ejemplo del contenido de este fichero puede ser

18876543	Llopis Bernat, Jaume	C/ Artana, 3	964213243	07/12/1955
18900111	Garrido Vidal, Rosa	C/ Herrero, 54	964253545	25/01/1958
18922222	Nebot Aliaga, Carme	C/ Sant Vicent, 5	964216191	08/06/1959
18932165	Folch Mestre, Pilar	C/ Palància, 22	964234567	08/06/1960
18933333	Peris Andreu, Joan	C/ Balmes, 3	964223344	15/03/1960
18934567	Sebastià Broch, Ferran	C/ Magallanes, 38	964281706	14/07/1962
18944444	Garcia Tomàs, Alicia	C/ Amunt, 15	964205080	10/05/1964

Supongamos ahora que la empresa se plantea ampliar su sistema informático, e incluir el sueldo de cada empleado para poder enviar también las nóminas. Entonces decide ampliar el archivo, incluyendo el nuevo campo (sueldo). Lo primero que debería hacer es cambiar el fichero, ya que la estructura no es la misma. Se deberá crear un archivo con la nueva estructura, hacer un programita para pasar los datos del fichero viejo al nuevo archivo, borrar el viejo y en todo caso cambiar el nombre del nuevo. Después tendrán que modificar todos los programas que utilizaban el archivo para que la estructura del fichero sea la correcta, volver a compilarlos, ...

```
...
Struct templo
{char dni [10];
  char nombre [30];
  char direccion [30];
  char teléfono [10];
  char data_n [10];
  int sueldo;
} V_empleat; FILE * pFEmpleats; ...
```

Y un ejemplo del contenido de este fichero puede ser

18876543	Llopis Bernat, Jaume	C/ Artana, 3	964213243	07/12/1955	2100
18900111	Garrido Vidal, Rosa	C/ Herrero, 54	964253545	25/01/1958	2000
18922222	Nebot Aliaga, Carme	C/ Sant Vicent, 5	964216191	08/06/1959	1500
18932165	Folch Mestre, Pilar	C/ Palància, 22	964234567	08/06/1960	3000
18933333	Peris Andreu, Joan	C/ Balmes, 3	964223344	15/03/1960	2100
18934567	Sebastià Broch, Ferran	C/ Magallanes, 38	964281706	14/07/1962	2500
18944444	Garcia Tomàs, Àlicia	C/ Amunt, 15	964205080	10/05/1964	2500

Supongamos que la empresa decide seguir ampliando la información, y ahora quiere mantener información de los distintos departamentos, y los empleados que pertenecen. Quizás para no tener que repetir el proceso de adecuación, se construye un nuevo archivo de empleados donde está el DNI del empleado, el nombre, el teléfono y el departamento al que pertenece, ya que esta es la información que le interesa.

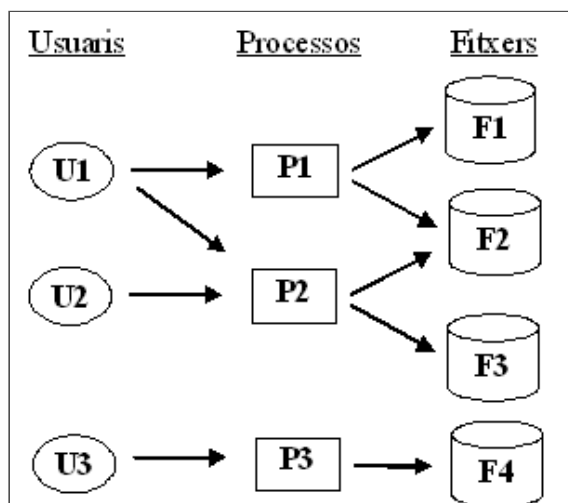
```
...
Struct TEmpleat2
{char dni [10];
  char nombre [30];
  char telèfono [10];
  char departament [20]
} V_empleat2; FILE * pFEmpleats2; ...
```

Y un ejemplo del contenido de este fichero puede ser ahora:

18876543	Llopis Bernat, Jaume	964213243	Comptabilitat
18900111	Garrido Vidal, Rosa	964253545	Administració
18922222	Nebot Aliaga, Carme	964216191	Comercial
18932165	Folch Mestre, Pilar	964234567	Direcció
18933333	Peris Andreu, Joan	964223344	Comptabilitat
18934567	Sebastià Broch, Ferran	964281706	Informàtica
18944444	Garcia Tomàs, Àlicia	964205080	Comercial

En definitiva, como el sistema está orientado a los procesos, se tiende, a medida que crece el sistema, a tener muchos archivos, además con mucha redundancia. El problema de la redundancia es, además de desperdiciar espacio, la posible inconsistencia de los datos. Por ejemplo, supongamos que ahora un determinado empleado cambia de teléfono. No se tendrá que cambiar este en un fichero, sino en dos. Si sólo se cambiara en uno de ellos habría inconsistencia.

Este sería el esquema de un sistema basado en ficheros tradicionales.



donde recuerde que en cada programa debe estar definida la estructura de los ficheros que utiliza.

Los problemas más importantes que lleva este sistema que ha ido creciendo serían:

- **Redundancia de los datos** . Los mismos datos están en múltiples ficheros, lo que conlleva ocupar más lugar de lo necesario, y además una dificultad de mantenimiento para que los datos sean consistentes.

- **Dificultad para modificar la estructura de los ficheros** . Ya se ha comentado antes que se deberían modificar tanto el archivo (con la actualización del fichero antiguo al nuevo) como los programas que lo utilizan.

- **Problemas de seguridad** . Cuando hay muchos usuarios que acceden a los archivos hay que vigilar mucho que no se puedan hacer actualizaciones no autorizadas

Volveremos a comentar los problemas cuando veamos la comparación entre ficheros tradicionales y Bases de Datos.

2. Concepto de Base de Datos

Para mejorar los problemas que se nos presentaban con los ficheros tradicionales introduciremos el concepto de Base de Datos.

2.1 Definición

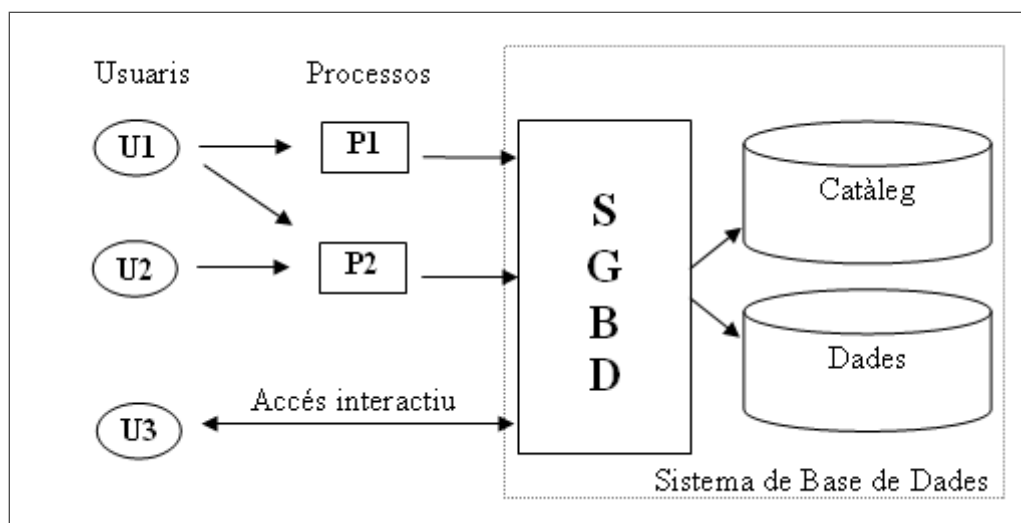
Hay muchas definiciones de Bases de Datos. Intentaremos dar una que no sigue demasiado pesada.

Una **Base de Datos** es un conjunto de datos relacionados entre sí, integrado en una única estructura global que es independiente de los programas que la utilizan y sin redundancias innecesarias.

Comentamos un poco esta definición. El primer comentario es sobre los datos. No intentaremos coger todos los datos del mundo, sino únicamente sobre las cosas que queremos estudiar, y de ellas únicamente los datos interesantes, para lograr el propósito perseguido. Por tanto, una base de datos tiene una fuente de la que se derivan los datos y unos usuarios a los que interesan los datos.

Por la propia definición, la Base de Datos se podría crear y mantener manualmente (pensemos en las fichas tradicionales de una biblioteca). Se podría crear y mantener también por medio de un grupo de programas de aplicación realizados específicamente para este trabajo, pero lo más normal, y lo que nos interesa a nosotros es que esto se haga por medio de un Sistema Gestor de Bases de datos.

Un **Sistema Gestor de Bases de Datos** (**SGBD** ; en inglés *DataBase Management System: DBMS*) es un conjunto de programas que permite a los usuarios crear y mantener bases de datos. Así la Base de Datos son nuestros datos (organizadas) y el SGBD serían productos como ACCESS o ORACLE que nos permiten crear y mantener esta base de datos.



Para la figura ya vemos que la Base de Datos se organiza en dos grandes bloques, uno es el **catálogo o diccionario de datos**, donde está la estructura y tipo de los datos. En el otro están

los propios datos. Por lo tanto vemos que la estructura de los datos con la propia Base de Datos.

Todas las peticiones de acceso a los datos no se hacen directamente, sino que se piden al SGBD, que es quien se encarga de acceder a ellas. El acceso se puede hacer por medio de procesos (programas, formularios, ...) o bien de forma interactiva (por ejemplo con consultas SQL). Algunos autores denominan al conjunto de la Base de Datos y el SGBD como **Sistema de Base de Datos**.

2.2 Comparación con ficheros tradicionales

Una vez hemos visto la definición de Base de Datos y alguna de sus características vamos a comparar con los ficheros tradicionales para ver las ventajas.

- Una Base de Datos es autodescriptiva, es decir la definición de la estructura, tipo de datos, etc. forma parte de la propia BD (el **diccionario de datos**). En los ficheros tradicionales la estructura de estos tenía que ir en cada programa.
- Independencia de los datos de los programas. Es consecuencia de la anterior. Así, en una BD, la modificación de la estructura (añadiendo algún campo o modificando alguno) no afecta a los programas que puedan acceder a ella. Ya habíamos visto que no era así en los ficheros tradicionales.
- No hay redundancia, o muy poca. Quizás conveniente un poco de redundancia para un mejor rendimiento, pero esta puede ser controlada (para evitar inconsistencias, ...)
- Múltiples vistas de usuario. Podríamos pensar que al concentrar en una única BD todos los datos se pierde la visión particular de un usuario (que en ficheros tradicionales son los ficheros individuales de cada uno de ellos). Pero no es así, ya que en las BD se pueden definir vistas, que son la parte de los datos que le interesan a un usuario determinado.
- Compartimiento de los datos por parte de muchos usuarios, con acceso simultáneo sobre ellas. Deberá haber en el software un control de concurrencia, para asegurar que la actualización simultánea de varios usuarios sea correcta.

Los inconvenientes que supone la utilización de BD son los siguientes:

- El Hardware y el software son más costosos.
- Necesidad de personal especializado.
- La implantación es más larga y difícil, y por tanto es una solución buena a medio y largo plazo.

Es decir, que si es una aplicación sencilla, tal vez sea conveniente utilizar ficheros tradicionales, pero si es de cierta envergadura o hay que ir ampliando a lo largo del tiempo, conviene una BD

2.3 Usuarios de la BD

Los múltiples usuarios que pueden utilizar la BD son de distintos tipos. Vamos a hacer una clasificación de los mismos. Esta clasificación agrupa dos grandes bloques: los usuarios con conocimientos informáticos y los usuarios que no tienen (o no tienen por qué tener).

A. Usuarios informáticos .

A.1 Diseñadores .

Encargados de diseñar la BD: elegir el tipo de información necesaria, organizarla en estructuras adecuadas.

A.2 Administradores .

Es el encargado de velar por el buen funcionamiento del sistema: administrar los usuarios y permisos, proteger la BD de errores (haciendo copias de seguridad, ..), optimizar el sistema, etc. Es decir mucho trabajo.

A.3 Analistas y programadores .

Estudian los requerimientos de los usuarios finales para hacer programas, formularios, ... que posibilitan el trabajo de éstos.

B. Usuarios finales .

Son las personas que necesitan el acceso a la BD para introducir datos, actualizarlos, consultarlos, generar informes, ... Entre ellos distinguimos:

B.1 Habituales .

También llamados **paramétricos** , suelen hacer consultas y actualizaciones constantes (siempre las mismas). Normalmente los analistas y programadores les harán los programas o formularios para hacer su trabajo.

B.2 Esporádicos .

Utilizan la BD de vez en cuando, y cada vez para obtener una información diferente. Lo más normal es que utilizan un lenguaje de consulta de la BD avanzado pero sencillo de utilizar.

3. Características deseables de un SGBD

Ya hemos visto algunas características que deben tener los SGBD. Vamos a completar esta lista.

- **Control de redundancia** . Ya hemos comentado que quizá sea conveniente en algunas ocasiones un poco de redundancia. Pero en caso de que haya, ésta debe estar controlada. Es decir, el SGBD debería proporcionar mecanismos para controlar los datos que están duplicados, y que cuando una información se actualice en un lugar, automáticamente se actualice en los demás.
- **Restricción de accesos no autorizados** . Cuando muchos usuarios utilizan la BD, es normal que no todos tengan autorización para acceder a toda la información. Puede haber información confidencial a la que pocos usuarios tienen acceso, y puede haber información en la que un usuario puede tener acceso pero que no puede modificar. El SGBD debe tener, por tanto, un subsistema de seguridad y autorización para crear usuarios (que se identifican con una contraseña) y dar distintos permisos de acceso a cada uno.
- **Suministro de múltiples interfaces para los usuarios** . Como hay muchos tipos de usuarios, con niveles de conocimientos técnicos distintos, el SGBD debe ofrecer diferentes interfaces: lenguajes de consulta para usuarios esporádicos, lenguajes de definición y control para administradores, formularios para usuarios habituales, ..
- **Cumplimiento de restricciones** . Habrá restricciones que deben cumplir los datos. Por ejemplo la nota de un alumno debe estar entre 0 y 10; la fecha de nacimiento no puede ser posterior a la actual; el curso donde está matriculado un alumno debe ser un curso existente; un alumno que esté de baja debe tener contenido en la fecha de baja, o no se le pueden poner notas; ... Algunas de estas restricciones las soportará directamente el SGBD. Otras requerirán verificación por medio de programa cuando s'introduzca o se modifique el dato.
- **Apoyo y recuperación** . Todo SGBD debe contar con recursos para recuperarse de errores del hardware o del software y dejar los datos como estaban antes de la quiebra.

4. Modelos de datos

Para poder definir la estructura de la BD nos hará falta alguna herramienta, alguna manera de expresar lo que vemos en el mundo real.

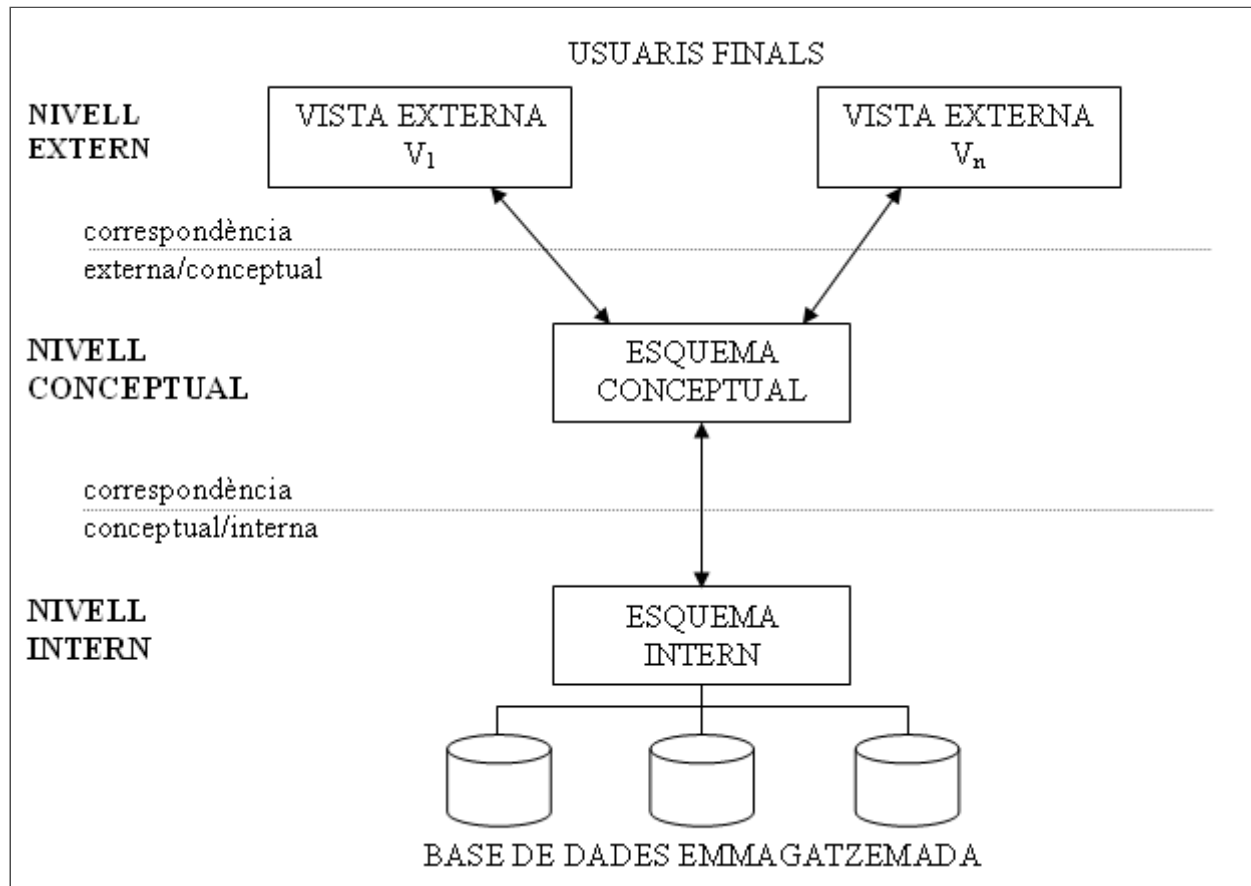
Un **Modelo de Datos** será un conjunto de conceptos que pueden servir para describir, a distintos niveles de abstracción, la estructura de una BD. A esta estructura expresada por medio del modelo de datos lo llamaremos **esquema**. Habrá que diferenciar entre esquema y los datos concretos que puede tener la BD en un momento determinado (en lo que algunos autores denominan **estado de la BD** o conjunto de **ocurrencias** o **ejemplares**).

A lo largo de la historia se han propuesto muchos modelos de datos, tantos que se clasificarán un poco.

- **Modelos de alto nivel o conceptuales** : disponen de conceptos muy próximos a la manera de percibir los datos las personas. El más conocido es el **Modelo Entidad-Relación**. Otros serían los **orientados a objetos**. Servirán para poder diseñar la BD a partir de las percepciones del mundo real. Nosotros veremos el Modelo Entidad-Relación (M. E / R). El modelo orientado a objetos está utilizándose suficiente en la actualidad, bien de forma pura o bien mezclado con el Modelo Relacional. Al final de curso veremos un ejemplo.
- **Modelos de bajo nivel o físicos** : proporcionan detalles de cómo se guardarán los datos en el ordenador.
- Modelos de **representación** o **implementación** : están a medio camino de los dos anteriores, ya que los conceptos pueden entender los usuarios finales, aunque no están demasiado alejados de la forma en que se guardan los datos en el ordenador. Son los más utilizados en los SGBD actuales. Se pueden destacar el **Jerárquico**, el **Red** (o **CODASYL**) y el **Relacional**, que es lo que nosotros veremos en profundidad.

5. Arquitectura en 3 niveles

Para estandarizar la arquitectura de los SGBD, el comité *ANSI / X3 / SPARC* propuso una arquitectura en 3 niveles, para asegurar algunas de las características que hemos visto como deseables, en concreto la de la separación entre programas y datos, las múltiples vistas y la utilización de un catálogo para la descripción de la BD. Estos son los 3 niveles:



1. Al **NIVEL INTERNO** o **FÍSICO** hay un **esquema interno** que describe la estructura física de almacenamiento de la BD utiliza un modelo físico, y describe todos los detalles para su almacenamiento (donde están los archivos, cuántos, estrategias de acceso a los datos, ...).
2. Al **NIVEL CONCEPTUAL** o **LÓGICO** hay un **esquema conceptual**, que describe la estructura de toda la BD para el conjunto de los usuarios. Oculta los detalles de las estructuras físicas de almacenamiento. Se utiliza un modelo de datos conceptual o de implementación.
3. Al **NIVEL EXTERNO** o **DE VISTAS** incluyen varios **esquemas externos** o **vistas de usuario**. Cada esquema externo describe la parte de la BD que interesa a un grupo de usuarios determinado, y oculta el resto de la BD utiliza también un modelo conceptual o de implementación.

Por lo tanto los tres esquemas son distintas maneras de describir los datos, aunque estas sólo existen realmente en el nivel físico. Pero el esquema interno debe ser totalmente transparente a los usuarios, y ellos tienen que "ver" su esquema externo. Cualquier referencia a este esquema deberá traducirse, por parte del SGBD, en referencias a los datos oportunos del esquema lógico. Y posteriormente deberá traducirse en una solicitud al esquema físico. Por ejemplo,

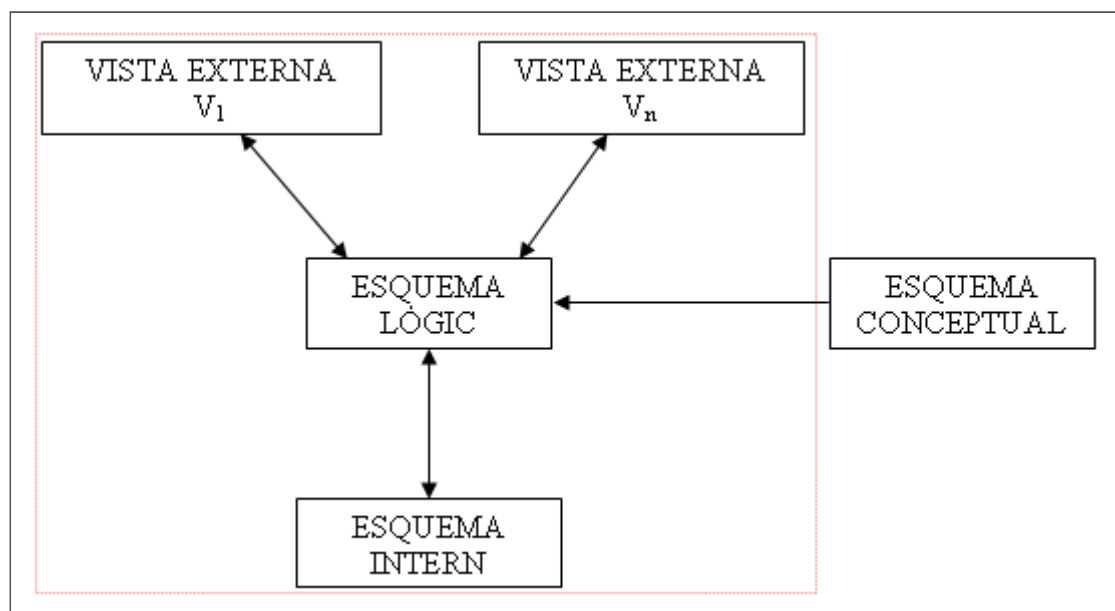
supongamos una vista donde tenemos el nombre y la edad de los empleados. Una solicitud de toda la vista externa deberá traducirse en una solicitud de los datos correctos del esquema lógico, el nombre y la fecha de nacimiento. Y por su parte deberá traducirse en una solicitud al esquema físico, donde se sabrán dónde están exactamente los datos, si hay índice para hacer más rápido el acceso, ... Posteriormente cuando ya están los datos, deberán pasar al nivel superior, y después de hacer el cálculo oportuno pasarlas al nivel externo.

El proceso de transformar solicitudes y resultados de un nivel a otro se denomina correspondencia o transformación (*mapping*). Evidentemente consumirán tiempo, pero facilitarán lo que se pretendía:

- **Independencia lógica respecto a los datos** : se puede modificar el esquema lógico sin tener que modificar los esquemas externos. Por ejemplo, en el esquema lógico puedo incorporar un campo para los empleados que sea la fecha de ingreso en la empresa. Esto no afectará para nada a la vista externa de empleados y edades.
- **Independencia física respecto a los datos** : se puede modificar el esquema físico sin tener que modificar el esquema lógico, y mucho menos los esquemas externos. Así por ejemplo se puede añadir más espacio para la BD incorporando un nuevo archivo donde se guardarán las cosas (Oracle), o se incluye un nuevo índice para acceder más rápidamente a los datos para un determinado orden.

También debemos decir que no siempre los SGBD comerciales cumplen estrictamente estos 3 niveles en su arquitectura. Sobre todo en los más pequeños, ya que obligatoriamente la correspondencia entre los niveles tarde tiempo, y retrasa los resultados.

Vamos a ampliar la anterior arquitectura en tres niveles para incorporar todo el proceso que vamos a hacer en la construcción de una BD



Donde hemos incorporado, fuera de la arquitectura en 3 niveles, es decir fuera del SGBD, el esquema conceptual, que sería la concepción de la BD en un modelo de alto nivel como el Modelo E / R. A partir de él realizaríamos el esquema lógico (con un modelo de implantación). Después ya se pasaría al esquema interno, y también a los esquemas externos.

6. Lenguajes del SGBD

Tanto para la construcción de los distintos esquemas (administradores) como para la posterior utilización para introducir, modificar, eliminar y consultar datos (todos los usuarios) se le deberán dar órdenes al SGBD. Por lo tanto el SGBD debe proporcionar unos lenguajes a los usuarios para poder realizar estas acciones. Dos serán los grupos de lenguajes:

- Lenguajes de definición. Servirán para construir los distintos esquemas. Podremos distinguir:
 - **Lenguaje de definición de datos** (*DDL : fecha definition language*). Servirán para construir el esquema lógico.
 - **Lenguaje de definición de almacenamiento** (*SDL : storage definition language*), para el esquema físico.
 - **Lenguaje de definición de vistas** (*VDL : view definition language*), para las vistas

En la práctica todos estos (que como se comentará más adelante suelen ir juntos) se les llama **DDL**

- **Lenguajes de manipulación de datos** (*DML : fecha manipulation language*). Servirán para manipular los datos, es decir hacer consultas, inserciones, modificaciones y eliminaciones.

En los SGBD actuales no se acostumbra a separar los lenguajes, sino que se utiliza un amplio lenguaje que combina todos los aspectos y permite hacerlo todo. Así por ejemplo, el lenguaje más utilizado, el SQL, cuenta con características de DML (SELECT), y de DDL en todos los niveles (CREATE TABLE esquema lógico, CREATE INDEX esquema físico, CREATE VIEW esquema externo).

Hay dos maneras de aplicar las sentencias de los lenguajes.

- De forma interactiva: se escribe una sentencia, se ejecuta, luego otra, ... Obtienen conjuntos de resultados, por eso se llaman **de conjunto en conjunto** .
- Inmersos en otro lenguaje de propósito general, llamado lenguaje **anfitrión** , donde entre los procedimientos incluyen las sentencias del DML, llamado lenguaje **huésped** o **sublenguaje** . Normalmente se obtienen registros individuales de la BD, por lo que se deberán utilizar bucles para hacer toda una consulta. Por eso se llama **de registro a registro** .

7. Clasificaciones de los SGBD

Veremos la clasificación respecto varios criterios.

- Según el modelo de datos utilizado, tendremos los SGBD **relacionales** , jerárquicos y de red. Los dos últimos, aunque hace tiempo tuvieron mucha importancia y aplicación comercial, han quedado anticuados. Nosotros nos centraremos en los SGBD Relacionales. Últimamente han aparecido los **orientados a objetos** . Intentaremos ver algunos aspectos en el último tema.
- Según el número de usuarios, tendremos sistemas **monousuario** , que sólo atienden a un usuario a la vez, y los sistemas **multiusuario** , que atienden muchos usuarios al mismo tiempo.
- Según el número de lugares donde está ubicado el sistema, tendremos los **centralizados** , en los que tanto el SGBD como los datos están en un único ordenador central (aunque se pueda acceder a la BD de forma remota). Por el contrario estarán los SGBD **distribuidos (SGBDD)** , en los que el SGBD y los datos están distribuidos en varios lugares conectados por una red. Dentro de estos podríamos hacer una distinción entre los **homogéneos** , que utilizan todos los lugares el mismo software, y los **heterogéneos o federados** que utilizan distinto software y tiene cada SGBD particular un cierto grado de autonomía local.



Llenar los espacios en blanco

una es un conjunto de datos relacionados entre sí, integrado en una única estructura global que es independiente de los programas que la utilizan y sin redundancias innecesarias.

En el conjunto de programas que permite a los usuarios crear y mantener la anterior llama



Elige la correcta

El usuuari encargado de gestionar la base de datos, dar de alta usuarios, hacer copias de seguridad, recibe el nombre de:

- ☐ programador
- ☐ paramétrico
- ☐ administrador
- ☐ analista



Preguntas Verdad / Falso

Con las herramientas que nos proporciona un **Modelo de Datos** se puede describir la estructura de una Base de Datos

verdad ☐ falso ☐

Un **Modelo de Datos** es la parte de un SGBD que permite definir la estructura de la Base de Datos

verdad ☐ falso ☐

Un ejemplo de Modelo de Implementación es el Modelo Entidad-Relación

verdad ☐ falso ☐

Un ejemplo de Modelo de alto nivel es el Modelo Jerárquico

verdad ☐ falso ☐

El Modelo Relacional, el más extendido en la actualidad, se puede considerar tanto de alto nivel como de implementación.

verdad ☐ falso ☐



elección múltiple

La arquitectura en 3 niveles significa que:

- ☐ la Base de Datos debe construirse únicamente en uno de los tres niveles "estándar"
- ☐ se pueden describir los datos en tres niveles diferentes de abstracción, aunque los datos sólo existen realmente en el nivel físico
- ☐ toda Base de Datos debe tener 3 arquitectos o "constructores"
- ☐ ninguna de las opciones anteriores es correcta



selección múltiple

Indica qué afirmaciones son correctas.

- ☐ Los SGBD no pueden controlar la redundancia, por eso es conveniente que no haya nada
- ☐ Los SGBD pueden ofrecer diferentes interfaces los usuarios
- ☐ Los SGBD pueden "imponer" restricciones a los datos que se deben introducir, como por ejemplo que una nota esté entre 0 y 10
- ☐ Todos los usuarios de la Base de Datos tienen los mismos permisos
- ☐ Los SGBD se basan en otros sistemas (como por ejemplo el Sistema Operativo) para hacer las copias de seguridad

muestra Realimentación



elección múltiple

El hecho de que la definición de los datos esté incorporada en la misma base de datos, se denomina técnicamente que la Base de Datos es

- ☐ automática
- ☐ autómata
- ☐ automóvil
- ☐ autodescriptiva

[«Anterior](#)

Licenciado bajo la [Creative Commons Attribution Non-commercial Share Alike 3.0 License](#)