



Los SGBDR y la Empresa

Fundamentos de las Bases de Datos. Teoría general de los Sistemas de Gestión de Bases de Datos Relacionales (SGBDR)

Jose Ignacio González Gómez

Departamento de Economía Financiera y Contabilidad - Universidad de La Laguna

www.jggomez.eu

INDICE

1	La gestión de datos empresariales, aspectos generales	3
1.1	De los archivos como almacén de datos a las bases de datos	3
1.1.1	Introducción	3
1.1.2	Problemas generales de los sistemas de archivos antiguos, ejemplo de bd con Excel. Caso empresa Inmobiliaria	4
1.2	Las bases de datos y la empresa, aspectos generales	6
1.2.1	La relación entre las distintas áreas de la empresa y la necesidad del registro de los flujos de información	6
1.2.2	El papel de las BDR en los sistemas de información empresarial	8
1.2.3	El área de control de gestión de la empresa y las bases de datos	8
1.3	Los Sistemas de Gestión de Bases de Datos (SGBS) como alternativa a los ficheros de almacén de datos. Nociones básicas.....	9
1.4	Bases de Datos (Hoja de Cálculo) vs SGBD (Access, SAP, ORACLE..Etc.).....	10
1.5	Elementos que compone un Sistema de Gestión de Bases de Datos (SGBD o DBMS)	10
1.6	Estructura de una base de datos: Estructura lógica y física. Independencia de los datos	11
2	Arquitectura de los SGBD.....	13
2.1	Introducción	13
2.2	Los Modelos de Datos, distintas formas de representar la misma información	13
2.3	Modelos conceptuales, genérico modelo E-R.....	13
2.4	Modelos lógicos	14
2.4.1	El modelo en red.....	14
2.4.2	El modelo jerárquico	15
2.4.3	El modelo relacional	15

2.4.4	Modelo orientados a objetos.....	16
2.4.5	Modelo objeto relacionales.....	16
2.5	Modelos físicos.....	16
3	Objetos de un SGBD. Referencia al Access.....	17
3.1	Introducción.....	17
3.2	Componentes básicos.....	17
3.2.1	Esquema general.....	17
3.2.2	El modelo relacional. Relación de tablas asociadas por campos comunes.....	18
3.3	Conceptos fundamentales: Tabla y sus elementos. Campo y Registro.....	19
3.3.1	Las tablas y los registros aspectos generales.....	19
3.3.2	Campo, estructura del registro y registro de datos (o registro).....	19
3.3.3	Campo índice.....	20
3.3.4	Campo Clave Principal.....	20
3.4	Consultas.....	20
3.5	Formularios.....	20
3.6	Informes.....	21
3.7	Macros.....	21
3.8	Módulos.....	21
4	Lenguajes e interfaces de los SGBD.....	22
4.1	Comunicación entre el usuario y el SGBD.....	22
4.2	Lenguaje de definición de datos.....	22
4.3	Lenguaje de manipulación de datos, DML- SQL.....	22
4.4	Lenguajes de cuarta generación.....	23
4.5	Interfaces para SGBD.....	23
5	Clasificación de los SGBD.....	24
6	Servicios que deben ofrecer los SGBD.....	25
7	Pasos previos a crear una base de datos. El estudio y diseño.....	28
7.1	Consideraciones previas.....	28
7.2	Fases en el diseño de una BDR.....	28
8	Caso de estudio y reflexión. Estructura básica de un programa Contable – Financiero (I)	30
9	Bibliografía.....	33

1 La gestión de datos empresariales, aspectos generales

1.1 De los archivos como almacén de datos a las bases de datos

1.1.1 Introducción

Todas las empresas requieren almacenar información. Desde siempre lo han hecho. La información puede ser de todo tipo. Cada elemento informativo (nombre, dirección, sueldo, etc.) es lo que se conoce como dato (en inglés data).

Las soluciones utilizadas por las empresas para almacenar los datos son diversas. Antes de la aparición de la informática se almacenaban en ficheros con cajones y carpetas y fichas. Tras la aparición de la informática estos datos se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador (a veces en archivos binarios, o en hojas de cálculo, ...).

Además las empresas requieren utilizar aplicaciones informáticas para realizar tareas propias de la empresa a fin de mecanizar a las mismas. Estas aplicaciones requieren manejar los datos de la empresa.

En los inicios de la era informática, cada programa almacenaba y utilizaba sus propios datos de forma un tanto caótica. La ventaja de este sistema (la única ventaja), es que los procesos eran independientes por lo que la modificación de uno no afectaba al resto. Pero tiene grandes inconvenientes:

- Coste de almacenamiento elevado
- Datos redundantes (se repiten continuamente)
- Probabilidad alta de inconsistencia en los datos
- Difícil modificación en los datos y facilidad de problemas de inconsistencia al realizar esas modificaciones (ya que es difícil que esa modificación afecte a todos los datos)

Lógicamente la solución a este problema es hacer que todas las aplicaciones utilicen los mismos datos. Esto provoca que los datos deban estar mucho más protegidos y controlados. Además los datos forman una estructura física y funcional que es lo que se conoce como base de datos.

De esta forma una base de datos es una serie de datos relacionados que forman una estructura lógica, es decir una estructura reconocible desde un programa informático.

Esa estructura no sólo contiene los datos en sí, sino la forma en la que se relacionan. Las bases de datos empiezan a aparecer en los años 60 y triunfan en los años setenta y ochenta.

Por tanto y como resumen podemos señalar que una base de datos es un archivo o conjunto de archivos que contienen múltiples informaciones que de alguna forma guardan relación. Por ej. Una base de datos para gestionar un Videoclub almacenará múltiples informaciones sobre películas, socios, etc... y entre ellas existirán relaciones como por ej. una película con un socio por medio de un préstamo o alquiler.

Otro ejemplo es la gestión de fichas de los artículos de nuestro almacén, introduciendo, modificando, actualizando, sacando informes por impresora, etc. Podemos por ejemplo realizar consultas tales como ¿qué artículo se ha vendido más este mes? ¿qué clientes compran un determinado artículo? ¿cuál es la ganancia total del mes? etc.

En definitiva una base de datos es un archivo que contiene datos (estructurados e interrelacionados) y los objetos que definen y manejan esos datos: tablas, consultas, formularios, informes, macros y módulos.

Históricamente, las primeras aplicaciones de ordenadores se concentraron en tareas de oficina: procesamiento de entradas o procesamiento de pedidos, o ambos, nóminas, planificación del trabajo, etc. Tales aplicaciones tenían acceso a datos guardados en archivos de ordenador. Se solicitaba información (¿cuántos productos se vendieron, por quién y a quién?); y se generaban informes para transformar los datos en información útil para las decisiones gerenciales.

Por tanto el concepto de sistema de archivos es fundamental en el área empresarial en cuanto que estos son los almacenes de información. Los sistemas de archivos surgieron al tratar de informatizar el manejo de los archivadores manuales con objeto de proporcionar un acceso más eficiente a los datos.

Un sistema de archivos es un conjunto de programas que prestan servicio a los usuarios finales. Cada programa define y maneja sus propios datos.

1.1.2 Problemas generales de los sistemas de archivos antiguos, ejemplo de bd con Excel. Caso empresa Inmobiliaria

Para comprender esto vamos a utilizar como ejemplo una empresa inmobiliaria.

En esta inmobiliaria, el departamento de ventas se encarga de alquilar inmuebles. Por ejemplo, cuando un propietario pasa por el departamento de ventas para ofrecer en alquiler su piso, se rellena un formulario en donde se recogen los datos del piso, como la dirección y el número de habitaciones, y los datos del propietario. El departamento de ventas también se encarga de atender a los clientes que desean alquilar un inmueble. Cuando un cliente (posible inquilino) pasa por este departamento se rellena un formulario con sus datos y sus preferencias: si quiere un piso o una casa, el importe mensual que está dispuesto a pagar por el alquiler, etc. Para gestionar toda esta información, el departamento de ventas posee un sistema de información. El sistema tiene tres archivos: archivo de inmuebles, archivo de propietarios y archivo de inquilinos.

INMUEBLE							
Inum	Calle	Area	Población	Tipo	Hab	Alquiler	Pnum
IA14	Enmedio, 128	Centro	Castellón	Casa	6	600	P46
IL94	Riu Ebre, 24	Ronda Sur	Castellón	Piso	4	350	P87
IG4	Sorell, 5	Grao	Castellón	Piso	3	300	P40
IG36	Alicante, 1		Segorbe	Piso	3	325	P93
IG21	San Francisco, 10		Vinaroz	Casa	5	550	P87
IG16	Capuchinos, 19	Rafalafena	Castellón	Piso	4	400	P93

PROPIETARIO					
Pnum	Nombre	Apellido	Dirección	Pref	Teléfono
P46	Amparo	Felip	Asensi 24, Castellón	964	230 680
P87	Manuel	Obiol	Av.Libertad 15, Vinaroz	964	450 760
P40	Alberto	Estrada	Av.del Puerto 52, Castellón	964	200 740
P93	Yolanda	Robles	Purísima 4, Segorbe	964	710 430

INQUILINO							
Qnum	Nombre	Apellido	Dirección	Pref	Teléfono	Tipo	Alquiler
Q76	Juan	Felip	Barceló 47, Castellón	964	282 540	Piso	375
Q56	Ana	Grangel	San Rafael 45, Almazora	964	551 110	Piso	300
Q74	Elena	Abaso	Navarra 76, Castellón	964	205 560	Casa	700
Q62	Alicia	Mori	Alloza 45, Castellón	964	229 580	Piso	550

El departamento de contratos se ocupa de gestionar los contratos de alquiler de los inmuebles. Cuando un cliente desea formalizar un contrato, un empleado de la empresa rellena un formulario con los datos del inquilino y los datos del inmueble. Este formulario se pasa al departamento de contratos, que asigna un número al contrato y completa la información sobre el pago y el período del contrato. Para gestionar esta información, el departamento de contratos posee un sistema de información con tres archivos: el archivo de los contratos, el archivo de los inmuebles alquilados y el archivo de los inquilinos que tienen en vigor un contrato de alquiler.

CONTRATO							
Cnum	Inum	Qnum	Importe	Pago	Inicio	Fin	Meses
10024	IA14	Q62	600	Visa	01/06/1999	31/05/2000	12
10075	IL94	Q76	350	Efectivo	01/01/2000	30/06/2000	6
10012	IG21	Q74	550	Cheque	01/07/1999	30/06/2000	12

INMUEBLE				
Inum	Calle	Area	Población	Alquiler
IA14	Enmedio, 128	Centro	Castellón	600
IL94	Riu Ebre, 24	Ronda Sur	Castellón	350
IG21	San Francisco, 10		Vinaroz	550

INQUILINO					
Qnum	Nombre	Apellido	Dirección	Población	Teléfono
Q76	Juan	Felip	Barceló, 47	Castellón	964 282 540
Q74	Elena	Abaso	Navarra, 76	Castellón	964 205 560
Q62	Alicia	Mori	Alloza, 45	Castellón	964 229 580

Cada departamento accede a sus propios archivos mediante una serie de programas de aplicación escritos especialmente para ellos. Estos programas son totalmente independientes entre un departamento y otro, y se utilizan para introducir datos, mantener los archivos y generar los informes que cada departamento necesita. Es importante destacar que la estructura física de los archivos de datos y de sus registros está definida dentro de los programas de aplicación.

La situación es muy similar en el resto de departamentos. En el departamento de nóminas tienen un archivo con los datos de los salarios de los empleados. Los registros de este archivo tienen los siguientes campos: número de empleado, nombre, apellido, dirección, fecha de nacimiento, salario, DNI y número de la oficina en la que trabaja. El departamento de personal tiene un archivo con los datos de los empleados. Sus registros tienen los siguientes campos: número de empleado, nombre, apellidos, dirección, teléfono, puesto, fecha de nacimiento, salario, DNI y número de la oficina en la que trabaja.

Se puede ver claramente que hay una gran cantidad de datos repetidos en los archivos de estos departamentos, algo que siempre ocurre en los sistemas de archivos. A raíz de esto, los sistemas de archivos presentan una serie de inconvenientes:

- ***Separación y aislamiento de los datos.*** Cuando los datos se separan en distintos archivos, es más complicado acceder a ellos, ya que el programador de aplicaciones debe sincronizar el procesamiento de los distintos archivos implicados para asegurar que se extraen los datos correctos.
- ***Duplicación de datos.*** La redundancia de datos existente en los sistemas de archivos hace que se desperdicie espacio de almacenamiento y lo que es más importante: puede llevar a que se pierda la consistencia de los datos. Se produce una inconsistencia cuando copias de los mismos datos no coinciden.
- ***Dependencia de datos.*** Ya que la estructura física de los datos (la definición de los archivos y de los registros) se encuentra codificada en los programas de aplicación, cualquier cambio en dicha estructura es difícil de realizar. El programador debe identificar todos los programas afectados por este cambio, modificarlos y volverlos a probar, lo que cuesta mucho tiempo y está sujeto a que se produzcan errores. A este problema, tan característico de los sistemas de archivos, se le denomina también falta de independencia de datos lógica-física.
- ***Formatos de archivos incompatibles.*** Ya que la estructura de los archivos se define en los programas de aplicación, es completamente dependiente del lenguaje de programación utilizado para desarrollar dichos programas. La incompatibilidad entre archivos generados por distintos lenguajes hace que los archivos sean difíciles de procesar de modo conjunto.
- ***Consultas fijas y proliferación de programas de aplicación.*** Desde el punto de vista de los usuarios finales, los sistemas de archivos fueron un gran avance comparados a los sistemas manuales. A consecuencia de esto, creció la necesidad de realizar distintos tipos de consultas de datos. Sin embargo, los sistemas de archivos son muy dependientes del programador de aplicaciones: cualquier consulta o informe que se quiera realizar debe ser programado por él. En algunas organizaciones se conformaron con fijar el tipo de consultas e informes, siendo imposible realizar otro tipo de consultas que no se hubieran tenido en cuenta a la hora de escribir los programas de aplicación. En otras organizaciones hubo una proliferación de programas de aplicación para resolver todo tipo de consultas, hasta el punto de desbordar al departamento de proceso de datos, que no daba abasto para validar, mantener y documentar dichos programas.

1.2 *Las bases de datos y la empresa, aspectos generales*

1.2.1 *La relación entre las distintas áreas de la empresa y la necesidad del registro de los flujos de información*

Como hemos expuesto en todas las áreas fundamentales de la empresa se desarrollan múltiples actividades que en la mayoría de ocasiones se encuentran interrelacionadas, así por ejemplo supongamos la operación cotidiana como puede ser la recepción de la mercancía por parte de un proveedor con su correspondiente factura o albarán. Esta simple operación origina una serie de actividades o tareas que afectan a varias funciones

de la empresa como puede ser el área de finanzas con el reconocimiento de la deuda o pago y con el área comercial con la necesidad de actualización y control de inventarios.

De forma esquemática esta interrelación entre las diferentes áreas de la empresa podría ser resumida con el siguiente esquema:

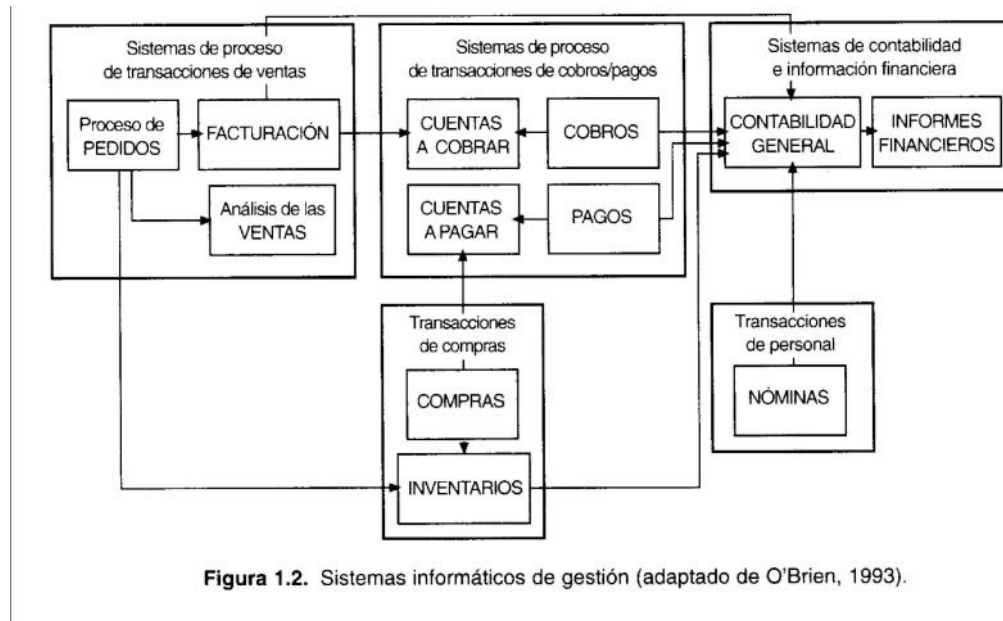


Figura 1.2. Sistemas informáticos de gestión (adaptado de O'Brien, 1993).

El volumen de información generada en la empresa conjuntamente con la complejidad de las interrelaciones existentes entre las diferentes áreas empresariales exige contar con un sistemas de información correctamente diseñados.

Estos sistemas tienen como objetivo principal el coordinar los flujos y los registros de información para desarrollar las actividades de acuerdo a las necesidades de cada empresa y en especial para la dirección, toma de decisiones y control de la actividad de la misma. Además

Un papel fundamental también de estos sistemas de información es el hecho de que constituyen instrumentos de control propiamente dicho (presupuestos, contabilidad, etc..) y consta al menos de las siguientes tareas:

- Emitir información detallada de cada unidad para que esta compruebe su cumplimiento de objetivos.
- Emitir informes que analicen y comenten los hechos más relevantes del negocio, destacando desviaciones significativas. El destinatario de estos informes es tanto la dirección general como el responsable de cada una de las unidades.

La información relevante ha de pasar a formar parte de los cuadros de mandos de cada uno de los responsables, empezando por la dirección general. Estos cuadros de mando han de responder a las variables claves del negocio y de ahí que, a diferencia del clásico sistema de ratios financieros, sean propios de cada sector de actividad e incluso de cada empresa. De hecho en una empresa diversificada en diversos sectores los cuadros de mando difieren según el sector de cada filial o unidad para adaptarse al modelo de negocio. Esta diversidad trae consigo una gran variedad de datos a usar para control de gestión, que se alejan de la medida monetaria única de los cuadros financieros.

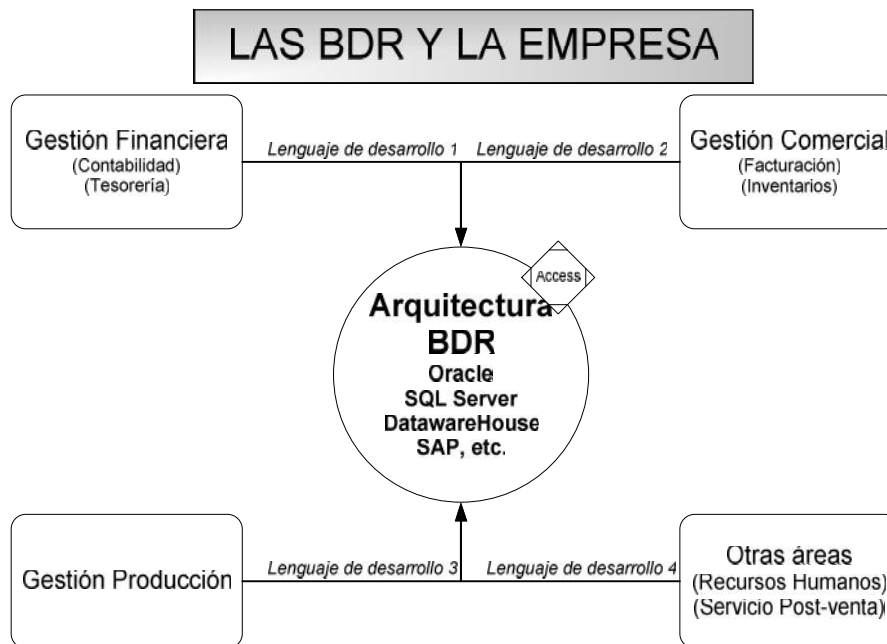
Según un estudio (Arthur Andersen, 1996), la mayor parte de las empresas españolas basan su control de gestión en hojas de cálculo. Si se acepta esta como la situación ideal, bastaría con remitir al lector a los manuales de los productos estrella, Excel, Lotus, etc.

Sin ánimo de minusvalorar este magnífico producto, hay que citar otros instrumentos útiles en el control de gestión, en este caso el de las BDR.

1.2.2 El papel de las BDR en los sistemas de información empresarial

Las bdr constituye una herramienta fundamental en el establecimiento e implantación de los sistemas de información empresarial ya que constituye un instrumento fundamental sobre los que se basan los actuales sistemas de información, es decir el actual software orientado a la gestión empresarial, independientemente del lenguaje de desarrollo, usan como arquitectura las bdr.

El uso de este tipo de arquitectura nos permite no solo el diseño de los sistemas de información de cada área empresarial sino además la interrelación y comunicación entre las distintas áreas y sistemas de información empresarial, de ahí la importancia y estudio de este tipo de arquitectura.



Un gran número de empresas disponen de sistemas de información desarrollados en distintos lenguajes de programación e incluso uno para cada área empresarial pero la característica común es el uso de una arquitectura de BDR lo que garantiza la posible comunicación e interrelación, aun cuando, como hemos dicho los mismos estén basados en diferentes lenguajes, ya que han sido adquiridos a diferentes desarrolladores de soluciones (proveedores) y/o diseñados exclusivamente para la empresa (desarrollo propio)

1.2.3 El área de control de gestión de la empresa y las bases de datos

A diferencia de otros departamentos, el control de gestión es demandante de datos contenidos en otros sistemas informáticos de la empresa, de ahí que para el control de gestión sea de gran interés el uso de lo que se ha venido en llamar "almacén de datos".

En la práctica, la forma de almacenar datos más importante y más extendida ha sido la construcción de bases de datos relacionales (BDR). Comercialmente son innumerables los sistemas de BDR que existen pudiendo citar: Oracle, SyBase, Informix, SQL Server, Dbase, FoxPro, Lotus Approach, Microsoft Access.

1.3 *Los Sistemas de Gestión de Bases de Datos (SGBS) como alternativa a los ficheros de almacén de datos. Nociones básicas.*

Estos sistemas surgieron como alternativa a los sistemas de archivos y para trabajar de un modo más efectivo, surgieron las bases de datos y los sistemas gestores de bases de datos (SGBD).

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y que ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción.

La base de datos es un gran almacén de datos que se define una sola vez y que se utiliza al mismo tiempo por muchos departamentos y usuarios. En lugar de trabajar con archivos desconectados e información redundante, todos los datos se integran con una mínima cantidad de duplicidad. La base de datos no pertenece a un departamento, se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción, que se denomina metadatos, se almacena en el diccionario de datos o catálogo y es lo que permite que exista independencia de datos lógica-física.

El SGBD es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos. En general, un SGBD proporciona los siguientes servicios:

- **Permite la definición de la base de datos mediante el lenguaje de definición de datos**. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos. Todo esto se almacenará en la base de datos.
- **Permite la inserción, actualización, eliminación y consulta de datos mediante el lenguaje de manejo de datos**. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de archivos, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de un gran número de programas de aplicación costosos de gestionar.
- **El lenguaje estándar utilizado por los SGBDR es el (Structured Query Language)**
- **Proporciona un acceso controlado a la base de datos mediante:**
 1. Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.
 2. Un sistema de integridad que mantiene la integridad y la consistencia de los datos.
 3. Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.
 4. Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del hardware o del software;
 5. Un diccionario de datos o catálogo accesible por el usuario que contiene la descripción de los datos de la base de datos.

Todos los SGBD no presentan la misma funcionalidad, depende de cada producto. En general, los grandes SGBD multiusuario ofrecen todas las funciones que se acaban de citar y muchas más. Los sistemas modernos son conjuntos de programas extremadamente complejos y sofisticados, con millones de líneas de código y con una documentación consistente en varios volúmenes. Lo que se pretende es proporcionar un sistema que permita gestionar cualquier tipo de requisitos y que tenga un 100% de fiabilidad ante cualquier fallo hardware o software. Los SGBD están en continua evolución, tratando de satisfacer los requerimientos de todo tipo de usuarios. Por ejemplo, muchas aplicaciones de hoy en día necesitan almacenar imágenes, vídeo, sonido, etc. Para satisfacer a este mercado, los SGBD deben cambiar. Conforme vaya pasando el tiempo irán surgiendo nuevos requisitos, por lo que los SGBD nunca permanecerán estáticos.

1.4 Bases de Datos (Hoja de Cálculo) vs SGBD (Access, SAP, ORACLE..Etc.)

Una base de datos no es más que un conjunto de datos relacionados entre sí, creados para un propósito específico.

Nombre	Teléfono	Dirección
Juan	223344	Paseo de la Esperanza nº7
Luisa	224455	Calle de la Pereza nº5

Este ejemplo constituye una base de datos de un listín telefónico. Lo más importante es que los datos han de estar organizados de una forma lógica, no de una forma aleatoria. Es decir, consideramos que no todas las colecciones de datos son bases de datos.

En esta línea podemos señalar también que las tareas tradicionales asociadas a las bases de datos es la introducción y actualización de información, y la realización de consultas. Como ejemplos, podemos citar sistemas bancarios, sistemas de reserva de vuelos y habitaciones de hoteles, catálogos de bibliotecas o sistemas de gestión de supermercados. A este conjunto de tareas se han sumado otras favorecidas por el avance tecnológico como son el manejo de sonidos y gráficos, como bases de datos multimedia o sistemas de información geográfica (GIS), etc.

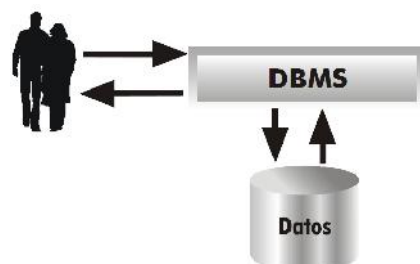
El tamaño y la complejidad de la bases de datos depende del problema que se esté resolviendo, por lo que podemos tener una base de datos personal para guardar las direcciones y teléfonos de nuestros mejores amigos, o bien tener una base de datos que almacene todos los datos clínicos de los usuarios de la Seguridad Social. Por tanto, tal cantidad de información, tiene que organizarse y controlarse para que la información pueda ser accedida y manipulada cuando sea necesario. Además, su almacenamiento ha de ser eficiente, así como las operaciones de manipulación que se llevan a cabo sobre los datos.

Un SGBD (Sistema de Gestión de Bases de Datos) es un conjunto de programas (en realidad, un sistema software) de propósito general que facilita el proceso de definición, construcción y manipulación de bases de datos para usos diversos.

1.5 Elementos que compone un Sistema de Gestión de Bases de Datos (SGBD o DBMS)

Para construir una base de datos en soporte informático con todas las informaciones a almacenar (estructuras de datos, tablas, índices , etc...), es necesario disponer de una herramienta que lo permita. Este es el cometido de un SGBD; básicamente permite crear, manipular gestionar y eliminar tanto los datos como las estructuras de una base de datos, permitiendo de esta forma el tratamiento automatizado y fácil de la información almacenada en ella. No se debe confundir base de datos con SGBD con Base de Datos, el primero es una herramienta para la creación mientras que la segunda es una solución concreta a un problema de almacenamiento de datos determinado. Por ej. Con Access (es un SGBD) podemos crear y gestionar múltiples bases de datos (para gestionar un videoclub, una biblioteca, un comercio, etc...) con diferentes necesidades de almacenamiento.

Un sistema gestor de bases de datos o SGBD (aunque se suele utilizar más a menudo las siglas DBMS procedentes del inglés, Data Base Management System) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.



Los elementos que componen un SGDB o DBMS son básicamente:

- Hardware. Máquinas en las que se almacenan las bases de datos. Incorporan unidades de almacenamiento masivo para este fin.
- Software. Es el sistema gestor de bases de datos. El encargado de administrar las bases de datos.
- Datos. Incluyen los datos que se necesitan almacenar y los metadatos que son datos que sirven para describir lo que se almacena en la base de datos.
- Usuarios. Personas que manipulan los datos del sistema. Hay tres categorías:
 - Usuarios finales. Aquellos que utilizan datos de la base de datos para su trabajo cotidiano que no tiene por qué tener que ver con la informática. Normalmente no utilizan la base de datos directamente, sino que utilizan aplicaciones creadas para ellos a fin de facilitar la manipulación de los datos. Estos usuarios sólo acceden a ciertos datos.
 - Desarrolladores. Analistas y programadores encargados de generar aplicaciones para los usuarios finales.
 - Administradores. También llamados DBA (*Data Base Administrator*), se encargan de gestionar las bases de datos.

Hay que tener en cuenta que las necesidades de los usuarios son muy diferentes en función del tipo de usuario que sean: a los finales les interesa la facilidad de uso, a los desarrolladores la potencia y flexibilidad de los lenguajes incorporados del sistema de bases de datos, a los administradores herramientas de gestión avanzada para la base de datos.

1.6 Estructura de una base de datos: Estructura lógica y física. Independencia de los datos

Las bases de datos están compuestas, de datos y de metadatos. Los metadatos son datos (valga la redundancia) que sirven para especificar la estructura de la base de datos; por ejemplo qué tipo de datos se almacenan (si son texto o números o fechas ...), qué nombre se le da a cada dato (nombre, apellidos,...), cómo están agrupados, cómo se relacionan,....

De este modo se producen dos visiones de la base de datos:

- Estructura lógica. Indica la composición y distribución teórica de la base de datos. La estructura lógica sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando. Es una estructura que permite idealizar a la base de datos. Sus elementos son objetos, entidades, nodos, relaciones, enlaces,... que realmente no tienen presencia real en la física del sistema. Por ello para acceder a los datos tiene que haber una posibilidad de traducir la estructura lógica en la estructura física.
- Estructura física. Es la estructura de los datos tan cual se almacenan en las unidades de disco. La correspondencia entre la estructura lógica y la física se almacena en la base de datos (en los metadatos).

Es la capacidad de modificar un esquema de un nivel sin afectar a los esquemas de nivel superior. Hay dos niveles de independencia de datos:

- Independencia física de datos. Es la capacidad de modificar el esquema físico sin necesidad de modificar los programas de aplicación. Esto es, por ejemplo, modificar el tamaño de un campo o modificar los índices del archivo.

- Independencia lógica de datos . Es la capacidad de modificar el esquema conceptual sin necesidad de modificar los programas de aplicación. Si, por ejemplo, se reduce la base de datos eliminando un tipo de datos, los esquemas externos que no se refieran a esta parte de la base de datos no deberán verse afectados. La independencia lógica es más difícil de conseguir que la independencia física, ya que los programas de aplicación suelen ser muy dependientes de la estructura lógica de los datos a los que acceden (se han creado a partir de ella).

2 Arquitectura de los SGBD

2.1 Introducción

Cuando un arquitecto diseña una casa, no empieza decidiendo el tipo de grifería con la que equipará los baños de la misma ni el origen de la madera de las puertas. En su actuación primeramente se ocupará de ciertas características mucho más generales de la casa, sin preocuparse de pequeños aspectos como los mencionados inicialmente, es decir, se abstraerá de los detalles. Una vez que estas consideraciones generales hayan sido resueltas irá resolviendo otros problemas, siempre yendo de lo general a lo específico. Como a continuación se explica, los SGBD se estructuran de una manera similar.

Un objetivo importante de los SGBD es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema debe ocultar los detalles sobre cómo se acceden y se manipulan los datos.

2.2 Los Modelos de Datos, distintas formas de representar la misma información

Una de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características del almacenamiento físico (características que la mayoría de usuarios no necesita conocer). Los modelos de datos son el instrumento principal que utilizan para ofrecer dicha abstracción. En esta sección se explica el concepto de modelo de datos y se mencionan los más utilizados por los SGBD.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos.

Los modelos de datos se pueden clasificar en:

- Modelos de datos de alto nivel, o modelos conceptuales, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos.
- Modelos lógicos, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente. Los modelos lógicos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implementarse de manera directa en un ordenador.
- Modelos de datos de bajo nivel, o modelos físicos, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador. Los conceptos de los modelos físicos están dirigidos al personal informático, no a los usuarios finales.

2.3 Modelos conceptuales, genérico modelo E-R

Los modelos conceptuales se usan para describir los datos y se caracterizan porque proporcionan una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. A continuación describiremos brevemente el modelo más importante dentro de esta categoría, el modelo entidad-relación.

El modelo entidad-relación

El modelo de datos E-R (entidad-relación) se basa en una percepción del mundo real que consiste en una colección de objetos básicos denominados entidades y las relaciones entre estos objetos.

Una **entidad** es un objeto que se diferencia de otros objetos mediante una serie de atributos. Por ejemplo, los atributos númeroDeCuenta y saldo describen una cuenta concreta de un banco. De la misma manera, los atributos nombre y dni describen un cliente del banco.

Una relación es una asociación entre varias entidades. Por ejemplo, una relación clienteCuenta asocia a cada cliente todas las cuentas que tiene en el banco.

La estructura lógica global de una base de datos puede representarse gráficamente mediante un diagrama E-R, que consta de los siguientes componentes:

- Rectángulos : Representan entidades
- Elipses : Representan atributos.
- Rombos : Representan relaciones entre entidades.
- Líneas : Conectan atributos con entidades y entidades con relaciones.

Cada una de estas componentes se etiqueta con la entidad, relación o atributo que representa, de forma que los diagramas obtenidos tienen un aspecto como el de la figura siguiente.



En líneas generales podemos entender por entidad como todo aquello sobre lo que es necesario almacenar información en una base de datos. Por ej. En una base de datos de una biblioteca entidades serían LIBROS, SOCIOS, etc....

Entre las diferentes entidades de una base de datos suele haber **relaciones**, las cuales permiten un tratamiento más globalizado de la información y reflejan situaciones del mundo real. Por ej. En un videoclub hay dos entidades claras PELICULAS y SOCIOS. Entre ellas existe una relación bastante evidente que es el préstamo o alquiler. De esta forma las entidades y relaciones en una base de datos se representan de forma lógica con estructuras del tipo:

(Entidad) → (Relación) → (Entidad)
SOCIO → ALQUILA → PELICULA

Tanto el Socio como el alquiler y la película, tienen sus informaciones asociadas que serán las que se almacenen en la base de datos. Informaciones podrían ser para SOCIO (nº socio, Nombre, Apellidos, Tfno.), para ALQUILER (fecha, nº de cinta, importe, nº de socio) y de PELICULA (nº de cinta, Título, Director, Genero).

Estas relaciones no necesariamente siempre tienen que constituirse en tabla, a veces se integran dentro de las propias entidades en función de los datos que contengan.

2.4 Modelos lógicos

Los tres modelos lógicos más ampliamente extendidos son el modelo relacional, el modelo en red y el modelo jerárquico.

2.4.1 El modelo en red

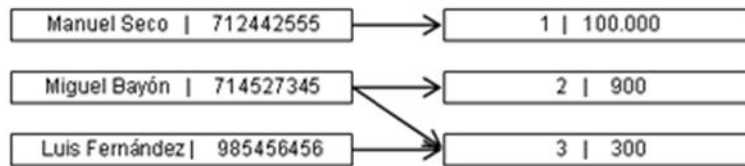
Se trata de un modelo que se utilizó durante mucho tiempo. Organiza la información en registros y enlaces. Los registros representan las entidades del modelo entidad / relación. En los registros se almacenan los datos utilizando atributos. Los enlaces permiten relacionar los registros de la base de datos.

El modelo en red más aceptado es el llamado codasyl, que durante mucho tiempo se ha convertido en un estándar.

Las bases de datos en red son parecidas a las jerárquicas sólo que en ellas puede haber más de un padre. En este modelo se pueden representar perfectamente relaciones varios a

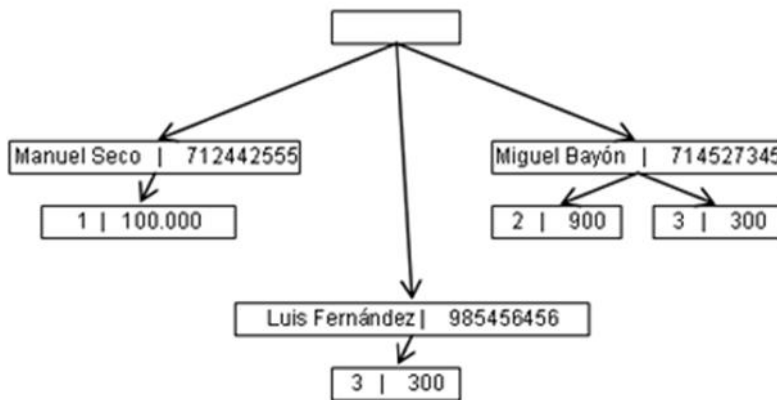
varios. Pero su dificultad de manejo y complejidad hace que se estén abandonando completamente.

Se trata de un modelo en el que los datos se representan mediante conjuntos de registros y las relaciones mediante enlaces, los cuales pueden ser vistos como punteros.



2.4.2 El modelo jerárquico

El modelo jerárquico es similar al modelo en red, en el sentido en que los datos y las relaciones entre los datos se representan mediante registros y enlaces. La diferencia con el modelo en red es que la organización de los registros se representa mediante estructuras jerárquicas (árboles), por lo que los nodos hijo, sólo pueden tener un padre, tal y como se muestra a continuación.



En ellas se organiza la información se organiza con un jerarquía en la que la relación entre las entidades de este modelo siempre es del tipo padre / hijo. De 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 tiene un padre).

2.4.3 El modelo relacional

Los datos se muestran en forma de tablas y relaciones. Este es el modelo más popular.

El modelo relacional representa los datos y las relaciones entre los datos mediante un conjunto de tablas. Cada una de las tablas tiene un nombre único y una serie de columnas también con nombres únicos. En el modelo relacional la base de datos es percibida por el usuario como un conjunto de tablas. Esta percepción es sólo a nivel lógico (en los niveles externo y conceptual de la arquitectura de tres niveles), ya que a nivel físico puede estar implementada mediante distintas estructuras de almacenamiento.

nombre	dni	saldo	numCuenta
Miguel Bayón	714527345	100.000	1
Manuel Seco	712442555	900	2
Luis Fernández	985456456	300	3

numCuenta	nombre
1	Luis Fernández
2	Miguel Bayón
3	Manuel Seco
3	Miguel Bayón

2.4.4 *Modelo orientados 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. En estos lenguajes los datos y los procedimientos se almacenan juntos. Esta es la idea de las bases de datos orientadas a objetos.

A través de esta idea se intenta que estas bases de datos consiguen arreglar las limitaciones de las relacionales. Por ejemplo el problema de la herencia, tipos definidos por el usuario, disparadores 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 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 cada vez hay más).

2.4.5 *Modelo objeto relacionales*

Tratan de ser un híbrido entre el modelo relacional y el orientado a objetos. Estas bases de datos se basan en el estándar SQL 99 que dictó las normas para estas bases de datos. En ese estándar se añade a las bases relacionales la posibilidad de almacenar procedimientos de usuario, triggers, tipos definidos por el usuario, consultas recursivas, bases de datos OLAP, tipos LOB,...

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

2.5 *Modelos físicos*

Los modelos físicos describen cómo se almacenan los datos en el ordenador: el formato de los registros, la estructura de los ficheros (desordenados, ordenados, etc.) y los métodos de acceso utilizados (índices, etc.).

3 Objetos de un SGBD. Referencia al Access

3.1 Introducción

Para organizar los datos de una base de datos es necesario crear estructuras que los contengan. Existe varios modelos de almacenamiento pero el más extendido es el modelo Relacional. Este modelo se caracteriza por almacenar organizar los datos en forma de tabla, esta es una forma habitual de organizar la información.

A continuación se muestran con un ejemplo, los aspectos básicos de una estructura tabla para el almacenamiento de datos.

Veamos el siguiente ejemplo:

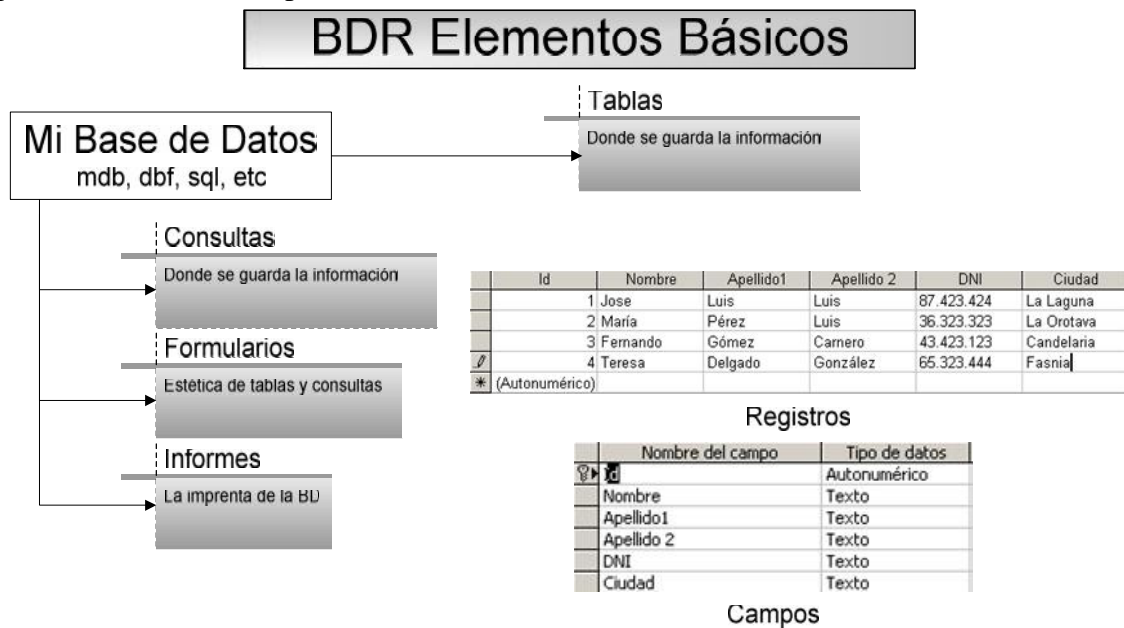
Tabla Agenda (Podemos considerar las entidades como tablas)

<i>Estructura del registro</i> →	<i>Nombre</i>	<i>Apellidos</i>	<i>Teléfono</i>
<i>Registro de datos</i> →	Antonio	García Ramos	956587412
<i>Registro de datos</i> →	María	Cantero López	956258974
	↑ <i>Campo nombre</i>	↑ <i>Campo Apellidos</i>	↑ <i>Campo Teléfono</i>

3.2 Componentes básicos

3.2.1 Esquema general

A continuación se detallan los objetos más importantes que se pueden crear y manipular en las bases de datos. Esta referencia es específica de MS-ACCESS aunque la mayor parte de ellos están disponibles en otros sistemas.





3.2.2 El modelo relacional. Relación de tablas asociadas por campos comunes

Una base de datos no tiene sentido si sus tablas no están asociadas para mostrar la información de una forma globalizada. Para asociar tablas hay que establecer relaciones (normalmente entre campos comunes). Por ej. supongamos que tenemos dos tablas en las que se almacenan datos sobre socios y prestamos de películas. En la tabla SOCIOS aparecerían todos los datos personales de los socios. En la tabla ALQUILER aparecerían todos los datos del alquiler de la película como por ej (fecha, importe, título)... Pero faltaría un campo que identificara el socio que la alquiló. Una solución podría ser poner en esta tabla el nombre del socio pero es posible que hubiese dos socios con nombre y apellidos iguales. En este caso lo mejor es colocar un campo llamado “socio” en la tabla SOCIOS donde se almacena el nº de socio para cada uno de ellos, por otro lado en la tabla ALQUILER colocaremos el campo “socio” para cada alquiler, reflejando así solamente el nº de socio que alquila. Gráficamente la estructura de las tablas sería la que se muestra a continuación:

SOCIO	NOMBRE	APELLIDOS	TFNO
1	Antonio	García	913565872
2	Maria	López	912598762
3	Carlos	Milla	912587931

SOCIO	FECHA	IMPORTE	TITULO
1	10/10/02	2	La His...
2	10/10/02	1	El Señ...
2	11/10/02	1	Seis...
1	12/10/02	2	Dos...

En el registro de alquileres se puede apreciar que Antonio García (socio nº 1) ha realizado dos alquileres uno el día 10 y otro el día 12. Hay que observar que la asociación de tablas se ha realizado mediante un campo común "SOCIO" que contiene el nº de socio para cada socio del videoclub. Este campo en la tabla SOCIOS es clave principal e identifica unívocamente a cada socio (No puede haber dos socios con el mismo nº) y en la tabla ALQUILER permite identificar al socio que alquila la película.

3.3 Conceptos fundamentales: Tabla y sus elementos. Campo y Registro.

3.3.1 Las tablas y los registros aspectos generales

Las tablas es la estructura que almacena todos los datos sobre un tipo de entidad o relación. Por ej. Clientes, Socios, Libros, Préstamos, etc...

Son las unidades básicas donde se almacena la información. Una base de datos puede contener múltiples tablas. Al usar una tabla independiente para cada tema, los datos se almacenan sólo una vez. Esto tiene como resultado una base de datos más eficaz y menos errores de entrada de datos. Las tablas organizan datos en columnas (denominadas campos) y filas (denominadas registros). Por ejemplo, cada campo de una tabla Productos contiene el mismo tipo de información para cada producto, por ejemplo, su nombre. Cada uno de los registros de esa tabla contiene toda la información acerca de un producto, por ejemplo, el nombre del producto, el Id. de proveedor, las unidades en existencia, etc

- **Tablas:** *Son los almacenes de datos. Se puede definir como la unidad donde crearemos el conjunto de datos de nuestra base de datos. Estos datos estarán ordenados en columnas verticales. Aquí definiremos los campos y sus características. El diseño de tablas consiste en la definición de los campos necesarios para almacenar correctamente la información de la base de datos.*

3.3.2 Campo, estructura del registro y registro de datos (o registro)

- Campo: unidad básica de una base de datos. Un campo puede ser, por ejemplo, el Nombre de una persona
- Registro: conjunto de campos. Un registro vendría a ser algo así como una ficha.
- Base de datos: conjunto de registros total.

Observemos el siguiente cuadro:

	Nombre del campo	Tipo de datos
	Código propiedad	Texto
	Nombre	Texto
	Apellidos	Texto
	Dirección	Texto
	Población	Texto
	DNI	Texto
	Fecha de contrato	Fecha/Hora
	Fecha de expiración	Fecha/Hora
	Alquiler mensual	Númérico
	Oficina vendedora	Texto

Este sería el primer registro de una base de datos compuesto por 4 campos: (nombre, apellido1, teléfono y provincia)

El número total de registros podemos verlo en la parte inferior (1 de 9)

Más concretamente tenemos por tanto que:

- Campo: cada uno de las informaciones que se almacenan sobre una entidad. Por ej. Edad, Teléfono, DNI, etc.... (columnas de la tabla).
- Estructura del Registro. Conjunto de campos que determinan que datos se almacenan sobre una entidad. Por ej. Nombre, Apellidos, Teléfono, etc.... (nombres de los campos que definen la estructura de la tabla).
- Registro de datos (o Registro). Cada uno de los conjuntos de datos almacenados sobre una ocurrencia de una entidad determinada. Por ej. Antonio, García Ramos,

956432178 son los datos que conforman el registro de Antonio. (filas de datos de la tabla).

3.3.3 *Campo índice*

Es un tipo especial de campo por el que se establece un orden para realiza búsquedas. La necesidad de estos campos es evidente. Si por ej. tenemos una tabla con 5000 registros de personas, resultará más fácil buscar a una persona por sus apellidos si la tabla está ordenada por este campo. Si no estuviera ordenada se podría realizar la búsqueda pero para un apellido como “Martín” en el peor de los casos tendríamos que buscar hasta el final de la tabla, mientras que con un orden (índice) sobre el campo apellidos, en el peor de los casos no pasaríamos de los registros que tuvieran el apellido “Martín” ya que mas allá no podría haber ninguna persona con ese apellido.

3.3.4 *Campo Clave Principal*

Se trata de un campo de la tabla al que se le define una propiedad especial para identificar unívocamente a una ocurrencia de la entidad. Supongamos que en nuestra tabla de 5000 registros deseamos localizar a una persona llamada Antonio García Fernández; ¿Existe la posibilidad de que dos personas tengan estos mismos nombres y apellidos?, la respuesta es Sí, de forma que si hubiese dos, no sabríamos a quien nos estaríamos refiriendo, con lo que tendríamos un problema. Para solucionarlo las bases de datos contienen campos que identifican a las entidades para evitar confusiones o dudas. Uno de los campos típicos para identificar a cada persona en una base de datos suele ser el DNI ya que este identifica unívocamente a una persona (uno a uno – dada una persona solo tiene un DNI; dado un DNI solo pertenece e identifica a una persona). Hay que destacar que en una tabla no puede haber dos ocurrencias de entidad con el mismo DNI (en una tabla de clientes no puede haber dos clientes con igual DNI; tampoco podrá registrarse dos veces al mismo cliente).

3.4 *Consultas*

Una consulta es una visión particular del conjunto de datos de la base de datos. En definitiva es un subconjunto de registros de una o varias tablas que cumplen una determinada condición. También se conocen como filtros. En cualquier caso responde a una necesidad del usuario. Por ej. Socios que hayan alquilado la película “El señor de los anillos” . Las consultas no contienen datos los obtienen de los datos almacenados en las tablas. Las consulta no tienen solo función de visualización pueden también realizar funciones de creación de tablas o de actualización de datos.

Se utilizan consultas para ver, modificar y analizar datos de formas diferentes. También pueden utilizarse como el origen de registros para formularios, informes y páginas de acceso a datos. Existen varios tipos de consultas, las más importantes son: Consulta de Selección, Consultas de parámetros y de acción.

- **Consultas:** *Es donde definiremos las operaciones que pueden realizarse sobre los datos de las tablas: limitaciones de acceso, selección de datos, inserción, modificación y borrado, cálculos...*

3.5 *Formularios*

Los formularios son herramientas que mejoran la presentación y manipulación de datos mediante ventanas similares a los programas de entorno Windows. El diseño lo realiza el usuario colocando controles y campos con el formato deseado.

- **Formulario:** *elemento en forma de ficha que permite la gestión de los datos de una forma más cómoda y visiblemente más atractiva, en definitiva son pantallas*

que se presentan a los usuarios de una base de datos para que tengan un acceso amigable a los datos y operaciones.

3.6 Informes

Los informes permiten preparar los datos para su listado en formato impreso con algunas funcionalidades adicionales como la de cálculos aritméticos (sumas, totales, cuenta de registros, etc...) o estadísticos (medias, varianzas, etc...).

- **Informe:** *permite preparar los registros de la base de datos de forma personalizada para imprimirlos.*

3.7 Macros

Son estructuras que permiten automatizar varias funciones que han de realizarse siempre bajo un único nombre de forma que al invocarlo se ejecutan todas las tareas. Es un paso previo a la programación pero sin necesidad de tantos conocimientos ya que algunas tareas están implementadas directamente en Access.

- **Macro:** *conjunto de instrucciones que se pueden almacenar para automatizar tareas repetitivas.*

3.8 Módulos

Los módulos es la parte de Access que necesita de mayores conocimientos debido a la necesidad de utilización de código de programación en Visual Basic.

- **Módulo:** *programa o conjunto de instrucciones en lenguaje Visual Basic*

4 Lenguajes e interfaces de los SGBD

4.1 Comunicación entre el usuario y el SGBD

Como los usuarios de un SGBD pueden tener distintos privilegios y distintos conocimientos informáticos, es necesario disponer de diferentes lenguajes e interfaces para cada tipo de usuarios. En esta sección vamos a ver los distintos lenguajes e interfaces que emplean los SGBD.

4.2 Lenguaje de definición de datos

Una vez que se ha finalizado la tarea de diseño de la base de datos, y que se ha seleccionado un SGBD para su implementación, el primer paso consiste en la especificación del esquema conceptual de la base de datos.

El esquema conceptual de la base de datos se especifica mediante una serie de definiciones expresadas en un Lenguaje de definición de datos (DDL, Data Definition Language). El SGBD contará con un compilador de DDL cuya función será procesar las sentencias en DDL para identificar las descripciones de los elementos de los esquemas y guardar la descripción del esquema en el diccionario de datos.

El diccionario de datos es un archivo que contiene metadatos, es decir, datos acerca de los datos. Este archivo se consulta cada vez que se leen o modifican los datos del Sistema de base de datos.

4.3 Lenguaje de manipulación de datos, DML- SQL

Una vez que se han compilado los esquemas de la base de datos, y que ya se han introducido datos en la base de datos, los usuarios necesitarán algún mecanismo para obtener información de la base de datos. Las operaciones más comunes de manipulación son la consulta, inserción, eliminación y modificación de datos. Para ello, todos los SGBD ofrecen un Lenguaje de manipulación de datos (DML, Data Manipulation Language).

En general existen dos tipos de DML:

- ***Procedimentales***. Requieren que el usuario (normalmente será un programador) especifique qué datos desea y cómo hay que obtenerlos. Esto quiere decir que hay que especificar todas las operaciones de acceso a datos llamando a los procedimientos necesarios para obtener la información requerida. Estos lenguajes acceden a un registro, lo procesan y basándose en los resultados obtenidos, acceden a otro registro, que también deben procesar. Así se va accediendo a registros y se van procesando hasta que se obtienen los datos deseados. Las sentencias de un LMD procedimental deben estar embebidas en un lenguaje de alto nivel, ya que se necesitan sus estructuras (bucles, condicionales, etc.) para obtener y procesar cada registro individual. A este lenguaje se le denomina lenguaje anfitrión. Las bases de datos jerárquicas y de red utilizan LMD procedimentales.
- ***No procedimentales***. Un LMD no procedimental se puede utilizar de manera independiente para especificar operaciones complejas sobre la base de datos de forma concisa. En muchos SGBD se pueden introducir interactivamente instrucciones del LMD desde un terminal o bien embeberlas en un lenguaje de programación de alto nivel. Los LMD no procedimentales permiten especificar los datos a obtener en una consulta o los datos que se deben actualizar, mediante una sola y sencilla sentencia. El usuario o programador especifica qué datos quiere obtener sin decir cómo se debe acceder a ellos. El SGBD traduce las sentencias del LMD en uno o varios procedimientos que manipulan los conjuntos de registros necesarios. Esto libera al usuario de tener que conocer cuál es la estructura física de los datos y qué algoritmos se deben utilizar para acceder a ellos. Las bases de datos relacionales utilizan LMD no procedurales, como SQL (Structured Query Language)

o QBE (Query-By-Example). Los lenguajes no procedimentales son más fáciles de aprender y de usar que los procedimentales, y el usuario debe realizar menos trabajo, siendo el SGBD quien hace la mayor parte.

4.4 Lenguajes de cuarta generación

No existe consenso sobre lo que es un lenguaje de cuarta generación (4GL). Lo que en un lenguaje de tercera generación (3GL) como COBOL, FORTRAN, Pascal o C requiere cientos de líneas de código, tan solo necesita de diez o veinte líneas en un 4GL. Comparado con un 3GL, que es procedimental, un 4GL es un lenguaje no procedimental: el usuario define qué se debe hacer, no cómo debe hacerse. Los 4GL se apoyan en unas herramientas de mucho más alto nivel denominadas herramientas de cuarta generación . El usuario no debe definir los pasos a seguir en un programa para realizar una determinada tarea, tan sólo debe definir una serie de parámetros que estas herramientas utilizarán para generar un programa de aplicación. Se dice que los 4GL pueden mejorar la productividad de los programadores en un factor de 10, aunque se limita el tipo de problemas que pueden resolver.

Los lenguajes SQL y QBE son ejemplos de 4GL pero hay otro tipos de 4GL incluidos habitualmente en los SGBD:

- **Un generador de formularios** es una herramienta interactiva que permite crear rápidamente formularios de pantalla para introducir o visualizar datos. Los generadores de formularios permiten que el usuario defina el aspecto de la pantalla, qué información se debe visualizar y en qué lugar de la pantalla debe visualizarse.
- **Un generador de informes** es una herramienta para crear informes a partir de los datos almacenados en la base de datos. Se parece a un lenguaje de consultas en que permite al usuario hacer preguntas sobre la base de datos y obtener información de ella para un informe. Sin embargo, en el generador de informes se tiene un mayor control sobre el aspecto de la salida. Se puede dejar que el generador determine automáticamente el aspecto de la salida o se puede diseñar ésta para que tenga el aspecto que desee el usuario final.
- **Un generador de gráficos** es una herramienta para obtener datos de la base de datos y visualizarlos en un gráfico mostrando tendencias y relaciones entre datos. Normalmente se pueden diseñar distintos tipos de gráficos: barras, líneas, etc.

4.5 Interfaces para SGBD

Normalmente, mientras que los programadores utilizan el DML para la creación de consultas, los usuarios de un SGBD utilizan un lenguaje de consulta de alto nivel. Para la mayoría de los usuarios se suelen definir interfaces de usuario amigables para la interacción con la base de datos. A continuación vamos a ver los tipos de interfaces que hay:

- **Interfaces basadas en menús.** Presentan al usuario una lista de opciones en forma de menú que guían al usuario en la petición de consultas. De esta forma no es necesario conocer la sintaxis de un lenguaje de consulta, pues permiten la creación de la consulta eligiendo las opciones que presenta la interfaz.
- **Interfaces gráficas.** Suelen presentar al usuario los esquemas en forma de diagrama, y las consultas se especifican manipulando el diagrama con el ratón.
- **Interfaces basadas en formularios.** Estas interfaces presentan un formulario al usuario en el que se rellenan los huecos del formulario para la modificación de los datos, o bien para especificar los parámetros de la consulta.
- **Interfaces de lenguaje natural.** Estas interfaces aceptan la especificación de una consulta descrita en términos de un idioma concreto y construyen expresiones DML a partir de dicha especificación.

5 Clasificación de los SGBD

Distintos criterios para diferenciar los SGBD

Conocer las diferentes alternativas que ofrecen los SGBD es importante para realizar, en el caso de que nos tengamos que enfrentar a ello, una buena elección como solución informatizada para una organización u empresa. En esta sección se nombran los distintos tipos existentes junto a una breve explicación de los mismos.

El criterio principal que se utiliza para clasificar los SGBD es el modelo lógico en que se basan. El modelo lógico empleados con mayor frecuencia en los SGBD comerciales actuales es el relacional; los modelos de red y jerárquico se utilizaron mucho en productos más antiguos. Últimamente se está desarrollando productos que tienen como base otro modelo: el modelo orientado a objetos.

Un segundo criterio para clasificar los SGBD es el número de usuarios a los que da servicio el sistema. Los sistemas monousuario sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales. Los sistemas multiusuario, entre los que se encuentran la mayor parte de los SGBD, atienden a varios usuarios al mismo tiempo.

Un tercer criterio es el número de sitios en los que está distribuida la base de datos . Casi todos los SGBD son centralizados: sus datos se almacenan en un solo computador. Los SGBD centralizados pueden atender a varios usuarios, pero el SGBD y la base de datos en sí residen por completo en una sola máquina. En los SGBD distribuidos la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red. Los SGBD distribuidos homogéneos utilizan el mismo SGBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a varias bases de datos autónomas preexistentes almacenadas en SGBD distribuidos heterogéneos. Esto da lugar a los SGBD federados o sistemas multibase de datos en los que los SGBD participantes tienen cierto grado de autonomía local.

Un cuarto criterio es el coste del SGBD. La mayor parte de los paquetes de SGBD cuestan entre 10.000 y 100.000 euros. Los sistemas monousuario más económicos para microcomputadores cuestan entre 0 y 3.000 euros. En el otro extremo, los paquetes más completos cuestan más de 100.000 euros.

Por último, los SGBD pueden ser de propósito general o de propósito específico . Cuando el rendimiento es fundamental, se puede diseñar y construir un SGBD de propósito especial para una aplicación específica, y este sistema no sirve para otras aplicaciones. Muchos sistemas de reservas de líneas aéreas son SGBD de propósito especial y pertenecen a la categoría de sistemas de procesamiento de transacciones en línea (OLTP), que deben atender un gran número de transacciones concurrentes sin imponer excesivos retrasos.

6 Servicios que deben ofrecer los SGBD

Cuando una empresa u organización decide adquirir una licencia de un SGBD para su infraestructura informática antes ha debido pasar por una fase de análisis de sus necesidades y de revisión de las posibilidades que cada producto brinda. En una situación ideal, los SGBD deberían ofrecer las mismas funcionalidades, pero como todo en la vida, esto no sucede así.

Codd, el creador del modelo relacional, ha establecido una lista con los ocho servicios que debe ofrecer todo SGBD:

1. Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).
2. Un SGBD debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos). Normalmente, un diccionario de datos almacena:
 - a. Nombre, tipo y tamaño de los datos.
 - b. Nombre de las relaciones entre los datos.
 - c. Restricciones de integridad sobre los datos.
 - d. Nombre de los usuarios autorizados a acceder a la base de datos.
 - e. Esquemas externos, conceptual e interno, y correspondencia entre los esquemas.
 - f. Estadísticas de utilización.

Algunos de los beneficios que reporta el diccionario de datos son los siguientes:

- a. La información sobre los datos se puede almacenar de un modo centralizado por lo que las redundancias y las inconsistencias se pueden identificar más fácilmente.
 - b. El significado de los datos se puede definir, lo que ayudará a los usuarios a entender el propósito de los mismos.
 - c. El impacto que puede producir un cambio se puede determinar antes de que sea implementado, ya que el diccionario de datos mantiene información sobre cada tipo de dato, todas sus relaciones y todos sus usuarios.
 - d. Se puede hacer respetar la seguridad y la integridad.
 - e. Se puede proporcionar información para auditorías.
3. Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos. Una transacción en el sistema informático de la empresa inmobiliaria sería 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 base de datos. Si la transacción falla durante su realización, por ejemplo porque falla el hardware, la base de datos quedará en un estado inconsistente. Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.
4. Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los datos que comparten. El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está

actualizando datos, pueden interferir de modo que se produzcan inconsistencias en la base de datos. El SGBD se debe encargar de que estas interferencias no se produzcan en el acceso simultáneo.

5. Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente. Este fallo puede ser a causa de un fallo en algún dispositivo hardware o un error del software, que hagan que el SGBD aborte, o puede ser a causa de que el usuario detecte un error durante la transacción y la aborte antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.
6. Un SGBD debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.
7. Un SGBD debe ser capaz de integrarse con algún software de comunicación. Muchos usuarios acceden a la base de datos desde terminales. En ocasiones estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD. En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes y responde de modo similar. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él para que el sistema sea comercialmente viable.
8. Un SGBD debe proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar. Por ejemplo, se puede establecer la restricción de que cada empleado no puede tener asignados más de diez inmuebles. En este caso sería deseable que el SGBD controlara que no se sobrepase este límite cada vez que se asigne un inmueble a un empleado.

Además, de estos ocho servicios, es razonable esperar que los SGBD proporcionen un par de servicios más:

- Un SGBD debe permitir que se mantenga la independencia entre los programas y la estructura de la base de datos. La independencia de datos se alcanza mediante las vistas o subesquemas. La independencia de datos física es más fácil de alcanzar, de hecho hay varios tipos de cambios que se pueden realizar sobre la estructura física de la base de datos sin afectar a las vistas. Sin embargo, lograr una completa independencia de datos lógica es más difícil. Añadir una nueva entidad, un atributo o una relación puede ser sencillo, pero no es tan sencillo eliminarlos.
- Un SGBD debe proporcionar una serie de herramientas que permitan administrar la base de datos de modo efectivo. Algunas herramientas trabajan a nivel externo, por lo que habrán sido producidas por el administrador de la base de datos. Las herramientas que trabajan a nivel interno deben ser proporcionadas por el distribuidor del SGBD. Algunas de ellas son:
 - Herramientas para importar y exportar datos.
 - Herramientas para monitorizar el uso y el funcionamiento de la base de datos.
 - Programas de análisis estadístico para examinar las prestaciones o las estadísticas de utilización.

- Herramientas para reorganización de índices.
- Herramientas para aprovechar el espacio dejado en el almacenamiento físico por los registros borrados y que consoliden el espacio liberado para reutilizarlo cuando sea necesario.

7 Pasos previos a crear una base de datos. El estudio y diseño

7.1 Consideraciones previas

Antes de comenzar a crear una base de datos algo más compleja que la de nuestro primer ejemplo, convendría establecer un croquis o borrador en papel de los objetivos de la misma. Si es una base de datos para otra persona o empresa, es conveniente primero hablar con los que van a utilizar la BD para determinar qué uso se va a dar de la misma. Si fuese por ejemplo una base de datos para gestionar un comercio, estas preguntas podrían ser:

1. Posibilidad de introducir, modificar, consultar y listar los datos
2. Informes sobre artículos más vendidos
3. Etiquetas postales para clientes
4. Existencias de artículos
5. Pedidos a proveedores, etc

A partir de las preguntas, correcciones y apuntes que hagamos, podemos comenzar a construir las tablas que formarán nuestra BD. Es muy importante no introducir 200 campos por ejemplo en una misma tabla, sino dividir la información en varias tablas que puedan gestionarse por separado, pero enlazarse cuando sea necesario.

Por tanto es muy importante siempre pasar algo de tiempo estudiando el diseño de la base de datos, mejor si lo haces en papel. Los puntos que aparecen a continuación te puede ayudar o dar una idea de lo que tienes que tener en cuenta a la hora de crear esta estructura.

Lo primero que deberías hacer es pensar en todos los datos que debe recoger la base de datos (futuros campos). Posteriormente se debe agrupar en conjuntos menores (futuras tablas).

7.2 Fases en el diseño de una BDR

Las fases de diseño de una base de datos podrían ser:

1. Determinar el propósito: Saber para que vamos a utilizar la base de datos
 - Análisis de la información que vamos a procesar: Cuales son los datos con los que vamos a trabajar.
 - Comparar con la información ya existente (hojas de pedidos, facturas, etc). Si la base de datos se crea sobre datos existentes.
 - Pensar en el usuario medio: Saber quien la va a utilizar, y la forma que le será más fácil para manejar los datos.
2. Determinar las tablas.
 - Separar los datos por temas: Los datos a utilizar los separaremos de la forma más específica que podamos, para que a la hora de buscar cualquiera de ellos, sepamos exactamente donde encontrarlos.
 - Evitar duplicados de información: No repetiremos datos de unas tablas a otros (especificar las relaciones entre tablas).
 - Evitar eliminar información necesaria: Mirar que se incluyan en la base de datos toda la información que necesitemos a la hora de trabajar con la base de datos.
3. Determinar los campos.
 - Los campos han de ser específicos del tema de la tabla: Cada tabla tendrá sus propios campos.
 - No crear campos calculados o derivados de otros campos: No añadiremos campos a la tabla como por ejemplo la suma de

otros dos campos. Porque nos supondría un espacio innecesario en la tabla y en la base de datos.

- Decidir cual será el tipo de campo de la tabla, ya que si nos equivocamos al decidir el tipo de campo, después podría suponernos la pérdida de datos de la tabla, al cambiar el tipo de campo.
- Minimizar al máximo el tamaño de los campos: No crearemos campos excesivamente grandes que vayan a contener datos pequeños.
- Definir los índices: Especificar cuáles serán las claves de las tablas, para hacer que la búsqueda en las mismas sea más rápida.
 - Clave principal
 - Índices fáciles de recordar.

8 Caso de estudio y reflexión. Estructura básica de un programa Contable – Financiero (I)

Tomando en consideración la técnica contable basada en el método contable (partida doble) y el lenguaje contable (Diccionario Plan General Contable) y los requerimientos de básicos demandados como son consulta de mayores de cuentas, elaboración del estado patrimonial y cuenta de resultados básicamente.

Diario de Contabilidad			
DEBE			HABER
----- 02-01-03 -----			
Asiento1	12.000,00 um 570 Caja	A 100 Capital Social	12.000,00 um
(Asiento de creación de la sociedad)			
----- 10-01-03 -----			
Asiento2	9.500,00 um 572 Bancos c/c	A 570 Caja	9.500,00 um
(Nuestro ingreso n° 1234)			
----- 15-01-03 -----			
Asiento3	3.000,00 um 600 Compras de Mercaderías	A 400 Proveedores	3.000,00 um
(S/F n° 1234-B)			
----- 15-02-03 -----			
Asiento4	1.500,00 um 430 Clientes	A 700 Ventas de Mercaderías	1.500,00 um
(N/F n° 1-A)			
----- 07-03-03 -----			
Asiento5	1.000,00 um 640 Sueldos y salarios	A 572 Banco c/c	1.000,00 um
(n° de cheque 3434)			
----- 03-04-03 -----			
Asiento6	450,00 um 642 Seguridad Social Empresa	A 572 Banco c/c	450,00 um
(n° de cheque 3435)			
----- 25-05-03 -----			

Ilustración 1

CUENTAS DE MAYOR			
570 Caja		100 Capital Social	
Debe	Heber	Debe	Heber
(1) 12.000,00 um	(2) 9.500,00 um	(21) 12.000,00 um	(1) 12.000,00 um
(15) 1.500,00 um	(7) 600,00 um		
	(8) 750,00 um		
	(9) 100,00 um		
	(10) 400,00 um		
	(12) 350,00 um		
	(14) 100,00 um		
	(21) 1.700,00 um		
572 Bancos c/c		400 Proveedores	
Debe	Heber	Debe	Heber
(2) 9.500,00 um	(5) 1.000,00 um	(21) 5.000,00 um	(3) 3.000,00 um
	(6) 450,00 um		(13) 2.000,00 um
	(21) 8.050,00 um		
600 Compras de Mercaderías		700 Ventas de Mercaderías	
Debe	Heber	Debe	Heber
(3) 3.000,00 um	(19) 5.000,00 um	(20) 5.000,00 um	(4) 1.500,00 um
(13) 2.000,00 um			(11) 1.500,00 um
			(16) 2.000,00 um
411 Acreedores Efectos		411 Acreedores Efectos	
Debe	Heber	Debe	Heber
(21) 3.600,00 um	(10) 3.600,00 um		

Ilustración 2

Cuenta de Pérdidas y Ganancias Correspondiente al Ejercicio Terminado 200X				
Nº CUENTA		Nota Memoria	(Debe) Haber	
			200X	200X-1
700,701,702,703,704, 705,(706),(708),(709)	1. Importe neto de la cifra de negocios		5.000,00 €	
71*,(6930),7930	2. Variacion de existencias de Productos			
73	3. Trabajos realizados por la empresa para su activo			
(600),(601),(602),(607),608,609,61*,(6931),(6932),(6933),606,7931,7932,7933	4. Aprovisionamientos		3.500,00 €	
740,747,75	5. Otros ingresos de explotacionn			
(64),7950,7957	6. Gastos de personal		1.450,00 €	
(62),(631),(634),636,639,(65),(694),(695),794, 7954	7. Otros gastos de explotacion		1.300,00 €	
(68)	8. Amortización del inmovilizado			
746	9. Imputación de subvenciones de inmovilizado no financiero y otras			
7951,7952,7955,7956	10. Excesos de provisiones			
(670),(671),(672),(690),(691),(692),770,771,772, 790,791,792	11. Deterioro y resultado por enajenaciones del inmovilizado		4.000,00 €	
(678),778	12. Otros resultados			
A) RESULTADO DE EXPLOTACIÓN (1+2+3+4+5+6+7+8+9+10+11+12)			- 5.250,00 €	
760,761,762,769	13. Ingresos financieros			
(661),(662),(665),(669)	14. Gastos financieros			
(663),763	15. Variación de valor razonable en instrumentos financiero			
(668),768	16. Diferencias de cambio			
(666),(667),(673),(675),(696),(697),(698),(699),766,773,775,796,797,798,799	17. Deterioro y resultado por enajenaciones de instrumentos financieros			
B) RESULTADO FINANCIERO (13+14+15+16 +17)			- 5.250,00 €	
C) RESULTADO ANTES DE IMPUESTOS (A+B)			- 5.250,00 €	
6300*,6301*(,633),638	18. Impuestos sobre beneficios			
D) RESULTADO DEL EJERCICIO (C + 18)			- 5.250,00 €	

Ilustración 3

<i>Cuentas Anuales: Balance-Activo</i>				
Nº CUENTA	ACTIVO	NOTAS MEMORIA	200X	200X-1
A) ACTIVO NO CORRIENTE			600,00 €	
20,(280),(290)	I. Inmovilizado intangible.			
21,(281),(291),23	II. Inmovilizado material.		600,00 €	
22,(282),(292)	III. Inversiones inmobiliarias.			
2403,2404,2413,2414,2423,2424,2433,2434,(2483),(2484),(2933),(2934),(2943),(2944),(2953),(2954)	IV. Inversiones en empresas del grupo y asociadas.			
2405,2415,2425,2435,(2485),250,251,252,253,254,255,256,257,258,(259),26,(2935),(2945),(2955),(296),(297),(298),474	V. Inversiones financieras a largo plazo.			
	VI. Activos por Impuesto diferido.			
B) ACTIVO CORRIENTE			14.750,00 €	
580,581,582,583,584,(599)	I. Activos no corrientes mantenidos para la venta.			
30,31,32,33,34,35,36,(39),407	II. Existencias.		1.500,00 €	
	III. Deudores comerciales y otras cuentas a cobrar.			
430,431,433,434,435,436 (437)(490)	1. Clientes por ventas y Prestaciones de servicios		3.500,00 €	
558	2. Accionistas (socios) por desembolsos exigidos			
44,460,470,471,472,(493),5531,5533,544	3. Otros deudores			
(5933),(5934),(5943),(5944),(5953),(5954) 2550,5305,5315,5325,5335,(5385),540,541,542,543,545,546,548.	IV. Inversiones en empresas del grupo y asociadas a corto plazo.			
	V. Inversiones financieras a C/P			
	VI. Periodificaciones			
57	VII. Efectivo y otros activos líquidos equivalentes.		9.750,00 €	
TOTAL ACTIVO (A+B)			15.350,00 €	

Ilustración 4

Se pide

Llevar a cabo un análisis y reflexión que permita convertir o traducir el esquema de funcionamiento contable tradicional en términos de base de datos distinguiendo o identificado:

- Tablas principales y auxiliares necesarias.
- Definir las relaciones y dependencias que se pudieran provocar, tablas relacionados o vinculadas.
- Identificar los campos básicos de cada tabla, tipo y características.
- Consultas
- Informes básicos o plantillas.

Sugerencias, recomendaciones.

- Macros: Asientos automáticos. Asientos de cierre y apertura... otros asientos predefinidos.
- Diario contable (tabla maestra e informes)

- Estructura del Plan General Contable básica (Cuentas de mayor o mastras)
- Plan General contable de la empresa (Cuentas auxiliares o de desarrollo)

9 Bibliografía

http://aulavirtual.miguelbayon.com/typo3_aulaVirtualv1_0/192.168.0.5/index5255.html?id=9
Bases de Datos guía de aprendizaje: Jorge Sánchez (www.jorgesanchez.net) año 2004



Los SGBDR y la Empresa

Fundamentos de las Bases de Datos. Teoría general de los Sistemas de Gestión de Bases de Datos Relacionales (SGBDR)

Jose Ignacio González Gómez

Departamento de Economía Financiera y Contabilidad - Universidad de La Laguna

www.jggomez.eu

INDICE

- 1 La gestión de datos empresariales, aspectos generales 3
 - 1.1 De los archivos como almacén de datos a las bases de datos 3
 - 1.1.1 Introducción 3
 - 1.1.2 Problemas generales de los sistemas de archivos antiguos, ejemplo de bd con Excel. Caso empresa Inmobiliaria 4
 - 1.2 Las bases de datos y la empresa, aspectos generales 6
 - 1.2.1 La relación entre las distintas áreas de la empresa y la necesidad del registro de los flujos de información 6
 - 1.2.2 El papel de las BDR en los sistemas de información empresarial 8
 - 1.2.3 El área de control de gestión de la empresa y las bases de datos 8
 - 1.3 Los Sistemas de Gestión de Bases de Datos (SGBS) como alternativa a los ficheros de almacén de datos. Nociones básicas..... 9
 - 1.4 Bases de Datos (Hoja de Cálculo) vs SGBD (Access, SAP, ORACLE..Etc.).....10
 - 1.5 Elementos que compone un Sistema de Gestión de Bases de Datos (SGBD o DBMS) 10
 - 1.6 Estructura de una base de datos: Estructura lógica y física. Independencia de los datos 11
- 2 Arquitectura de los SGBD.....13
 - 2.1 Introducción13
 - 2.2 Los Modelos de Datos, distintas formas de representar la misma información13
 - 2.3 Modelos conceptuales, genérico modelo E-R.....13
 - 2.4 Modelos lógicos14
 - 2.4.1 El modelo en red14
 - 2.4.2 El modelo jerárquico15
 - 2.4.3 El modelo relacional15

2.4.4	<i>Modelo orientados a objetos</i>	16
2.4.5	<i>Modelo objeto relacionales</i>	16
2.5	<i>Modelos físicos</i>	16
3	Objetos de un SGBD. Referencia al Access	17
3.1	<i>Introducción</i>	17
3.2	<i>Componentes básicos</i>	17
3.2.1	<i>Esquema general</i>	17
3.2.2	<i>El modelo relacional. Relación de tablas asociadas por campos comunes</i>	18
3.3	<i>Conceptos fundamentales: Tabla y sus elementos. Campo y Registro</i>	19
3.3.1	<i>Las tablas y los registros aspectos generales</i>	19
3.3.2	<i>Campo, estructura del registro y registro de datos (o registro)</i>	19
3.3.3	<i>Campo índice</i>	20
3.3.4	<i>Campo Clave Principal</i>	20
3.4	<i>Consultas</i>	20
3.5	<i>Formularios</i>	20
3.6	<i>Informes</i>	21
3.7	<i>Macros</i>	21
3.8	<i>Módulos</i>	21
4	Lenguajes e interfaces de los SGBD	22
4.1	<i>Comunicación entre el usuario y el SGBD</i>	22
4.2	<i>Lenguaje de definición de datos</i>	22
4.3	<i>Lenguaje de manipulación de datos, DML- SQL</i>	22
4.4	<i>Lenguajes de cuarta generación</i>	23
4.5	<i>Interfaces para SGBD</i>	23
5	Clasificación de los SGBD	24
6	Servicios que deben ofrecer los SGBD	25
7	Pasos previos a crear una base de datos. El estudio y diseño	28
7.1	<i>Consideraciones previas</i>	28
7.2	<i>Fases en el diseño de una BDR</i>	28
8	Caso de estudio y reflexión. Estructura básica de un programa Contable – Financiero (I)	30
9	Bibliografía	33

1 La gestión de datos empresariales, aspectos generales

1.1 De los archivos como almacén de datos a las bases de datos

1.1.1 Introducción

Todas las empresas requieren almacenar información. Desde siempre lo han hecho. La información puede ser de todo tipo. Cada elemento informativo (nombre, dirección, sueldo, etc.) es lo que se conoce como dato (en inglés data).

Las soluciones utilizadas por las empresas para almacenar los datos son diversas. Antes de la aparición de la informática se almacenaban en ficheros con cajones y carpetas y fichas. Tras la aparición de la informática estos datos se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador (a veces en archivos binarios, o en hojas de cálculo, ...).

Además las empresas requieren utilizar aplicaciones informáticas para realizar tareas propias de la empresa a fin de mecanizar a las mismas. Estas aplicaciones requieren manejar los datos de la empresa.

En los inicios de la era informática, cada programa almacenaba y utilizaba sus propios datos de forma un tanto caótica. La ventaja de este sistema (la única ventaja), es que los procesos eran independientes por lo que la modificación de uno no afectaba al resto. Pero tiene grandes inconvenientes:

- Coste de almacenamiento elevado
- Datos redundantes (se repiten continuamente)
- Probabilidad alta de inconsistencia en los datos
- Difícil modificación en los datos y facilidad de problemas de inconsistencia al realizar esas modificaciones (ya que es difícil que esa modificación afecte a todos los datos)

Lógicamente la solución a este problema es hacer que todas las aplicaciones utilicen los mismos datos. Esto provoca que los datos deban estar mucho más protegidos y controlados. Además los datos forman una estructura física y funcional que es lo que se conoce como base de datos.

De esta forma una base de datos es una serie de datos relacionados que forman una estructura lógica, es decir una estructura reconocible desde un programa informático.

Esa estructura no sólo contiene los datos en sí, sino la forma en la que se relacionan. Las bases de datos empiezan a aparecer en los años 60 y triunfan en los años setenta y ochenta.

Por tanto y como resumen podemos señalar que una base de datos es un archivo o conjunto de archivos que contienen múltiples informaciones que de alguna forma guardan relación. Por ej. Una base de datos para gestionar un Videoclub almacenará múltiples informaciones sobre películas, socios, etc... y entre ellas existirán relaciones como por ej. una película con un socio por medio de un préstamo o alquiler.

Otro ejemplo es la gestión de fichas de los artículos de nuestro almacén, introduciendo, modificando, actualizando, sacando informes por impresora, etc. Podemos por ejemplo realizar consultas tales como ¿qué artículo se ha vendido más este mes? ¿qué clientes compran un determinado artículo? ¿cuál es la ganancia total del mes? etc.

En definitiva una base de datos es un archivo que contiene datos (estructurados e interrelacionados) y los objetos que definen y manejan esos datos: tablas, consultas, formularios, informes, macros y módulos.

Históricamente, las primeras aplicaciones de ordenadores se concentraron en tareas de oficina: procesamiento de entradas o procesamiento de pedidos, o ambos, nóminas, planificación del trabajo, etc. Tales aplicaciones tenían acceso a datos guardados en archivos de ordenador. Se solicitaba información (¿cuántos productos se vendieron, por quién y a quién?); y se generaban informes para transformar los datos en información útil para las decisiones gerenciales.

Por tanto el concepto de sistema de archivos es fundamental en el área empresarial en cuanto que estos son los almacenes de información. Los sistemas de archivos surgieron al tratar de informatizar el manejo de los archivadores manuales con objeto de proporcionar un acceso más eficiente a los datos.

Un sistema de archivos es un conjunto de programas que prestan servicio a los usuarios finales. Cada programa define y maneja sus propios datos.

1.1.2 Problemas generales de los sistemas de archivos antiguos, ejemplo de bd con Excel. Caso empresa Inmobiliaria

Para comprender esto vamos a utilizar como ejemplo una empresa inmobiliaria.

En esta inmobiliaria, el departamento de ventas se encarga de alquilar inmuebles. Por ejemplo, cuando un propietario pasa por el departamento de ventas para ofrecer en alquiler su piso, se rellena un formulario en donde se recogen los datos del piso, como la dirección y el número de habitaciones, y los datos del propietario. El departamento de ventas también se encarga de atender a los clientes que desean alquilar un inmueble. Cuando un cliente (posible inquilino) pasa por este departamento se rellena un formulario con sus datos y sus preferencias: si quiere un piso o una casa, el importe mensual que está dispuesto a pagar por el alquiler, etc. Para gestionar toda esta información, el departamento de ventas posee un sistema de información. El sistema tiene tres archivos: archivo de inmuebles, archivo de propietarios y archivo de inquilinos.

INMUEBLE							
Inum	Calle	Area	Población	Tipo	Hab	Alquiler	Pnum
IA14	Enmedio, 128	Centro	Castellón	Casa	6	600	P46
IL94	Riu Ebre, 24	Ronda Sur	Castellón	Piso	4	350	P87
IG4	Sorell, 5	Grao	Castellón	Piso	3	300	P40
IG36	Alicante,1		Segorbe	Piso	3	325	P93
IG21	San Francisco, 10		Vinaroz	Casa	5	550	P87
IG16	Capuchinos, 19	Rafalafena	Castellón	Piso	4	400	P93

PROPIETARIO					
Pnum	Nombre	Apellido	Dirección	Pref	Teléfono
P46	Amparo	Felip	Asensi 24, Castellón	964	230 680
P87	Manuel	Obiol	Av.Libertad 15, Vinaroz	964	450 760
P40	Alberto	Estrada	Av.del Puerto 52, Castellón	964	200 740
P93	Yolanda	Robles	Purísima 4, Segorbe	964	710 430

INQUILINO							
Qnum	Nombre	Apellido	Dirección	Pref	Teléfono	Tipo	Alquiler
Q76	Juan	Felip	Barceló 47, Castellón	964	282 540	Piso	375
Q56	Ana	Grangel	San Rafael 45, Almazora	964	551 110	Piso	300
Q74	Elena	Abaso	Navarra 76, Castellón	964	205 560	Casa	700
Q62	Alicia	Mori	Alloza 45, Castellón	964	229 580	Piso	550

El departamento de contratos se ocupa de gestionar los contratos de alquiler de los inmuebles. Cuando un cliente desea formalizar un contrato, un empleado de la empresa rellena un formulario con los datos del inquilino y los datos del inmueble. Este formulario se pasa al departamento de contratos, que asigna un número al contrato y completa la información sobre el pago y el período del contrato. Para gestionar esta información, el departamento de contratos posee un sistema de información con tres archivos: el archivo de los contratos, el archivo de los inmuebles alquilados y el archivo de los inquilinos que tienen en vigor un contrato de alquiler.

CONTRATO							
Cnum	Inum	Qnum	Importe	Pago	Inicio	Fin	Meses
10024	IA14	Q62	600	Visa	01/06/1999	31/05/2000	12
10075	IL94	Q76	350	Efectivo	01/01/2000	30/06/2000	6
10012	IG21	Q74	550	Cheque	01/07/1999	30/06/2000	12

INMUEBLE				
Inum	Calle	Area	Población	Alquiler
IA14	Enmedio, 128	Centro	Castellón	600
IL94	Riu Ebre, 24	Ronda Sur	Castellón	350
IG21	San Francisco, 10		Vinaroz	550

INQUILINO					
Qnum	Nombre	Apellido	Dirección	Población	Teléfono
Q76	Juan	Felip	Barceló, 47	Castellón	964 282 540
Q74	Elena	Abaso	Navarra, 76	Castellón	964 205 560
Q62	Alicia	Mori	Alloza, 45	Castellón	964 229 580

Cada departamento accede a sus propios archivos mediante una serie de programas de aplicación escritos especialmente para ellos. Estos programas son totalmente independientes entre un departamento y otro, y se utilizan para introducir datos, mantener los archivos y generar los informes que cada departamento necesita. Es importante destacar que la estructura física de los archivos de datos y de sus registros está definida dentro de los programas de aplicación.

La situación es muy similar en el resto de departamentos. En el departamento de nóminas tienen un archivo con los datos de los salarios de los empleados. Los registros de este

archivo tienen los siguientes campos: número de empleado, nombre, apellido, dirección, fecha de nacimiento, salario, DNI y número de la oficina en la que trabaja. El departamento de personal tiene un archivo con los datos de los empleados. Sus registros tienen los siguientes campos: número de empleado, nombre, apellidos, dirección, teléfono, puesto, fecha de nacimiento, salario, DNI y número de la oficina en la que trabaja.

Se puede ver claramente que hay una gran cantidad de datos repetidos en los archivos de estos departamentos, algo que siempre ocurre en los sistemas de archivos. A raíz de esto, los sistemas de archivos presentan una serie de inconvenientes:

- ***Separación y aislamiento de los datos.*** Cuando los datos se separan en distintos archivos, es más complicado acceder a ellos, ya que el programador de aplicaciones debe sincronizar el procesamiento de los distintos archivos implicados para asegurar que se extraen los datos correctos.
- ***Duplicación de datos.*** La redundancia de datos existente en los sistemas de archivos hace que se desperdicie espacio de almacenamiento y lo que es más importante: puede llevar a que se pierda la consistencia de los datos. Se produce una inconsistencia cuando copias de los mismos datos no coinciden.
- ***Dependencia de datos.*** Ya que la estructura física de los datos (la definición de los archivos y de los registros) se encuentra codificada en los programas de aplicación, cualquier cambio en dicha estructura es difícil de realizar. El programador debe identificar todos los programas afectados por este cambio, modificarlos y volverlos a probar, lo que cuesta mucho tiempo y está sujeto a que se produzcan errores. A este problema, tan característico de los sistemas de archivos, se le denomina también falta de independencia de datos lógica-física.
- ***Formatos de archivos incompatibles.*** Ya que la estructura de los archivos se define en los programas de aplicación, es completamente dependiente del lenguaje de programación utilizado para desarrollar dichos programas. La incompatibilidad entre archivos generados por distintos lenguajes hace que los archivos sean difíciles de procesar de modo conjunto.
- ***Consultas fijas y proliferación de programas de aplicación.*** Desde el punto de vista de los usuarios finales, los sistemas de archivos fueron un gran avance comparados a los sistemas manuales. A consecuencia de esto, creció la necesidad de realizar distintos tipos de consultas de datos. Sin embargo, los sistemas de archivos son muy dependientes del programador de aplicaciones: cualquier consulta o informe que se quiera realizar debe ser programado por él. En algunas organizaciones se conformaron con fijar el tipo de consultas e informes, siendo imposible realizar otro tipo de consultas que no se hubieran tenido en cuenta a la hora de escribir los programas de aplicación. En otras organizaciones hubo una proliferación de programas de aplicación para resolver todo tipo de consultas, hasta el punto de desbordar al departamento de proceso de datos, que no daba abasto para validar, mantener y documentar dichos programas.

1.2 Las bases de datos y la empresa, aspectos generales

1.2.1 La relación entre las distintas áreas de la empresa y la necesidad del registro de los flujos de información

Como hemos expuesto en todas las áreas fundamentales de la empresa se desarrollan múltiples actividades que en la mayoría de ocasiones se encuentran interrelacionadas, así por ejemplo supongamos la operación cotidiana como puede ser la recepción de la mercancía por parte de un proveedor con su correspondiente factura o albarán. Esta simple operación origina una serie de actividades o tareas que afectan a varias funciones de la empresa como puede ser el área de finanzas con el reconocimiento de la deuda o pago y con el área comercial con la necesidad de actualización y control de inventarios.

De forma esquemática esta interrelación entre las diferentes áreas de la empresa podría ser resumida con el siguiente esquema:

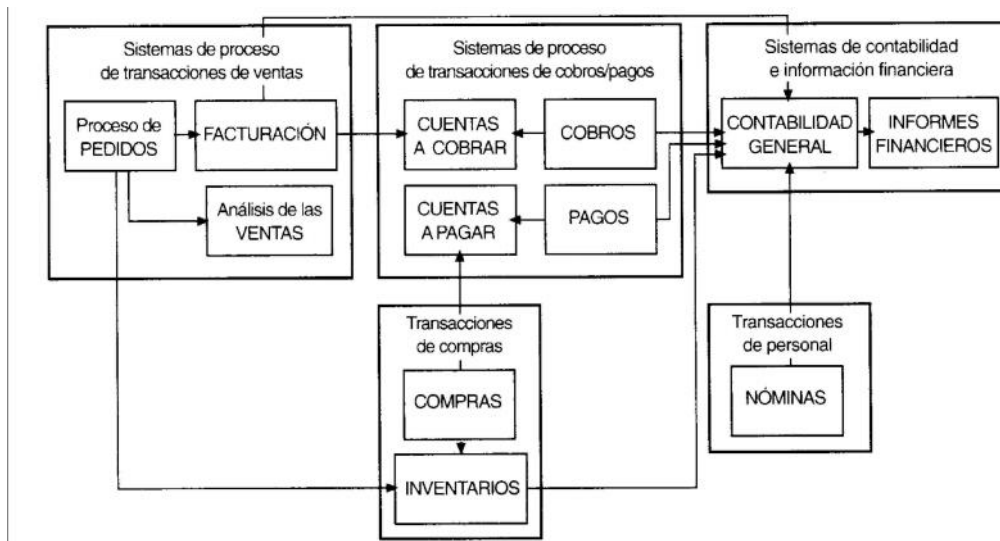


Figura 1.2. Sistemas informáticos de gestión (adaptado de O'Brien, 1993).

El volumen de información generada en la empresa conjuntamente con la complejidad de las interrelaciones existentes entre las diferentes áreas empresariales exige contar con un sistemas de información correctamente diseñados.

Estos sistemas tienen como objetivo principal el coordinar los flujos y los registros de información para desarrollar las actividades de acuerdo a las necesidades de cada empresa y en especial para la dirección, toma de decisiones y control de la actividad de la misma. Además

Un papel fundamental también de estos sistemas de información es el hecho de que constituyen instrumentos de control propiamente dicho (presupuestos, contabilidad, etc..) y consta al menos de las siguientes tareas:

- Emitir información detallada de cada unidad para que esta compruebe su cumplimiento de objetivos.
- Emitir informes que analicen y comenten los hechos más relevantes del negocio, destacando desviaciones significativas. El destinatario de estos informes es tanto la dirección general como el responsable de cada una de las unidades.

La información relevante ha de pasar a formar parte de los cuadros de mandos de cada uno de los responsables, empezando por la dirección general. Estos cuadros de mando han de responder a las variables claves del negocio y de ahí que, a diferencia del clásico sistema de ratios financieros, sean propios de cada sector de actividad e incluso de cada empresa. De hecho en una empresa diversificada en diversos sectores los cuadros de mando difieren según el sector de cada filial o unidad para adaptarse al modelo de negocio. Esta diversidad trae consigo una gran variedad de datos a usar para control de gestión, que se alejan de la medida monetaria única de los cuadros financieros.

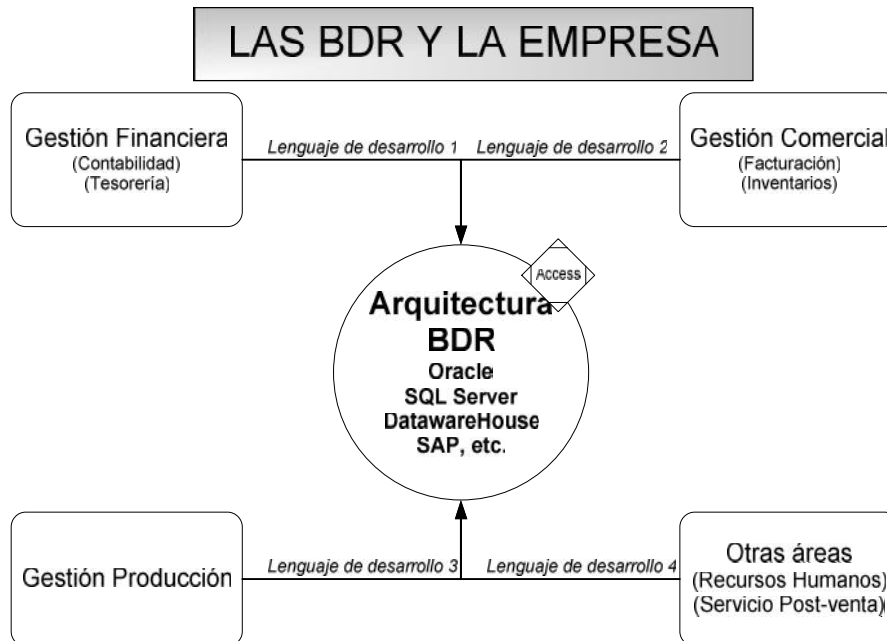
Según un estudio (Arthur Andersen, 1996), la mayor parte de las empresas españolas basan su control de gestión en hojas de cálculo. Si se acepta esta como la situación ideal, bastaría con remitir al lector a los manuales de los productos estrella, Excel, Lotus, etc.

Sin ánimo de minusvalorar este magnífico producto, hay que citar otros instrumentos útiles en el control de gestión, en este caso el de las BDR.

1.2.2 El papel de las BDR en los sistemas de información empresarial

Las bdr constituye una herramienta fundamental en el establecimiento e implantación de los sistemas de información empresarial ya que constituye un instrumento fundamental sobre los que se basan los actuales sistemas de información, es decir el actual software orientado a la gestión empresarial, independientemente del lenguaje de desarrollo, usan como arquitectura las bdr.

El uso de este tipo de arquitectura nos permite no solo el diseño de los sistemas de información de cada área empresarial sino además la interrelación y comunicación entre las distintas áreas y sistemas de información empresarial, de ahí la importancia y estudio de este tipo de arquitectura.



Un gran número de empresas disponen de sistemas de información desarrollados en distintos lenguajes de programación e incluso uno para cada área empresarial pero la característica común es el uso de una arquitectura de BDR lo que garantiza la posible comunicación e interrelación, aun cuando, como hemos dicho los mismos estén basados en diferentes lenguajes, ya que han sido adquiridos a diferentes desarrolladores de soluciones (proveedores) y/o diseñados exclusivamente para la empresa (desarrollo propio)

1.2.3 El área de control de gestión de la empresa y las bases de datos

A diferencia de otros departamentos, el control de gestión es demandante de datos contenidos en otros sistemas informáticos de la empresa, de ahí que para el control de gestión sea de gran interés el uso de lo que se ha venido en llamar "almacén de datos".

En la práctica, la forma de almacenar datos más importante y más extendida ha sido la construcción de bases de datos relacionales (BDR). Comercialmente son innumerables los sistemas de BDR que existen pudiendo citar: Oracle, SyBase, Informix, SQL Server, Dbase, FoxPro, Lotus Approach, Microsoft Access.

1.3 *Los Sistemas de Gestión de Bases de Datos (SGBS) como alternativa a los ficheros de almacén de datos. Nociones básicas.*

Estos sistemas surgieron como alternativa a los sistemas de archivos y para trabajar de un modo más efectivo, surgieron las bases de datos y los sistemas gestores de bases de datos (SGBD).

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y que ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción.

La base de datos es un gran almacén de datos que se define una sola vez y que se utiliza al mismo tiempo por muchos departamentos y usuarios. En lugar de trabajar con archivos desconectados e información redundante, todos los datos se integran con una mínima cantidad de duplicidad. La base de datos no pertenece a un departamento, se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción, que se denomina metadatos, se almacena en el diccionario de datos o catálogo y es lo que permite que exista independencia de datos lógica-física.

El SGBD es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos. En general, un SGBD proporciona los siguientes servicios:

- **Permite la definición de la base de datos mediante el lenguaje de definición de datos**. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos. Todo esto se almacenará en la base de datos.
- **Permite la inserción, actualización, eliminación y consulta de datos mediante el lenguaje de manejo de datos**. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de archivos, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de un gran número de programas de aplicación costosos de gestionar.
- **El lenguaje estándar utilizado por los SGBDR es el (Structured Query Language)**
- **Proporciona un acceso controlado a la base de datos mediante:**
 1. Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.
 2. Un sistema de integridad que mantiene la integridad y la consistencia de los datos.
 3. Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.
 4. Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del hardware o del software;
 5. Un diccionario de datos o catálogo accesible por el usuario que contiene la descripción de los datos de la base de datos.

Todos los SGBD no presentan la misma funcionalidad, depende de cada producto. En general, los grandes SGBD multiusuario ofrecen todas las funciones que se acaban de citar y muchas más. Los sistemas modernos son conjuntos de programas extremadamente complejos y sofisticados, con millones de líneas de código y con una documentación consistente en varios volúmenes. Lo que se pretende es proporcionar un sistema que permita gestionar cualquier tipo de requisitos y que tenga un 100% de fiabilidad ante cualquier fallo hardware o software. Los SGBD están en continua evolución, tratando de satisfacer los requerimientos de todo tipo de usuarios. Por ejemplo, muchas aplicaciones de hoy en día necesitan almacenar imágenes, vídeo, sonido, etc. Para satisfacer a este mercado, los SGBD deben cambiar. Conforme vaya pasando el tiempo irán surgiendo nuevos requisitos, por lo que los SGBD nunca permanecerán estáticos.

1.4 Bases de Datos (Hoja de Cálculo) vs SGBD (Access, SAP, ORACLE..Etc.)

Una base de datos no es más que un conjunto de datos relacionados entre sí, creados para un propósito específico.

Nombre	Teléfono	Dirección
Juan	223344	Paseo de la Esperanza nº7
Luisa	224455	Calle de la Pereza nº5

Este ejemplo constituye una base de datos de un listín telefónico. Lo más importante es que los datos han de estar organizados de una forma lógica, no de una forma aleatoria. Es decir, consideramos que no todas las colecciones de datos son bases de datos.

En esta línea podemos señalar también que las tareas tradicionales asociadas a las bases de datos es la introducción y actualización de información, y la realización de consultas. Como ejemplos, podemos citar sistemas bancarios, sistemas de reserva de vuelos y habitaciones de hoteles, catálogos de bibliotecas o sistemas de gestión de supermercados. A este conjunto de tareas se han sumado otras favorecidas por el avance tecnológico como son el manejo de sonidos y gráficos, como bases de datos multimedia o sistemas de información geográfica (GIS), etc.

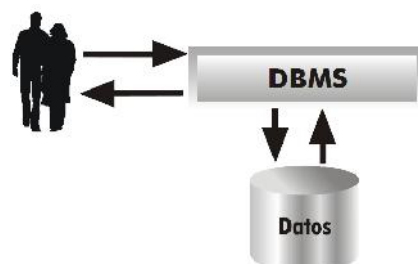
El tamaño y la complejidad de la bases de datos depende del problema que se esté resolviendo, por lo que podemos tener una base de datos personal para guardar las direcciones y teléfonos de nuestros mejores amigos, o bien tener una base de datos que almacene todos los datos clínicos de los usuarios de la Seguridad Social. Por tanto, tal cantidad de información, tiene que organizarse y controlarse para que la información pueda ser accedida y manipulada cuando sea necesario. Además, su almacenamiento ha de ser eficiente, así como las operaciones de manipulación que se llevan a cabo sobre los datos.

Un SGBD (Sistema de Gestión de Bases de Datos) es un conjunto de programas (en realidad, un sistema software) de propósito general que facilita el proceso de definición, construcción y manipulación de bases de datos para usos diversos.

1.5 Elementos que compone un Sistema de Gestión de Bases de Datos (SGBD o DBMS)

Para construir una base de datos en soporte informático con todas las informaciones a almacenar (estructuras de datos, tablas, índices, etc...), es necesario disponer de una herramienta que lo permita. Este es el cometido de un SGBD; básicamente permite crear, manipular gestionar y eliminar tanto los datos como las estructuras de una base de datos, permitiendo de esta forma el tratamiento automatizado y fácil de la información almacenada en ella. No se debe confundir base de datos con SGBD con Base de Datos, el primero es una herramienta para la creación mientras que la segunda es una solución concreta a un problema de almacenamiento de datos determinado. Por ej. Con Access (es un SGBD) podemos crear y gestionar múltiples bases de datos (para gestionar un videoclub, una biblioteca, un comercio, etc...) con diferentes necesidades de almacenamiento.

Un sistema gestor de bases de datos o SGBD (aunque se suele utilizar más a menudo las siglas DBMS procedentes del inglés, Data Base Management System) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.



Los elementos que componen un SGDB o DBMS son básicamente:

- Hardware. Máquinas en las que se almacenan las bases de datos. Incorporan unidades de almacenamiento masivo para este fin.
- Software. Es el sistema gestor de bases de datos. El encargado de administrar las bases de datos.
- Datos. Incluyen los datos que se necesitan almacenar y los metadatos que son datos que sirven para describir lo que se almacena en la base de datos.
- Usuarios. Personas que manipulan los datos del sistema. Hay tres categorías:
 - Usuarios finales. Aquellos que utilizan datos de la base de datos para su trabajo cotidiano que no tiene por qué tener que ver con la informática. Normalmente no utilizan la base de datos directamente, sino que utilizan aplicaciones creadas para ellos a fin de facilitar la manipulación de los datos. Estos usuarios sólo acceden a ciertos datos.
 - Desarrolladores. Analistas y programadores encargados de generar aplicaciones para los usuarios finales.
 - Administradores. También llamados DBA (*Data Base Administrator*), se encargan de gestionar las bases de datos.

Hay que tener en cuenta que las necesidades de los usuarios son muy diferentes en función del tipo de usuario que sean: a los finales les interesa la facilidad de uso, a los desarrolladores la potencia y flexibilidad de los lenguajes incorporados del sistema de bases de datos, a los administradores herramientas de gestión avanzada para la base de datos.

1.6 Estructura de una base de datos: Estructura lógica y física. Independencia de los datos

Las bases de datos están compuestas, de datos y de metadatos. Los metadatos son datos (valga la redundancia) que sirven para especificar la estructura de la base de datos; por ejemplo qué tipo de datos se almacenan (si son texto o números o fechas ...), qué nombre se le da a cada dato (nombre, apellidos,...), cómo están agrupados, cómo se relacionan,....

De este modo se producen dos visiones de la base de datos:

- Estructura lógica. Indica la composición y distribución teórica de la base de datos. La estructura lógica sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando. Es una estructura que permite idealizar a la base de datos. Sus elementos son objetos, entidades, nodos, relaciones, enlaces,... que realmente no tienen presencia real en la física del sistema. Por ello para acceder a los datos tiene que haber una posibilidad de traducir la estructura lógica en la estructura física.
- Estructura física. Es la estructura de los datos tan cual se almacenan en las unidades de disco. La correspondencia entre la estructura lógica y la física se almacena en la base de datos (en los metadatos).

Es la capacidad de modificar un esquema de un nivel sin afectar a los esquemas de nivel superior. Hay dos niveles de independencia de datos:

- Independencia física de datos. Es la capacidad de modificar el esquema físico sin necesidad de modificar los programas de aplicación. Esto es, por ejemplo, modificar el tamaño de un campo o modificar los índices del archivo.

- Independencia lógica de datos . Es la capacidad de modificar el esquema conceptual sin necesidad de modificar los programas de aplicación. Si, por ejemplo, se reduce la base de datos eliminando un tipo de datos, los esquemas externos que no se refieran a esta parte de la base de datos no deberán verse afectados. La independencia lógica es más difícil de conseguir que la independencia física, ya que los programas de aplicación suelen ser muy dependientes de la estructura lógica de los datos a los que acceden (se han creado a partir de ella).

2 Arquitectura de los SGBD

2.1 Introducción

Cuando un arquitecto diseña una casa, no empieza decidiendo el tipo de grifería con la que equipará los baños de la misma ni el origen de la madera de las puertas. En su actuación primeramente se ocupará de ciertas características mucho más generales de la casa, sin preocuparse de pequeños aspectos como los mencionados inicialmente, es decir, se abstraerá de los detalles. Una vez que estas consideraciones generales hayan sido resueltas irá resolviendo otros problemas, siempre yendo de lo general a lo específico. Como a continuación se explica, los SGBD se estructuran de una manera similar.

Un objetivo importante de los SGBD es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema debe ocultar los detalles sobre cómo se acceden y se manipulan los datos.

2.2 Los Modelos de Datos, distintas formas de representar la misma información

Una de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características del almacenamiento físico (características que la mayoría de usuarios no necesita conocer). Los modelos de datos son el instrumento principal que utilizan para ofrecer dicha abstracción. En esta sección se explica el concepto de modelo de datos y se mencionan los más utilizados por los SGBD.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos.

Los modelos de datos se pueden clasificar en:

- Modelos de datos de alto nivel, o modelos conceptuales, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos.
- Modelos lógicos, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente. Los modelos lógicos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implementarse de manera directa en un ordenador.
- Modelos de datos de bajo nivel, o modelos físicos, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador. Los conceptos de los modelos físicos están dirigidos al personal informático, no a los usuarios finales.

2.3 Modelos conceptuales, genérico modelo E-R

Los modelos conceptuales se usan para describir los datos y se caracterizan porque proporcionan una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. A continuación describiremos brevemente el modelo más importante dentro de esta categoría, el modelo entidad-relación.

El modelo entidad-relación

El modelo de datos E-R (entidad-relación) se basa en una percepción del mundo real que consiste en una colección de objetos básicos denominados entidades y las relaciones entre estos objetos.

Una **entidad** es un objeto que se diferencia de otros objetos mediante una serie de atributos. Por ejemplo, los atributos númeroDeCuenta y saldo describen una cuenta concreta de un banco. De la misma manera, los atributos nombre y dni describen un cliente del banco.

Una relación es una asociación entre varias entidades. Por ejemplo, una relación clienteCuenta asocia a cada cliente todas las cuentas que tiene en el banco.

La estructura lógica global de una base de datos puede representarse gráficamente mediante un diagrama E-R, que consta de los siguientes componentes:

- Rectángulos : Representan entidades
- Elipses : Representan atributos.
- Rombos : Representan relaciones entre entidades.
- Líneas : Conectan atributos con entidades y entidades con relaciones.

Cada una de estas componentes se etiqueta con la entidad, relación o atributo que representa, de forma que los diagramas obtenidos tienen un aspecto como el de la figura siguiente.



En líneas generales podemos entender por entidad como todo aquello sobre lo que es necesario almacenar información en una base de datos. Por ej. En una base de datos de una biblioteca entidades serían LIBROS, SOCIOS, etc....

Entre las diferentes entidades de una base de datos suele haber **relaciones**, las cuales permiten un tratamiento más globalizado de la información y reflejan situaciones del mundo real. Por ej. En un videoclub hay dos entidades claras PELICULAS y SOCIOS. Entre ellas existe una relación bastante evidente que es el préstamo o alquiler. De esta forma las entidades y relaciones en una base de datos se representan de forma lógica con estructuras del tipo:

(Entidad) → (Relación) → (Entidad)
SOCIO → ALQUILA → PELICULA

Tanto el Socio como el alquiler y la película, tienen sus informaciones asociadas que serán las que se almacenen en la base de datos. Informaciones podrían ser para SOCIO (nº socio, Nombre, Apellidos, Tfno.), para ALQUILER (fecha, nº de cinta, importe, nº de socio) y de PELICULA (nº de cinta, Título, Director, Genero).

Estas relaciones no necesariamente siempre tienen que constituirse en tabla, a veces se integran dentro de las propias entidades en función de los datos que contengan.

2.4 Modelos lógicos

Los tres modelos lógicos más ampliamente extendidos son el modelo relacional, el modelo en red y el modelo jerárquico.

2.4.1 El modelo en red

Se trata de un modelo que se utilizó durante mucho tiempo. Organiza la información en registros y enlaces. Los registros representan las entidades del modelo entidad / relación. En los registros se almacenan los datos utilizando atributos. Los enlaces permiten relacionar los registros de la base de datos.

El modelo en red más aceptado es el llamado codasyl, que durante mucho tiempo se ha convertido en un estándar.

Las bases de datos en red son parecidas a las jerárquicas sólo que en ellas puede haber más de un padre. En este modelo se pueden representar perfectamente relaciones varios a

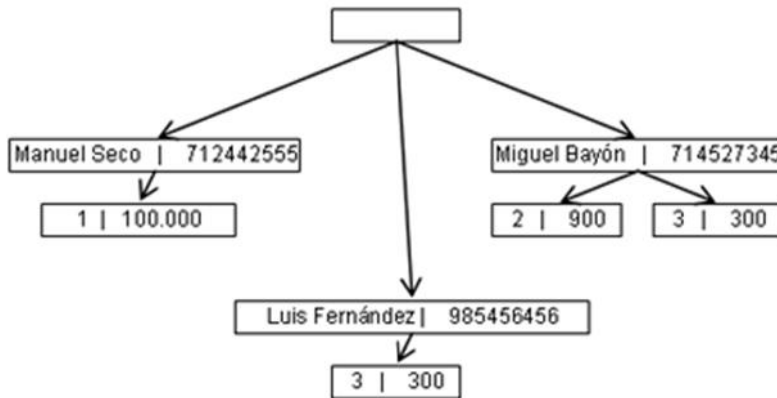
varios. Pero su dificultad de manejo y complejidad hace que se estén abandonando completamente.

Se trata de un modelo en el que los datos se representan mediante conjuntos de registros y las relaciones mediante enlaces, los cuales pueden ser vistos como punteros.



2.4.2 El modelo jerárquico

El modelo jerárquico es similar al modelo en red, en el sentido en que los datos y las relaciones entre los datos se representan mediante registros y enlaces. La diferencia con el modelo en red es que la organización de los registros se representa mediante estructuras jerárquicas (árboles), por lo que los nodos hijo, sólo pueden tener un padre, tal y como se muestra a continuación.



En ellas se organiza la información se organiza con un jerarquía en la que la relación entre las entidades de este modelo siempre es del tipo padre / hijo. De 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 tiene un padre).

2.4.3 El modelo relacional

Los datos se muestran en forma de tablas y relaciones. Este es el modelo más popular.

El modelo relacional representa los datos y las relaciones entre los datos mediante un conjunto de tablas. Cada una de las tablas tiene un nombre único y una serie de columnas también con nombres únicos. En el modelo relacional la base de datos es percibida por el usuario como un conjunto de tablas. Esta percepción es sólo a nivel lógico (en los niveles externo y conceptual de la arquitectura de tres niveles), ya que a nivel físico puede estar implementada mediante distintas estructuras de almacenamiento.

nombre	dni	saldo	numCuenta
Miguel Bayón	714527345	100.000	1
Manuel Seco	712442555	900	2
Luis Fernández	985456456	300	3

numCuenta	nombre
1	Luis Fernández
2	Miguel Bayón
3	Manuel Seco
3	Miguel Bayón

2.4.4 *Modelo orientados 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. En estos lenguajes los datos y los procedimientos se almacenan juntos. Esta es la idea de las bases de datos orientadas a objetos.

A través de esta idea se intenta que estas bases de datos consiguen arreglar las limitaciones de las relacionales. Por ejemplo el problema de la herencia, tipos definidos por el usuario, disparadores 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 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 cada vez hay más).

2.4.5 *Modelo objeto relacionales*

Tratan de ser un híbrido entre el modelo relacional y el orientado a objetos. Estas bases de datos se basan en el estándar SQL 99 que dictó las normas para estas bases de datos. En ese estándar se añade a las bases relacionales la posibilidad de almacenar procedimientos de usuario, triggers, tipos definidos por el usuario, consultas recursivas, bases de datos OLAP, tipos LOB,...

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

2.5 *Modelos físicos*

Los modelos físicos describen cómo se almacenan los datos en el ordenador: el formato de los registros, la estructura de los ficheros (desordenados, ordenados, etc.) y los métodos de acceso utilizados (índices, etc.).

3 Objetos de un SGBD. Referencia al Access

3.1 Introducción

Para organizar los datos de una base de datos es necesario crear estructuras que los contengan. Existe varios modelos de almacenamiento pero el más extendido es el modelo Relacional. Este modelo se caracteriza por almacenar organizar los datos en forma de tabla, esta es una forma habitual de organizar la información.

A continuación se muestran con un ejemplo, los aspectos básicos de una estructura tabla para el almacenamiento de datos.

Veamos el siguiente ejemplo:

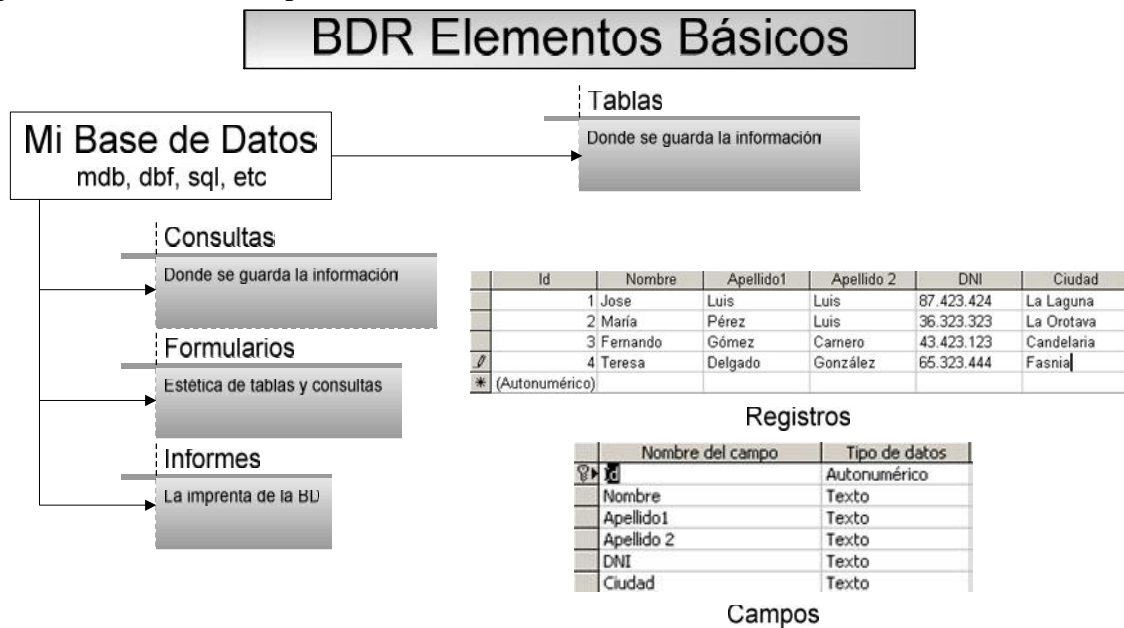
Tabla Agenda (Podemos considerar las entidades como tablas)

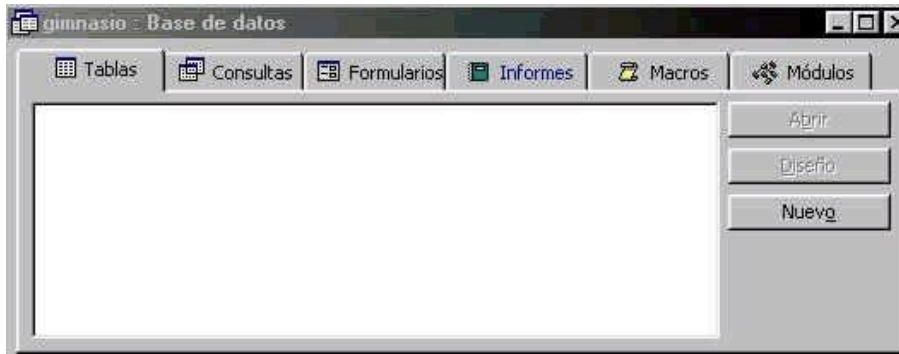
<i>Estructura del registro</i> →	<i>Nombre</i>	<i>Apellidos</i>	<i>Teléfono</i>
<i>Registro de datos</i> →	Antonio	García Ramos	956587412
<i>Registro de datos</i> →	María	Cantero López	956258974
	↑ <i>Campo nombre</i>	↑ <i>Campo Apellidos</i>	↑ <i>Campo Teléfono</i>

3.2 Componentes básicos

3.2.1 Esquema general

A continuación se detallan los objetos más importantes que se pueden crear y manipular en las bases de datos. Esta referencia es específica de MS-ACCESS aunque la mayor parte de ellos están disponibles en otros sistemas.





3.2.2 El modelo relacional. Relación de tablas asociadas por campos comunes

Una base de datos no tiene sentido si sus tablas no están asociadas para mostrar la información de una forma globalizada. Para asociar tablas hay que establecer relaciones (normalmente entre campos comunes). Por ej. supongamos que tenemos dos tablas en las que se almacenan datos sobre socios y prestamos de películas. En la tabla SOCIOS aparecerían todos los datos personales de los socios. En la tabla ALQUILER aparecerían todos los datos del alquiler de la película como por ej (fecha, importe, título).... Pero faltaría un campo que identificara el socio que la alquiló. Una solución podría ser poner en esta tabla el nombre del socio pero es posible que hubiese dos socios con nombre y apellidos iguales. En este caso lo mejor es colocar un campo llamado “socio” en la tabla SOCIOS donde se almacena el nº de socio para cada uno de ellos, por otro lado en la tabla ALQUILER colocaremos el campo “socio” para cada alquiler, reflejando así solamente el nº de socio que alquila. Gráficamente la estructura de las tablas sería la que se muestra a continuación:

SOCIO	NOMBRE	APELLIDOS	TFNO
1	Antonio	García	913565872
2	Maria	López	912598762
3	Carlos	Milla	912587931

SOCIO	FECHA	IMPORTE	TITULO
1	10/10/02	2	La His...
2	10/10/02	1	El Señ...
2	11/10/02	1	Seis...
1	12/10/02	2	Dos...

En el registro de alquileres se puede apreciar que Antonio García (socio nº 1) ha realizado dos alquileres uno el día 10 y otro el día 12. Hay que observar que la asociación de tablas se ha realizado mediante un campo común "SOCIO" que contiene el nº de socio para cada socio del videoclub. Este campo en la tabla SOCIOS es clave principal e identifica unívocamente a cada socio (No puede haber dos socios con el mismo nº) y en la tabla ALQUILER permite identificar al socio que alquila la película.

3.3 Conceptos fundamentales: Tabla y sus elementos. Campo y Registro.

3.3.1 Las tablas y los registros aspectos generales

Las tablas es la estructura que almacena todos los datos sobre un tipo de entidad o relación. Por ej. Clientes, Socios, Libros, Préstamos, etc...

Son las unidades básicas donde se almacena la información. Una base de datos puede contener múltiples tablas. Al usar una tabla independiente para cada tema, los datos se almacenan sólo una vez. Esto tiene como resultado una base de datos más eficaz y menos errores de entrada de datos. Las tablas organizan datos en columnas (denominadas campos) y filas (denominadas registros). Por ejemplo, cada campo de una tabla Productos contiene el mismo tipo de información para cada producto, por ejemplo, su nombre. Cada uno de los registros de esa tabla contiene toda la información acerca de un producto, por ejemplo, el nombre del producto, el Id. de proveedor, las unidades en existencia, etc

- **Tablas:** *Son los almacenes de datos. Se puede definir como la unidad donde crearemos el conjunto de datos de nuestra base de datos. Estos datos estarán ordenados en columnas verticales. Aquí definiremos los campos y sus características. El diseño de tablas consiste en la definición de los campos necesarios para almacenar correctamente la información de la base de datos.*

3.3.2 Campo, estructura del registro y registro de datos (o registro)

- Campo: unidad básica de una base de datos. Un campo puede ser, por ejemplo, el Nombre de una persona
- Registro: conjunto de campos. Un registro vendría a ser algo así como una ficha.
- Base de datos: conjunto de registros total.

Observemos el siguiente cuadro:

	Nombre del campo	Tipo de datos
	Código propiedad	Texto
	Nombre	Texto
	Apellidos	Texto
	Dirección	Texto
	Población	Texto
	DNI	Texto
	Fecha de contrato	Fecha/Hora
	Fecha de expiración	Fecha/Hora
	Alquiler mensual	Númérico
	Oficina vendedora	Texto

Este sería el primer registro de una base de datos compuesto por 4 campos: (nombre, apellido1, teléfono y provincia)

El número total de registros podemos verlo en la parte inferior (1 de 9)

Más concretamente tenemos por tanto que:

- Campo: cada uno de las informaciones que se almacenan sobre una entidad. Por ej. Edad, Teléfono, DNI, etc.... (columnas de la tabla).
- Estructura del Registro. Conjunto de campos que determinan que datos se almacenan sobre una entidad. Por ej. Nombre, Apellidos, Teléfono, etc.... (nombres de los campos que definen la estructura de la tabla).
- Registro de datos (o Registro). Cada uno de los conjuntos de datos almacenados sobre una ocurrencia de una entidad determinada. Por ej. Antonio, García Ramos,

956432178 son los datos que conforman el registro de Antonio. (filas de datos de la tabla).

3.3.3 *Campo índice*

Es un tipo especial de campo por el que se establece un orden para realiza búsquedas. La necesidad de estos campos es evidente. Si por ej. tenemos una tabla con 5000 registros de personas, resultará más fácil buscar a una persona por sus apellidos si la tabla está ordenada por este campo. Si no estuviera ordenada se podría realizar la búsqueda pero para un apellido como “Martín” en el peor de los casos tendríamos que buscar hasta el final de la tabla, mientras que con un orden (índice) sobre el campo apellidos, en el peor de los casos no pasaríamos de los registros que tuvieran el apellido “Martín” ya que mas allá no podría haber ninguna persona con ese apellido.

3.3.4 *Campo Clave Principal*

Se trata de un campo de la tabla al que se le define una propiedad especial para identificar unívocamente a una ocurrencia de la entidad. Supongamos que en nuestra tabla de 5000 registros deseamos localizar a una persona llamada Antonio García Fernández; ¿Existe la posibilidad de que dos personas tengan estos mismos nombres y apellidos?, la respuesta es Sí, de forma que si hubiese dos, no sabríamos a quien nos estaríamos refiriendo, con lo que tendríamos un problema. Para solucionarlo las bases de datos contienen campos que identifican a las entidades para evitar confusiones o dudas. Uno de los campos típicos para identificar a cada persona en una base de datos suele ser el DNI ya que este identifica unívocamente a una persona (uno a uno – dada una persona solo tiene un DNI; dado un DNI solo pertenece e identifica a una persona). Hay que destacar que en una tabla no puede haber dos ocurrencias de entidad con el mismo DNI (en una tabla de clientes no puede haber dos clientes con igual DNI; tampoco podrá registrarse dos veces al mismo cliente).

3.4 *Consultas*

Una consulta es una visión particular del conjunto de datos de la base de datos. En definitiva es un subconjunto de registros de una o varias tablas que cumplen una determinada condición. También se conocen como filtros. En cualquier caso responde a una necesidad del usuario. Por ej. Socios que hayan alquilado la película “El señor de los anillos” . Las consultas no contienen datos los obtienen de los datos almacenados en las tablas. Las consulta no tienen solo función de visualización pueden también realizar funciones de creación de tablas o de actualización de datos.

Se utilizan consultas para ver, modificar y analizar datos de formas diferentes. También pueden utilizarse como el origen de registros para formularios, informes y páginas de acceso a datos. Existen varios tipos de consultas, las más importantes son: Consulta de Selección, Consultas de parámetros y de acción.

- **Consultas:** *Es donde definiremos las operaciones que pueden realizarse sobre los datos de las tablas: limitaciones de acceso, selección de datos, inserción, modificación y borrado, cálculos...*

3.5 *Formularios*

Los formularios son herramientas que mejoran la presentación y manipulación de datos mediante ventanas similares a los programas de entorno Windows. El diseño lo realiza el usuario colocando controles y campos con el formato deseado.

- **Formulario:** *elemento en forma de ficha que permite la gestión de los datos de una forma más cómoda y visiblemente más atractiva, en definitiva son pantallas*

que se presentan a los usuarios de una base de datos para que tengan un acceso amigable a los datos y operaciones.

3.6 Informes

Los informes permiten preparar los datos para su listado en formato impreso con algunas funcionalidades adicionales como la de cálculos aritméticos (sumas, totales, cuenta de registros, etc...) o estadísticos (medias, varianzas, etc...).

- **Informe:** *permite preparar los registros de la base de datos de forma personalizada para imprimirlos.*

3.7 Macros

Son estructuras que permiten automatizar varias funciones que han de realizarse siempre bajo un único nombre de forma que al invocarlo se ejecutan todas las tareas. Es un paso previo a la programación pero sin necesidad de tantos conocimientos ya que algunas tareas están implementadas directamente en Access.

- **Macro:** *conjunto de instrucciones que se pueden almacenar para automatizar tareas repetitivas.*

3.8 Módulos

Los módulos es la parte de Access que necesita de mayores conocimientos debido a la necesidad de utilización de código de programación en Visual Basic.

- **Módulo:** *programa o conjunto de instrucciones en lenguaje Visual Basic*

4 Lenguajes e interfaces de los SGBD

4.1 Comunicación entre el usuario y el SGBD

Como los usuarios de un SGBD pueden tener distintos privilegios y distintos conocimientos informáticos, es necesario disponer de diferentes lenguajes e interfaces para cada tipo de usuarios. En esta sección vamos a ver los distintos lenguajes e interfaces que emplean los SGBD.

4.2 Lenguaje de definición de datos

Una vez que se ha finalizado la tarea de diseño de la base de datos, y que se ha seleccionado un SGBD para su implementación, el primer paso consiste en la especificación del esquema conceptual de la base de datos.

El esquema conceptual de la base de datos se especifica mediante una serie de definiciones expresadas en un Lenguaje de definición de datos (DDL, Data Definition Language). El SGBD contará con un compilador de DDL cuya función será procesar las sentencias en DDL para identificar las descripciones de los elementos de los esquemas y guardar la descripción del esquema en el diccionario de datos.

El diccionario de datos es un archivo que contiene metadatos, es decir, datos acerca de los datos. Este archivo se consulta cada vez que se leen o modifican los datos del Sistema de base de datos.

4.3 Lenguaje de manipulación de datos, DML- SQL

Una vez que se han compilado los esquemas de la base de datos, y que ya se han introducido datos en la base de datos, los usuarios necesitarán algún mecanismo para obtener información de la base de datos. Las operaciones más comunes de manipulación son la consulta, inserción, eliminación y modificación de datos. Para ello, todos los SGBD ofrecen un Lenguaje de manipulación de datos (DML, Data Manipulation Language).

En general existen dos tipos de DML:

- **Procedimentales.** Requieren que el usuario (normalmente será un programador) especifique qué datos desea y cómo hay que obtenerlos. Esto quiere decir que hay que especificar todas las operaciones de acceso a datos llamando a los procedimientos necesarios para obtener la información requerida. Estos lenguajes acceden a un registro, lo procesan y basándose en los resultados obtenidos, acceden a otro registro, que también deben procesar. Así se va accediendo a registros y se van procesando hasta que se obtienen los datos deseados. Las sentencias de un LMD procedimental deben estar embebidas en un lenguaje de alto nivel, ya que se necesitan sus estructuras (bucles, condicionales, etc.) para obtener y procesar cada registro individual. A este lenguaje se le denomina lenguaje anfitrión. Las bases de datos jerárquicas y de red utilizan LMD procedimentales.
- **No procedimentales.** Un LMD no procedimental se puede utilizar de manera independiente para especificar operaciones complejas sobre la base de datos de forma concisa. En muchos SGBD se pueden introducir interactivamente instrucciones del LMD desde un terminal o bien embeberlas en un lenguaje de programación de alto nivel. Los LMD no procedimentales permiten especificar los datos a obtener en una consulta o los datos que se deben actualizar, mediante una sola y sencilla sentencia. El usuario o programador especifica qué datos quiere obtener sin decir cómo se debe acceder a ellos. El SGBD traduce las sentencias del LMD en uno o varios procedimientos que manipulan los conjuntos de registros necesarios. Esto libera al usuario de tener que conocer cuál es la estructura física de los datos y qué algoritmos se deben utilizar para acceder a ellos. Las bases de datos relacionales utilizan LMD no procedurales, como SQL (Structured Query Language)

o QBE (Query-By-Example). Los lenguajes no procedimentales son más fáciles de aprender y de usar que los procedimentales, y el usuario debe realizar menos trabajo, siendo el SGBD quien hace la mayor parte.

4.4 Lenguajes de cuarta generación

No existe consenso sobre lo que es un lenguaje de cuarta generación (4GL). Lo que en un lenguaje de tercera generación (3GL) como COBOL, FORTRAN, Pascal o C requiere cientos de líneas de código, tan solo necesita de diez o veinte líneas en un 4GL. Comparado con un 3GL, que es procedimental, un 4GL es un lenguaje no procedimental: el usuario define qué se debe hacer, no cómo debe hacerse. Los 4GL se apoyan en unas herramientas de mucho más alto nivel denominadas herramientas de cuarta generación . El usuario no debe definir los pasos a seguir en un programa para realizar una determinada tarea, tan sólo debe definir una serie de parámetros que estas herramientas utilizarán para generar un programa de aplicación. Se dice que los 4GL pueden mejorar la productividad de los programadores en un factor de 10, aunque se limita el tipo de problemas que pueden resolver.

Los lenguajes SQL y QBE son ejemplos de 4GL pero hay otro tipos de 4GL incluidos habitualmente en los SGBD:

- **Un generador de formularios** es una herramienta interactiva que permite crear rápidamente formularios de pantalla para introducir o visualizar datos. Los generadores de formularios permiten que el usuario defina el aspecto de la pantalla, qué información se debe visualizar y en qué lugar de la pantalla debe visualizarse.
- **Un generador de informes** es una herramienta para crear informes a partir de los datos almacenados en la base de datos. Se parece a un lenguaje de consultas en que permite al usuario hacer preguntas sobre la base de datos y obtener información de ella para un informe. Sin embargo, en el generador de informes se tiene un mayor control sobre el aspecto de la salida. Se puede dejar que el generador determine automáticamente el aspecto de la salida o se puede diseñar ésta para que tenga el aspecto que desee el usuario final.
- **Un generador de gráficos** es una herramienta para obtener datos de la base de datos y visualizarlos en un gráfico mostrando tendencias y relaciones entre datos. Normalmente se pueden diseñar distintos tipos de gráficos: barras, líneas, etc.

4.5 Interfaces para SGBD

Normalmente, mientras que los programadores utilizan el DML para la creación de consultas, los usuarios de un SGBD utilizan un lenguaje de consulta de alto nivel. Para la mayoría de los usuarios se suelen definir interfaces de usuario amigables para la interacción con la base de datos. A continuación vamos a ver los tipos de interfaces que hay:

- **Interfaces basadas en menús.** Presentan al usuario una lista de opciones en forma de menú que guían al usuario en la petición de consultas. De esta forma no es necesario conocer la sintaxis de un lenguaje de consulta, pues permiten la creación de la consulta eligiendo las opciones que presenta la interfaz.
- **Interfaces gráficas.** Suelen presentar al usuario los esquemas en forma de diagrama, y las consultas se especifican manipulando el diagrama con el ratón.
- **Interfaces basadas en formularios.** Estas interfaces presentan un formulario al usuario en el que se rellenan los huecos del formulario para la modificación de los datos, o bien para especificar los parámetros de la consulta.
- **Interfaces de lenguaje natural.** Estas interfaces aceptan la especificación de una consulta descrita en términos de un idioma concreto y construyen expresiones DML a partir de dicha especificación.

5 Clasificación de los SGBD

Distintos criterios para diferenciar los SGBD

Conocer las diferentes alternativas que ofrecen los SGBD es importante para realizar, en el caso de que nos tengamos que enfrentar a ello, una buena elección como solución informatizada para una organización u empresa. En esta sección se nombran los distintos tipos existentes junto a una breve explicación de los mismos.

El criterio principal que se utiliza para clasificar los SGBD es el modelo lógico en que se basan. El modelo lógico empleados con mayor frecuencia en los SGBD comerciales actuales es el relacional; los modelos de red y jerárquico se utilizaron mucho en productos más antiguos. Últimamente se está desarrollando productos que tienen como base otro modelo: el modelo orientado a objetos.

Un segundo criterio para clasificar los SGBD es el número de usuarios a los que da servicio el sistema. Los sistemas monousuario sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales. Los sistemas multiusuario, entre los que se encuentran la mayor parte de los SGBD, atienden a varios usuarios al mismo tiempo.

Un tercer criterio es el número de sitios en los que está distribuida la base de datos . Casi todos los SGBD son centralizados: sus datos se almacenan en un solo computador. Los SGBD centralizados pueden atender a varios usuarios, pero el SGBD y la base de datos en sí residen por completo en una sola máquina. En los SGBD distribuidos la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red. Los SGBD distribuidos homogéneos utilizan el mismo SGBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a varias bases de datos autónomas preexistentes almacenadas en SGBD distribuidos heterogéneos. Esto da lugar a los SGBD federados o sistemas multibase de datos en los que los SGBD participantes tienen cierto grado de autonomía local.

Un cuarto criterio es el coste del SGBD. La mayor parte de los paquetes de SGBD cuestan entre 10.000 y 100.000 euros. Los sistemas monousuario más económicos para microcomputadores cuestan entre 0 y 3.000 euros. En el otro extremo, los paquetes más completos cuestan más de 100.000 euros.

Por último, los SGBD pueden ser de propósito general o de propósito específico . Cuando el rendimiento es fundamental, se puede diseñar y construir un SGBD de propósito especial para una aplicación específica, y este sistema no sirve para otras aplicaciones. Muchos sistemas de reservas de líneas aéreas son SGBD de propósito especial y pertenecen a la categoría de sistemas de procesamiento de transacciones en línea (OLTP), que deben atender un gran número de transacciones concurrentes sin imponer excesivos retrasos.

6 Servicios que deben ofrecer los SGBD

Cuando una empresa u organización decide adquirir una licencia de un SGBD para su infraestructura informática antes ha debido pasar por una fase de análisis de sus necesidades y de revisión de las posibilidades que cada producto brinda. En una situación ideal, los SGBD deberían ofrecer las mismas funcionalidades, pero como todo en la vida, esto no sucede así.

Codd, el creador del modelo relacional, ha establecido una lista con los ocho servicios que debe ofrecer todo SGBD:

1. Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).
2. Un SGBD debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos). Normalmente, un diccionario de datos almacena:
 - a. Nombre, tipo y tamaño de los datos.
 - b. Nombre de las relaciones entre los datos.
 - c. Restricciones de integridad sobre los datos.
 - d. Nombre de los usuarios autorizados a acceder a la base de datos.
 - e. Esquemas externos, conceptual e interno, y correspondencia entre los esquemas.
 - f. Estadísticas de utilización.

Algunos de los beneficios que reporta el diccionario de datos son los siguientes:

- a. La información sobre los datos se puede almacenar de un modo centralizado por lo que las redundancias y las inconsistencias se pueden identificar más fácilmente.
 - b. El significado de los datos se puede definir, lo que ayudará a los usuarios a entender el propósito de los mismos.
 - c. El impacto que puede producir un cambio se puede determinar antes de que sea implementado, ya que el diccionario de datos mantiene información sobre cada tipo de dato, todas sus relaciones y todos sus usuarios.
 - d. Se puede hacer respetar la seguridad y la integridad.
 - e. Se puede proporcionar información para auditorías.
3. Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos. Una transacción en el sistema informático de la empresa inmobiliaria sería 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 base de datos. Si la transacción falla durante su realización, por ejemplo porque falla el hardware, la base de datos quedará en un estado inconsistente. Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.
4. Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los datos que comparten. El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está

actualizando datos, pueden interferir de modo que se produzcan inconsistencias en la base de datos. El SGBD se debe encargar de que estas interferencias no se produzcan en el acceso simultáneo.

5. Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente. Este fallo puede ser a causa de un fallo en algún dispositivo hardware o un error del software, que hagan que el SGBD aborte, o puede ser a causa de que el usuario detecte un error durante la transacción y la aborte antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.
6. Un SGBD debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.
7. Un SGBD debe ser capaz de integrarse con algún software de comunicación. Muchos usuarios acceden a la base de datos desde terminales. En ocasiones estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD. En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes y responde de modo similar. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él para que el sistema sea comercialmente viable.
8. Un SGBD debe proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar. Por ejemplo, se puede establecer la restricción de que cada empleado no puede tener asignados más de diez inmuebles. En este caso sería deseable que el SGBD controlara que no se sobrepase este límite cada vez que se asigne un inmueble a un empleado.

Además, de estos ocho servicios, es razonable esperar que los SGBD proporcionen un par de servicios más:

- Un SGBD debe permitir que se mantenga la independencia entre los programas y la estructura de la base de datos. La independencia de datos se alcanza mediante las vistas o subesquemas. La independencia de datos física es más fácil de alcanzar, de hecho hay varios tipos de cambios que se pueden realizar sobre la estructura física de la base de datos sin afectar a las vistas. Sin embargo, lograr una completa independencia de datos lógica es más difícil. Añadir una nueva entidad, un atributo o una relación puede ser sencillo, pero no es tan sencillo eliminarlos.
- Un SGBD debe proporcionar una serie de herramientas que permitan administrar la base de datos de modo efectivo. Algunas herramientas trabajan a nivel externo, por lo que habrán sido producidas por el administrador de la base de datos. Las herramientas que trabajan a nivel interno deben ser proporcionadas por el distribuidor del SGBD. Algunas de ellas son:
 - Herramientas para importar y exportar datos.
 - Herramientas para monitorizar el uso y el funcionamiento de la base de datos.
 - Programas de análisis estadístico para examinar las prestaciones o las estadísticas de utilización.

- Herramientas para reorganización de índices.
- Herramientas para aprovechar el espacio dejado en el almacenamiento físico por los registros borrados y que consoliden el espacio liberado para reutilizarlo cuando sea necesario.

7 Pasos previos a crear una base de datos. El estudio y diseño

7.1 Consideraciones previas

Antes de comenzar a crear una base de datos algo más compleja que la de nuestro primer ejemplo, convendría establecer un croquis o borrador en papel de los objetivos de la misma. Si es una base de datos para otra persona o empresa, es conveniente primero hablar con los que van a utilizar la BD para determinar qué uso se va a dar de la misma. Si fuese por ejemplo una base de datos para gestionar un comercio, estas preguntas podrían ser:

1. Posibilidad de introducir, modificar, consultar y listar los datos
2. Informes sobre artículos más vendidos
3. Etiquetas postales para clientes
4. Existencias de artículos
5. Pedidos a proveedores, etc

A partir de las preguntas, correcciones y apuntes que hagamos, podemos comenzar a construir las tablas que formarán nuestra BD. Es muy importante no introducir 200 campos por ejemplo en una misma tabla, sino dividir la información en varias tablas que puedan gestionarse por separado, pero enlazarse cuando sea necesario.

Por tanto es muy importante siempre pasar algo de tiempo estudiando el diseño de la base de datos, mejor si lo haces en papel. Los puntos que aparecen a continuación te puede ayudar o dar una idea de lo que tienes que tener en cuenta a la hora de crear esta estructura.

Lo primero que deberías hacer es pensar en todos los datos que debe recoger la base de datos (futuros campos). Posteriormente se debe agrupar en conjuntos menores (futuras tablas).

7.2 Fases en el diseño de una BDR

Las fases de diseño de una base de datos podrían ser:

1. Determinar el propósito: Saber para que vamos a utilizar la base de datos
 - o Análisis de la información que vamos a procesar: Cuales son los datos con los que vamos a trabajar.
 - o Comparar con la información ya existente (hojas de pedidos, facturas, etc). Si la base de datos se crea sobre datos existentes.
 - o Pensar en el usuario medio: Saber quien la va a utilizar, y la forma que le será más fácil para manejar los datos.
2. Determinar las tablas.
 - o Separar los datos por temas: Los datos a utilizar los separaremos de la forma más específica que podamos, para que a la hora de buscar cualquiera de ellos, sepamos exactamente donde encontrarlos.
 - o Evitar duplicados de información: No repetiremos datos de unas tablas a otros (especificar las relaciones entre tablas).
 - o Evitar eliminar información necesaria: Mirar que se incluyan en la base de datos toda la información que necesitemos a la hora de trabajar con la base de datos.
3. Determinar los campos.
 - o Los campos han de ser específicos del tema de la tabla: Cada tabla tendrá sus propios campos.
 - o No crear campos calculados o derivados de otros campos: No añadiremos campos a la tabla como por ejemplo la suma de

otros dos campos. Porque nos supondría un espacio innecesario en la tabla y en la base de datos.

- Decidir cual será el tipo de campo de la tabla, ya que si nos equivocamos al decidir el tipo de campo, después podría suponernos la pérdida de datos de la tabla, al cambiar el tipo de campo.
- Minimizar al máximo el tamaño de los campos: No crearemos campos excesivamente grandes que vayan a contener datos pequeños.
- Definir los índices: Especificar cuáles serán las claves de las tablas, para hacer que la búsqueda en las mismas sea más rápida.
 - Clave principal
 - Índices fáciles de recordar.

8 Caso de estudio y reflexión. Estructura básica de un programa Contable – Financiero (I)

Tomando en consideración la técnica contable basada en el método contable (partida doble) y el lenguaje contable (Diccionario Plan General Contable) y los requerimientos de básicos demandados como son consulta de mayores de cuentas, elaboración del estado patrimonial y cuenta de resultados básicamente.

Diario de Contabilidad			
DEBE		HABER	
<i>Asiento1</i> ----- 02-01-03 -----			
12.000,00 um 570	Caja	A 100	Capital Social
			12.000,00 um
<i>(Asiento de creación de la sociedad)</i>			
<i>Asiento2</i> ----- 10-01-03 -----			
9.500,00 um 572	Bancos c/c	A 570	Caja
			9.500,00 um
<i>(Nuestro ingreso n° 1234)</i>			
<i>Asiento3</i> ----- 15-01-03 -----			
3.000,00 um 600	Compras de Mercaderías	A 400	Proveedores
			3.000,00 um
<i>(S/F n° 1234-B)</i>			
<i>Asiento4</i> ----- 15-02-03 -----			
1.500,00 um 430	Clientes	A 700	Ventas de Mercaderías
			1.500,00 um
<i>(N/F n° 1-A)</i>			
<i>Asiento5</i> ----- 07-03-03 -----			
1.000,00 um 640	Sueldos y salarios	A 572	Banco c/c
			1.000,00 um
<i>(n° de cheque 3434)</i>			
<i>Asiento6</i> ----- 03-04-03 -----			
450,00 um 642	Seguridad Social Empresa	A 572	Banco c/c
			450,00 um
<i>(n° de cheque 3435)</i>			
<i>Asiento7</i> ----- 25-05-03 -----			

Ilustración 1

CUENTAS DE MAYOR			
570 Caja		100 Capital Social	
Debe	Heber	Debe	Heber
(1) 12.000,00 um	(2) 9.500,00 um	(21) 12.000,00 um	(1) 12.000,00 um
(15) 1.500,00 um	(7) 600,00 um		
	(8) 750,00 um		
	(9) 100,00 um		
	(10) 400,00 um		
	(12) 350,00 um		
	(14) 100,00 um		
	(21) 1.700,00 um		
572 Bancos c/c		400 Proveedores	
Debe	Heber	Debe	Heber
(2) 9.500,00 um	(5) 1.000,00 um	(21) 5.000,00 um	(3) 3.000,00 um
	(6) 450,00 um		(13) 2.000,00 um
	(21) 8.050,00 um		
600 Compras de Mercaderías		700 Ventas de Mercaderías	
Debe	Heber	Debe	Heber
(3) 3.000,00 um	(19) 5.000,00 um	(20) 5.000,00 um	(4) 1.500,00 um
(13) 2.000,00 um			(11) 1.500,00 um
			(16) 2.000,00 um
411 Acreedores Efectos			
Debe	Heber		
(21) 3.600,00 um	(10) 3.600,00 um		

Ilustración 2

Cuenta de Pérdidas y Ganancias Correspondiente al Ejercicio Terminado 200X				
Nº CUENTA		Nota Memoria	(Debe) Haber	
			200X	200X-1
700,701,702,703,704, 705,(706),(708),(709)	1. Importe neto de la cifra de negocios		5.000,00 €	
71*,(6930),7930	2. Variacion de existencias de Productos			
73	3. Trabajos realizados por la empresa para su activo			
(600),(601),(602),(607),608,609,61*,(6931),(6932),(6933),606,7931,7932,7933	4. Aprovisionamientos		3.500,00 €	
740,747,75	5. Otros ingresos de explotacionn			
(64),7950,7957	6. Gastos de personal		1.450,00 €	
(62),(631),(634),636,639,(65),(694),(695),794,7954	7. Otros gastos de explotacion		1.300,00 €	
(68)	8. Amortización del inmovilizado			
746	9. Imputación de subvenciones de inmovilizado no financiero y otras			
7951,7952,7955,7956	10. Excesos de provisiones			
(670),(671),(672),(690),(691),(692),770,771,772,790,791,792	11. Deterioro y resultado por enajenaciones del inmovilizado		4.000,00 €	
(678),778	12. Otros resultados			
A) RESULTADO DE EXPLOTACIÓN (1+2+3+4+5+6+7+8+9+10+11+12)			- 5.250,00 €	
760,761,762,769	13. Ingresos financieros			
(661),(662),(665),(669)	14. Gastos financieros			
(663),763	15. Variación de valor razonable en instrumentos financiero			
(668),768	16. Diferencias de cambio			
(666),(667),(673),(675),(696),(697),(698),(699),766,773,775,796,797,798,799	17. Deterioro y resultado por enajenaciones de instrumentos financieros			
B) RESULTADO FINANCIERO (13+14+15+16 +17)			- 5.250,00 €	
C) RESULTADO ANTES DE IMPUESTOS (A+B)			- 5.250,00 €	
6300*,6301*(,633),638	18. Impuestos sobre beneficios			
D) RESULTADO DEL EJERCICIO (C + 18)			- 5.250,00 €	

Ilustración 3

Cuentas Anuales: Balance-Activo				
Nº CUENTA	ACTIVO	NOTAS MEMORIA	200X	200X-1
A) ACTIVO NO CORRIENTE			600,00 €	
20,(280),(290)	I. Inmovilizado intangible.			
21,(281),(291),23	II. Inmovilizado material.		600,00 €	
22,(282),(292)	III. Inversiones inmobiliarias.			
2403,2404,2413,2414,2423,2424,2433,2434,(2483),(2484),(2933),(2934),(2943),(2944),(2953),(2954)	IV. Inversiones en empresas del grupo y asociadas.			
2405,2415,2425,2435,(2485),250,251,252,253,254,2553,256,257,258,(259),26,(2935),(2945),(2955),(296),(297),(298),474	V. Inversiones financieras a largo plazo.			
	VI. Activos por Impuesto diferido.			
B) ACTIVO CORRIENTE			14.750,00 €	
580,581,582,583,584,(599)	I. Activos no corrientes mantenidos para la venta.			
30,31,32,33,34,35,36,(39),407	II. Existencias.		1.500,00 €	
	III. Deudores comerciales y otras cuentas a cobrar.			
430,431,433,434,435,436 (437)(490)	1. Clientes por ventas y Prestaciones de servicios		3.500,00 €	
558	2. Accionistas (socios) por desembolsos exigidos			
44,460,470,471,472,(493),5531,5533,544	3. Otros deudores			
(5933),(5934),(5943),(5944),(5953),(5954)	IV. Inversiones en empresas del grupo y asociadas a corto plazo.			
2550,5305,5315,5325,5335,(5385),540,541,542,543,545,546,548,	V. Inversiones financieras a C/P			
	VI. Periodificaciones			
57	VII. Efectivo y otros activos líquidos equivalentes.		9.750,00 €	
TOTAL ACTIVO (A+B)			15.350,00 €	

Ilustración 4

Se pide

Llevar a cabo un análisis y reflexión que permita convertir o traducir el esquema de funcionamiento contable tradicional en términos de base de datos distinguiendo o identificado los principales componentes de una BD, en concreto:

- Tablas
- Consultas
- Informes
- Macros
- Módulos

Sugerencias, recomendaciones. (II)

- Tablas principales y auxiliares necesarias.
- Definir las relaciones y dependencias que se pudieran provocar, tablas relacionados o vinculadas.
- Identificar los campos básicos de cada tabla, tipo y características.

- Consultas
- Informes básicos o plantillas
- Macros: Asientos automáticos. Asientos de cierre y apertura... otros asientos predefinidos.
- Diario contable (tabla maestra e informes)
- Estructura del Plan General Contable básica (Cuentas de mayor o mastras)
- Plan General contable de la empresa (Cuentas auxiliares o de desarrollo)

9 Bibliografía

http://aulavirtual.miguelbayon.com/typo3_aulaVirtualv1_0/192.168.0.5/index5255.html?id=9
Bases de Datos guía de aprendizaje: Jorge Sánchez (www.jorgesanchez.net) año 2004

