

1.1 Introducción

En el mundo empresarial se maneja una gran cantidad de datos, a los que, en general, se debe acceder de una manera rápida, sencilla y fiable. Esto requiere disponer de los medios hardwares y software adecuados.

En un principio, la información se almacenaba en conjuntos de *ficheros* que estaban muy ligados a los programas de aplicación que trabajaban con ellos. Estos ficheros no guardaban ninguna relación entre sí, y los datos podían repetirse de unos a otros, lo que suponía la existencia de **información redundante** y, en muchos casos, **inconsistente**, es decir, difiere de unos ficheros a otros. A veces se precisaba cambiar la estructura de los registros de algunos ficheros, por ejemplo, para agregar nuevos campos a los registros, y esto implicaba que todos los programas de aplicación que se servían de esos ficheros tenían que modificarse y adaptarse, con la correspondiente pérdida de tiempo y dinero. Existía **dependencia entre los ficheros** que almacenaban la información y **los programas** que utilizaban esos ficheros.

Por otra parte, cualquier usuario podía acceder a los datos de un fichero, y no existían diferentes permisos para el acceso a los datos; así, el fichero no puede defenderse de accesos indebidos. Esta **falta de seguridad** conlleva una **falta de integridad**, es decir, se pueden perder datos por errores, y si se produce un acceso múltiple incontrolado por parte de dos o más programas, no hay forma de que un programa pueda saber los cambios realizados por el otro.

El **mantenimiento** de los datos resulta un proceso complicado y costoso. Así, si debido a un error se desea volver al estado anterior a dicho error, se tienen que recuperar antiguas copias de seguridad, asimismo, si se modifica algún tipo de registro, esta modificación debe reflejarse también en todos los programas que usan ese fichero.

A finales de los años sesenta surgieron las *bases de datos*, que van a eliminar o paliar muchos de estos inconvenientes. En una base de datos se almacenan todos los datos que necesite una empresa, y los programas que manejan esos datos no se han de preocupar del almacenamiento físico de los mismos: ningún cambio en la estructura de los datos afectará a los programas de aplicación que los utilicen (**independencia física y lógica de los datos**).

1.2 Repaso de bases de datos.

Una *base de datos*, es un conjunto, colección o depósito de datos almacenados en un soporte informático no volátil. Los datos están interrelacionados y estructurados de acuerdo con un modelo capaz de recoger el máximo contenido semántico. Dicho de otra forma, una *base de datos* viene a ser un conjunto de ficheros interrelacionados y unos programas que los gestionan a todos ellos, residentes sobre un hardware. Para poder llegar a esos ficheros hay que pasar ineludiblemente por el software de la base de datos.

Entre los **objetivos** que se marcan a las bases de datos están: **minimizar la redundancia y evitar la inconsistencia**; en el tema de la **seguridad**, *controlar el acceso de los usuarios, otorgando unos determinados permisos y preservar la integridad de los datos (contemplando sistemas automáticos de validación y mecanismos para la restauración de datos)*; **mantener una independencia entre datos y aplicaciones que trabajan con ellos; permitir los accesos concurrentes a los datos**.

1.2.1 Elementos relacionados con un sistema de base de datos.

Datos.- En una BD existen dos tipos de datos: *el diccionario de datos (DD)* que son los datos de uso interno del software que gestiona la base de datos, como son el tamaño de los campos, el tipo, los usuarios... Y por otra parte están los *datos que el cliente necesita almacenar*. La información de los datos debe estar integrada, es decir, como si se tratara de un único fichero. Además, los datos deben ser compartidos, es decir, muchos usuarios pueden acceder a la misma información, eso sí, con acceso controlado.

Hardware.- Toda la información a la que hace referencia en el punto anterior debe tener una capacidad adaptada a los requerimientos de la base de datos.

Software.- El software de una base de datos se conoce como *Sistema de Gestión de Base de Datos (SGBD)*. Este permite una total transparencia al usuario, que interactúa con los datos y el software pero no sabe realmente como están almacenados, facilitándole así su uso.

El SGBD debe permitir:

- Crear y reestructurar la información.
- Almacenar, actualizar y recuperar los datos.
- Verificar que cualquier modificación de los datos mantiene la integridad (desde un punto de vista racional) de los mismos.
- Acceso seguro a los datos verificando quién pide los datos y qué permisos tiene.
- Gestionar el acceso concurrente de varios usuarios o aplicaciones.
- Transparencia respecto al usuario, es decir, el usuario puede interactuar con los datos y el software, pero realmente éste no sabe como están almacenados.

Usuarios.- Entre los usuarios de una base de datos podemos distinguir:

- **Finales:** son aquellos que interactúan con los datos normales del sistema de información. Un usuario final no modifica la estructura de la base de datos. Puede tener permiso para insertar, modificar o eliminar datos, siendo la operación clásica la consulta. Este usuario no podrá modificar la estructura de una tabla. Pueden ser usuarios sin conocimientos informáticos, tan solo manejan un programa, también pueden saber SQL y hacer una consulta a través de un interprete de SQL.
- **Programador de aplicación:** es un usuario con conocimientos informáticos con la responsabilidad de escribir código para la aplicación de gestión. Habitualmente tiene la posibilidad de crear subesquemas en la base de datos y por tanto se le permite modificar y estructurar la base de datos. Tiene permiso para crear código.
- **Administrador (DBA):** Es el usuario con permisos más alto de la base de datos y tiene la responsabilidad de mantener el funcionamiento de la base de datos, definir todos los parámetros de inicialización y almacenamiento. Crea usuarios y les otorga los permisos pertinentes. Tiene que definir la forma de recuperar la base de datos y todas las políticas de seguridad.

1.2.2 Niveles de abstracción de la información.

Uno de los objetivos de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de la información, ocultando ciertos detalles acerca de cómo se almacenan los datos, pero permitiendo una recuperación eficaz de la información.

Una de las **arquitecturas más estandarizada** es la especificada por la normativa ANSI/X3/SPARC (*Standard Planning and Requirements Committee of the American National Standards Institute – Comité de Planificación y Requerimientos de estándares del Instituto Nacional de Estándares de Estados Unidos, división X3*). Según esta norma la arquitectura de una base de datos debe tener tres niveles de abstracción: **externo, conceptual e interno**.

- **Físico o Interno:** Es el nivel que define el formato de almacenamiento físico de los datos. Esta visión contempla: la organización de los ficheros, modos de acceso, índices y punteros, bloqueo de registros. El nivel interno se describe por medio de un **esquema interno** o *vista interna*.
- **Conceptual:** Es una visión total de los datos de la BD, de cómo está organizada toda la base de datos, tal y como la crea el analista conceptualmente (representación de las tablas...). Es la visión del administrador de la BD, que es el único usuario que trabaja a este nivel; el resto de usuarios trabajan a nivel externo utilizando subconjuntos de la estructura conceptual.
- **Externo:** es el nivel más alto de abstracción, es decir el más cercano al usuario, y proporciona una visión parcial de los datos. Es la visión que tiene un usuario o aplicación de la base de datos. Cada usuario tiene una vista externa diferente de la base de datos. Para cada tipo de usuario hay que especificar un **esquema externo**, **subesquema** o *vista externa*, que describe un subconjunto de la BD.

*Oracle asocia el concepto de subesquema al concepto de usuario.

El SGBD debe poder garantizar la transferencia de los datos desde el nivel interno al nivel externo, a este proceso se llama transformación de datos o mapeo (*data mapping*). Para ello existen dos niveles de correspondencia:

- **Correspondencia conceptual/interna:** Permite el paso de la vista conceptual a la vista interna, y viceversa. Especifica cómo se representan los registros y campos conceptuales en el nivel interno. Si se modifica la estructura interna de la base de datos, la correspondencia conceptual/interna deberá modificarse, para que no varíe el esquema conceptual. De este modo se conserva la independencia de los datos.
- **Correspondencia externa/conceptual:** Permite el paso de una vista externa específica a la vista conceptual, y viceversa.

Los subesquemas, el esquema conceptual, el esquema interno y las correspondencias conceptual/interna y externa/conceptual, las describe el **administrador de la base de datos** y quedan almacenados dentro del **diccionario** de la BD para futuras consultas del SGBD. Cuando un usuario desea acceder a la base de datos, el SGBD examina el diccionario de datos para comprobar si la solicitud puede ser realizada y el modo de realizar las transformaciones pertinentes de los datos.

Todo ello permitirá una mayor independencia de los datos:

Independencia a Nivel Físico: es la capacidad de modificar la estructura física de la base de datos sin modificar los programas. Por ejemplo: se cambia el disco duro pero los programas siguen igual.

Independencia lógica: es la capacidad de modificar el esquema lógico sin tener que modificar los programas. El SGBD puede controlar las dependencias entre las modificaciones de objetos bajo su control.

Independencia de los dispositivos: se consigue cuando la ejecución de los programas es independiente del sistema de almacenamiento

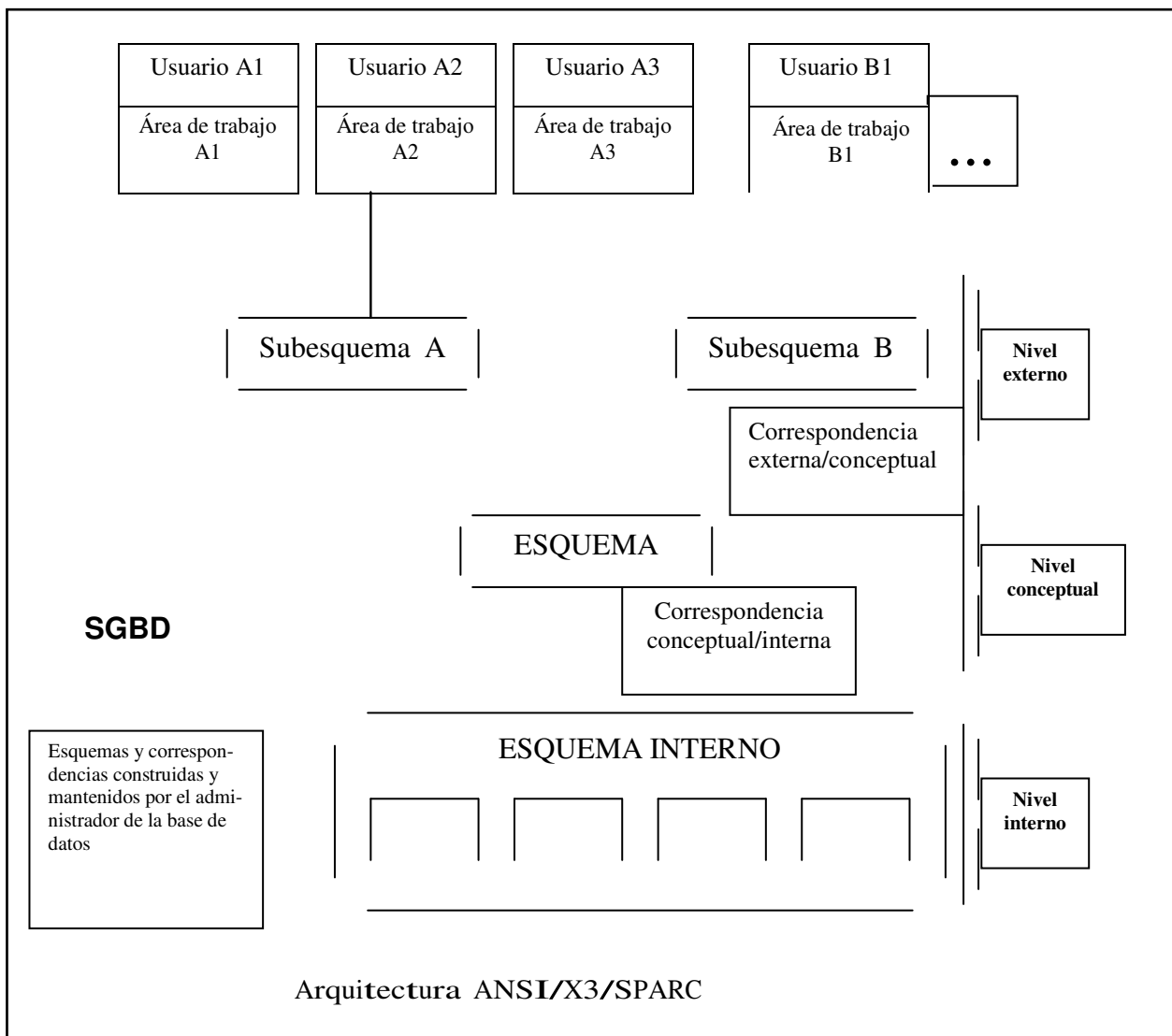


Figura 1.1

En el siguiente ejemplo de una base de datos se puede observar que se han obtenido dos subesquemas (esquemas externos) a partir del único esquema conceptual formado por las entidades: VENDEDORES, VENTAS y ARTICULOS. La estructura de cada subesquema obedece a las necesidades concretas del tipo de usuario que lo va a utilizar. Por ejemplo:

- Uno de los subesquemas será utilizado por un programa que totalizará ventas por departamento, por lo tanto la visión que le interesa tener de la base de datos es una relación de ventas ordenadas por departamentos.
- El otro subesquema lo utilizará un programa que listará una relación de empleados por departamento, por lo que sólo necesita conocer los nombres de todos los empleados y el departamento al que pertenecen.

SUBESQUEMAS

Esquema externo o **subesquema** para el programa “Ventas por departamento”

VENTAS_POR_DEPARTAMENTO

DEPARTAMENTO	CONCEPTO	IMPORTE
--------------	----------	---------

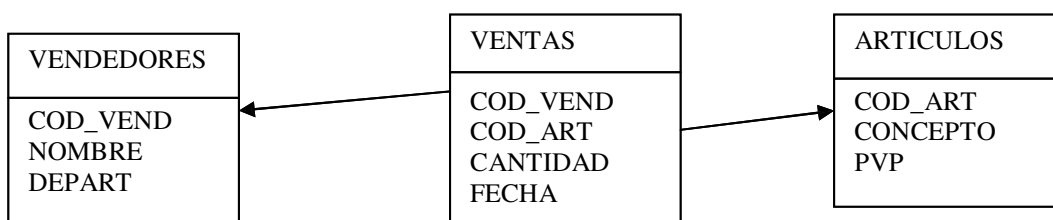
Esquema externo o **subesquema** para el programa “Relación de empleados por departamento”

VENDEDORES

DEPARTAMENTO	NOMBRE
--------------	--------

ESQUEMA CONCEPTUAL

Esquema conceptual de toda la base de datos, formado por los siguientes *registros conceptuales*:



ESQUEMA INTERNO

Esquema interno de toda la base de datos:

VENDEDORES

COD_VEND string(3)	NOMBRE string(30)	DEPART string(4)	...
--------------------	-------------------	------------------	-----

VENTAS

COD_VEND string(3)	COD_ART string(4)	CANTIDAD long	FECHA date
--------------------	-------------------	---------------	------------

ARTICULOS

COD_ART string(4)	CONCEPTO string(40)	PVP float	...
-------------------	---------------------	-----------	-----

Además se incluiría el tipo de organización, la secuencia física de los registros, modos de acceso, etc.

A continuación podemos observar algunas ocurrencias de los registros externos y conceptuales.

VISTAS EXTERNAS***VENTAS POR DEPARTAMENTO***

DEPARTAMENTO	CONCEPTO	IMPORTE
Automóvil	Antirrobo	3.450
Automóvil	Parasol	300
Hogar	Juego toallas	6.500
Hogar	Sábanas estampadas	4.500

EMPLEADOS POR DEPARTAMENTO

DEPARTAMENTO	NOMBRE
Automóvil	José López García
Hogar	Ana Ruiz Ramírez
Hogar	Julia Pérez Ramos

VISTA CONCEPTUAL***VENDEDORES***

COD_VEND	NOMBRE	DEPARTAMENTO	...
001	José López García	Automóvil	
002	Ana Ruiz Ramírez	Hogar	
003	Julia Pérez Ramos	Hogar	

VENTAS

COD_VEND	COD_ART	CANTIDAD	FECHA	...
002	H22	1	01-10-98	
001	A11	1	02-10-98	
003	H15	1	18-10-98	
001	A22	1	20-10-98	

ARTICULOS

COD_ART	CONCEPTO	PVP	...
A11	Antirrobo	3.450	
A22	Parasol	300	
H15	Juego toallas	6.500	
H22	Sábanas estampadas	4.500	

1.2.3 Funcionamiento de un SGBD.

Para describir el funcionamiento de un SGBD supondremos el caso de un programa de aplicación, que lee un registro de una base de datos, para ello realiza una solicitud al SGBD, esencialmente la secuencia, mostrada en el esquema de la figura 1.2, es la siguiente:

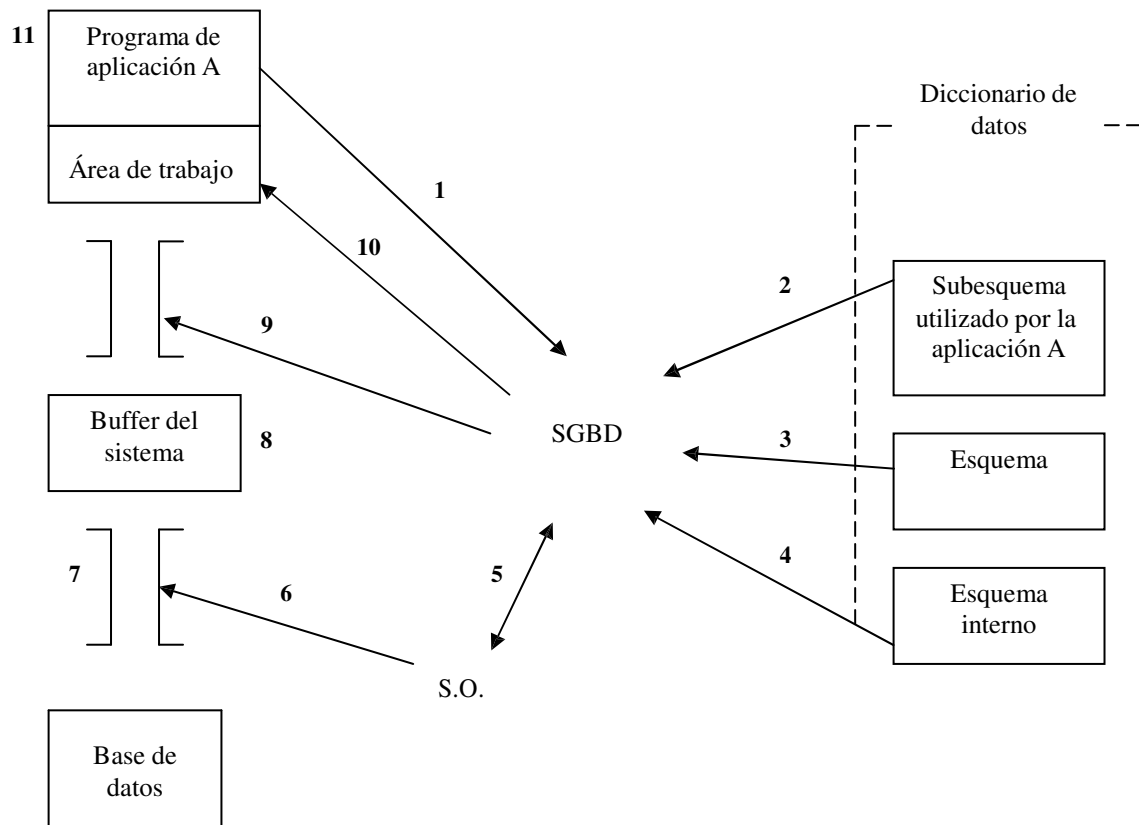


Figura 1.2

1. El programa de aplicación A pide al SGBD que lea un registro. Se proporciona la clave de acceso. El SGBD evalúa la capacidad del usuario para realizar la operación solicitada.
2. El SGBD obtiene el subesquema que utiliza el programa de aplicación A. Si no encuentra los datos solicitados, rechaza la solicitud.
3. El SGBD obtiene el esquema y determina qué tipo de datos lógicos necesita.
4. El SGBD examina el esquema interno y determina qué registro físico debe leer.
5. El SGBD ordena al S.O. que lea el registro pedido.
6. El S.O. interacciona con el dispositivo físico en el que se encuentran los datos.
7. Los datos pedidos se envían desde el dispositivo físico al buffer del sistema.
8. El SGBD analiza el esquema, el subesquema y las correspondencias externa/conceptual para realizar las transformaciones necesarias de los datos.
9. El SGBD transfiere los datos al área de trabajo del programa de aplicación A.
10. El SGBD informa al programa del éxito o fracaso de la operación de E/S, incluyendo cualquier indicación de error.
11. El programa puede ahora trabajar con los datos pedidos.

1.2.4 Lenguajes de un SGBD.

Para comunicarnos con la base de datos necesitamos un lenguaje, este lenguaje nos permitirá realizar varios procedimientos:

Lenguajes de Definición de Datos (DDL): este lenguaje es el que se utiliza para definir el esquema de la base de datos. La ejecución de sentencias de un DDL genera información de control de uso interno de la base datos, es decir genera la información en el diccionario de datos.

Lenguaje de Manejo de Datos (DML): las instrucciones del DML producen modificaciones sobre los datos en sí, pero no sobre las estructura de las tablas. Las sentencias de este lenguaje generan instrucciones de manejo de los datos.

Lenguaje de Control de Datos (DCL): son todas las sentencias que definen restricciones de control de acceso a los datos. Generan información en el diccionario de datos.

1.2.5 Tipos de Bases de Datos

Según la estructura lógica, a nivel conceptual o a nivel externo, los SGBD se clasifican en:

Jerárquico: es el primero que se utilizó, tiene una estructura arborescente, donde los nodos de nivel corresponden a los campos y cada rama a un registro. Para acceder a un campo que se encuentra en un determinado nivel, es preciso localizarlo partiendo del nivel superior y descendiendo por las ramas hasta llegar al mismo. Esta forma de organización puede hacer lenta la obtención de determinadas informaciones, aunque existen estructuras arborescentes más sofisticadas que incluyen índices y que permiten acelerar el resultado de las consultas

En red o plex: en el que se utilizan grafos cuya diferencia con la estructura en árbol donde existe más de una conexión entre los nodos de los diversos niveles, de forma que puedan recorrerse por distintos caminos sin necesidad de acudir cada vez a la raíz, y con ello la búsqueda es más flexible. Si se crean conexiones entre nodos de igual nivel, también se agiliza el acceso a campos de determinado nivel. El inconveniente esencial de esta estructura es que aumento de espacio que necesita, y que una vez más el programador tiene que recorrer la base de datos registro a registro, especificando esta vez la dirección y la relación a recorrer.

Relacional: prácticamente es el más extendido en la actualidad, será el que seguiremos en este módulo, y al que dedicaremos un tema posteriormente.