Temas Generales para la preparación de la Oposición al Cuerpo Superior de Estadísticos del Estado.

Cuerpo Superior de Estadísticos del Estado
Especialidad de Estadística-Ciencia de Datos.

Almacenamiento y modelos de datos

Tema 4. Bases de datos.

AUTOR: **Antonio J. Sánchez-Padial**

Asociación Profesional de Cuerpos Superiores de Sistemas y Tecnologías de la Información de las Administraciones Públicas

Creación: Agosto 2021

ÍNDICE

1 Introducción

En los últimos 50 años, las bases de datos se han convertido en uno de los principales medios de almacenamiento y recuperación de información. La industria generada en torno a esta tecnología está valorada entre los 35 y 50 mil millones de euros anuales. Hoy día, se trata de una tecnología con la que trabajamos de forma cotidiana, incluso de manera inadvertida.

En el tema consideraremos que una **base de datos** es un conjunto de datos relacionados entre ellos bajo el control de un **sistema de gestión de bases de datos**. Por otro lado, las **aplicaciones de bases de datos** son aquellos programas que hacen uso de la base de datos en algún momento de su ejecución. También se hace uso del término **sistema de base de datos**, más extenso, que define el conjunto de los programas de aplicación, junto con el sistema de gestión de la base de datos, más la base de datos propiamente dicha.

Se presentan algunos ejemplos de uso de bases de datos en nuestra vida cotidiana.

Compras en una tienda local

Muchas tiendas cuentan con una base de datos con la que controlan sus inventarios. Cada vez que se escanea un código de barras, se realiza una consulta a esta base de datos que sirve para consultar el precio. Además la aplicación de caja, descontará del inventario el número de elementos comprados. Cuando el nivel de inventario cae por debajo de un valor fijado, el responsable puede recibir una notificación para incluir el producto en el próximo pedido. La base de datos también sirve para consultar si un producto está disponible cuando se recibe un pedido por teléfono; y para analizar las ventas de los diversos productos a lo largo del tiempo.

Compras con tarjeta de crédito

Cuando hacemos una compra con nuestra tarjeta de crédito, es necesario que el sistema haga uso de una base de datos. Una vez que la tarjeta se lee en el lector, este se comunicará telefónicamente o a través de un ordenador conectado a Internet con una base de datos de nuestro banco. Una aplicación consultará si la clave que hemos introducido es correcta, y si lo es, a continuación decidirá si tenemos saldo suficiente para realizar la operación. Si la compra es aprobada, anotará en la base de datos los detalles de la misma, y descontará el importe del saldo. La base de datos también realiza otras operaciones de seguridad, puede comprobar si el número de la tarjeta de crédito está en una lista de tarjetas robadas o extraviadas. También utiliza sus datos almacenados para realizar operaciones estadísticas que minimizan el riesgo de fraude.

Usando la biblioteca

La mayoría de las bibliotecas, si no todas, cuentan con bases de datos donde almacenan la información de los ejemplares, de los socios, y de los préstamos. Cuando tomamos prestado un libro, el bibliotecario puede consultar que no tenemos ninguna penalización pendiente por entregas atrasadas, y también registrará en la base de datos qué usuario ha tomado prestado el libro y cuál es la fecha de devolución. Hoy día los usuarios también pueden consultar esa información a través de Internet, en una página web que utiliza la base de datos para mostrar dicha información. Una aplicación puede enviar una notificación al correo electrónico de los usuarios para recordar que se acerca el plazo de devolución de un ejemplar, o que ya está disponible un libro sobre el que el usuario hizo una reserva, que por supuesto, también fue registrada en la base de datos.

Usando Internet

Muchos de las páginas web que visitamos funcionan mediante aplicaciones de bases de datos. Por un lado, las tiendas online funcionan de una manera parecida a la descrita más arriba para las tiendas físicas, y hacen uso igualmente del pago mediante tarjeta. Pero también las utilizan otras páginas web, como las redes sociales. Cuando accedemos con nuestro usuario, se comprueban nuestros datos y contraseña con lo almacenado en la base de datos. También se almacenan en la base de datos nuestros comentarios y fotografías, así como los usuarios con los que tenemos relaciones, seguidores, amigos, etc. Utilizando esta información de la base de datos, la aplicación de la red social puede decidir a qué usuarios les mostrará nuestras novedades, y del mismo modo, crea una página web con las novedades de todos los usuarios y usuarias con los que tenemos conexión.

2 Sistemas tradicionales basados en ficheros: descripción y limitaciones.

Los sistemas basados en ficheros son un paso previo en la gestión automatizada de los datos a las bases de datos. Son anteriores, tanto desde el punto de vista histórico, ya que las bases de datos se inventaron como una respuesta a los problemas y limitaciones del uso de ficheros; pero también son anteriores desde el punto de vista educativo o profesional, ya que normalmente se aprende en primer lugar a desarrollar aplicaciones basadas en ficheros, para más adelante utilizar bases de datos y escribir aplicaciones para estas. Por estos motivos, es conveniente describir en primer lugar estos sistemas y sus limitaciones.

o Descripción

Los sistemas basados en ficheros son una versión digital de los sistemas de ficheros manuales de gestión de la información. En muchas organizaciones, se utilizan ficheros para almacenar la información de clientes, proyectos, facturas, contratos, etc. Estos ficheros pueden almacenarse en archivos bajo llave, o en determinada área de la organización en la que se puede realizar un control sobre el acceso a los mismos. A menor escala, también se utilizan estos ficheros en el ámbito doméstico para guardar facturas, contratos, números de teléfono, recuerdos y fotografías, que disponemos en carpetas o estanterías, y a los que podemos acudir para recuperar determinada información. Cuando un sistema de ficheros manual crece, se puede hacer necesario contar con algún sistema de indexado que permita conocer la ubicación de los distintos ficheros en la organización.

Un sistema de ficheros manual de estas características funcionará razonablemente bien mientras no crezca demasiado. Un sistema de grandes dimensiones también puede funcionar bien, si solo es necesario almacenar y recuperar los documentos o ficheros de manera individual. El sistema se vuelve menos eficiente y sostenible cuando es necesario realizar continuas búsquedas cruzadas complejas, como las siguientes:

- ¿Cuántas de casas de tres habitaciones hay disponibles para la venta con ascensor y garaje?
- ¿Cuántos apartamentos hay disponibles para alquiler que estén a menos de 3 kilómetros de la universidad?
- ¿Cuál es el precio medio de alquiler de los apartamentos de dos habitaciones?
- ¿Cuáles son los ingresos netos mensuales esperados para el año fiscal en curso?

Todas estas preguntas, y muchas otras similares, son muy difíciles de responder utilizando un sistema de ficheros manual. Así los sistemas basados en ficheros fueron desarrollados como una respuesta digital a estos problemas. El principal inconveniente es que en lugar de construir un sistema de datos centralizado para toda la organización, se optó por una solución distribuida, donde cada uno de los departamentos construye su propia solución, con poca o ninguna comunicación con el resto de departamentos, y el apoyo del departamento TIC.

En sistemas informáticos, un fichero es un conjunto de **registros**, con contiene información relacionada lógicamente. Cada registro contiene un conjunto de varios **campos** relacionados, en el que cada campo representa una característica del objeto del mundo real que representa

Imaginemos que el Departamento Comercial se encarga de recibir la información de propietarios que desean vender o alquilar su casa. Por otro lado, reciben solicitudes de clientes que buscan una casa para comprar o alquilar, describiendo las características que desean. Con ayuda del departamento TIC podría haber construido varias aplicaciones basadas en ficheros, que le permitieran dar de alta nuevas propiedades, gestionar los datos de sus dueños, realizar consultas para encontrar propiedades que cumplen los criterios de alguna solicitud, así como para gestionar los datos de los potenciales compradores. El sistema tendría tres ficheros, uno con registros de los propietarios, otro con registros de las viviendas, y un tercero con los registros de los clientes.

PropiedadesAlquiler

idViv	dirección	ciudad	codPostal	tipo	hab.	alquiler	idPropietario
1012	Calle de la Luna, 6	León	24003	casa	4	750	P0205
1013	Calle del río Duero, 25	Burgos	09006	piso	3	650	P0301
1023	Plaza de la Estrella, 3	Soria	42002	piso	2	350	P0708
1034	Calle de Medinaceli, 32	Soria	42001	casa	5	1000	P0708

Propietarios

idPropietario	nombre	apellidos	dirección	teléfono
P0205	Josefina	Valle Soler	c/ Angustias, 28. León	555 123432
P0301	Marta	García Foces	c/ Moscú, 12. Burgos	555 010267
P0708	Alfredo	Gómez Molina	c/ de la Paz, 8 . Madrid	555 252627

Clientes

idCliente	nombre	apellidos	direccion	telefono	tipobuscado	alquilerMax
C0123	Andrés	Soria Cruz	c/ España, 23.	555 234345	piso	425
C0431	Sonia	Álvarez	c/ Soledad, 121	555 789432	casa	700
C0675	Mercedes	González	c/ Remedios, 1	555 236754	piso	400

Por otro lado, el Departamento de Contratos se encarga de gestionar los acuerdos de contratos de cada una de las propiedades en la compañía. De igual modo, decide digitalizar sus archivos, y construye un sistema basado en ficheros con ayuda del departamento TIC. El sistema tiene tres archivos donde almacena información sobre el alquiler, las propiedades y los clientes; y que en buena parta, son muy similares a las utilizadas por el Departamento Comercial.

Alquileres

idAlquiler	idViv	idCliente	alquiler	modoPago	deposito	pagado	inicio	fin
90010	1013	C0431	600	tarjeta	1500	S	01 jun 11	31 may 12
90103	1034	C0675	1000	cheque	1000	N	1 ago 13	31 ene 14
90099	1023	C0123	350	efectivo	350	S	1 jul 13	30 jun 14

PropiedadesAlquiladas

idViv	dirección	ciudad	codPostal	alquiler
1013	Calle del río Duero, 25	Burgos	09006	600
1023	Plaza de la Estrella, 3	Soria	42002	350
1034	Calle de Medinaceli, 32	Soria	42001	1000

Clientes

idCliente	nombre	apellidos	direccion	telefono
C0123	Andrés	Soria Cruz	c/ España, 23.	555 234345
C0431	Sonia	Álvarez	c/ Soledad, 121	555 789432
C0675	Mercedes	González	c/ Remedios, 1	555 236754

En esta arquitectura de datos, cada departamento gestiona sus propias aplicaciones, así como los ficheros que utiliza. Cada aplicación tiene su propio ciclo de mantenimiento, y realiza modificaciones en los formatos de los ficheros, y en los formatos de entrada e informes de salida de modo independiente. Además, el soporte físico de cada uno de los sistemas se mantenía por separado, y las estructuras de datos están implementadas directamente en el código de las aplicaciones.

Se puede observar que hay bastante redundancia en la información que almacenan los dos sistemas. Y seguramente también la habría en otros sistemas de la organización. Por ejemplo, tanto el Departamento de Personal como el de Nóminas tendrían información sobre los empleados, que coincidiría en una parte, y en otra podría ser específica de las necesidades de cada departamento.

o Limitaciones

A partir del ejemplo de la sección anterior veremos algunos de los inconvenientes y limitaciones asociados a los sistemas basados en ficheros.

▪ Separación y aislamiento de datos

Al almacenar los datos en ficheros independientes resulta más complicado acceder a información que debería ser fácil de combinar. Por ejemplo, si queremos crear una lista de casas que cumplen los requisitos de algún cliente, en primer lugar deberemos buscar en el fichero de clientes a todos los que están interesados en viviendas de tipo 'casa'. A continuación buscaremos las viviendas de tipo 'casa' en el fichero de PropiedadesAlquiler y con rentas inferiores al límite fijado para cada uno de los clientes. Los sistemas basados en ficheros no facilitan este tipo de búsquedas, aunque se trata de un caso bastante sencillo y habitual. Los desarrolladores de aplicaciones deben sincronizar el acceso a dos ficheros para resolver la consulta, lo que se complica aún más cuando es necesario acceder a ficheros adicionales.

▪ Duplicación de datos

Debido a la estrategia descentralizada que siguen los departamentos de la organización, los sistemas basados en ficheros facilitan, y promueven en cierta medida, la duplicación innecesaria de los datos. En las tablas de ejemplo, se ve como existen duplicados de los datos de clientes y propiedades tanto en el Departamento Comercial como en el de Contratos. Los duplicados innecesarios deben ser evitados por varias razones, como:

- La duplicación consume recursos. Hay costes asociados al almacenamiento, y también al esfuerzo asociado a introducir y modificar los datos más de una vez.
- La duplicación conlleva un riesgo de ambigüedad, y por tanto una pérdida de integridad de los datos. Si los datos asociados al mismo cliente son distintos en cada fichero, ¿cómo podemos determinar cuál es el correcto? En este tipo de sistemas existe un riesgo constante, de que los datos de un cliente sean actualizados solo en uno de los sistemas que se gestiona. Y por otro lado, la actualización de datos duplicados en múltiples ficheros puede provocar errores tipográficos en alguno de ellos, que pasen inadvertidos, causando problemas más adelante.

▪ Dependencia entra datos y programas

En los sistemas basados en ficheros, la estructura de los ficheros de datos y sus registros suele estar definidos en el código de las aplicaciones que los manipulan. Esto dificulta los cambios a estas estructuras. Por ejemplo, ampliar el tamaño de un campo de texto de 40 a 45 caracteres (o 41), obliga a modificar todas las aplicaciones que leen este fichero, además de escribir una aplicación que modifique el fichero realizando una copia exacta del mismo que tenga en cuenta la ampliación del número de caracteres en el campo correspondiente.

Es necesario modificar todos los programas que hagan uso de este fichero, incluso aquellos que no hagan uso del campo modificado. En ocasiones, puede ser difícil para el equipo de desarrollo identificar todas estas aplicaciones. En general identificar, modificar y probar todas las aplicaciones que utilizan el fichero modificado supone un sobrecoste, además de ser una tarea muy proclive a errores y omisiones.

▪ Incompatibilidades entre formatos de fichero

Puesto que la estructura de los ficheros es generada por las programas de aplicación, dicha estructura depende del lenguaje de programación de la aplicación. Por ejemplo, la estructura de los ficheros generados por un programa COBOL puede ser diferente de la estructura de los ficheros generados en C. Estas diferencias puede que haga difícil hacer programas que trabajen con ficheros generados con distintos lenguajes de programación, ya que los desarrolladores de este nuevo programa tendrán que tener en cuenta las diferencias entre las estructuras de estos ficheros, y buscar la manera de solucionarlas, con el coste en tiempo y esfuerzo que esto supondrá para el desarrollo.

- **Rigidez de consultas y proliferación de programas de aplicación**

Las aplicaciones basadas en ficheros suponen una gran mejora para los usuarios con respecto a los sistemas de ficheros manuales. A partir de este éxito, se genera una demanda creciente de nuevas consultas e informes, así como de modificaciones a las existentes. Todas estas modificaciones recaen sobre el equipo de desarrollo, ya que todas las consultas e informes están estrechamente vinculadas a los programas de aplicación. Esta fuerte demanda provoca dos tipos de situaciones. Por un lado, algunas organizaciones proporcionan un catálogo fijo de consultas que no puede ser modificado. No hay manera de realizar nuevas consultas, si no es en base a una planificación periódica del equipo de desarrollo en la que se irá decidiendo qué nuevas consultas son incluidas, y qué modificaciones se realizan a las existentes.

Por otro lado, en otras organizaciones, crece de manera descontrolada el número de ficheros y programas de aplicación, ya que cada equipo hace los desarrollos que considera necesarios. En un momento, el departamento TIC no podrá proporcionar servicio a todos los cambios solicitados. El aumento de la presión sobre este departamento puede dar lugar a dificultades en satisfacer todas las solicitudes de los usuarios, una inadecuada documentación de los programas, y dificultades en el mantenimiento. A menudo, alguno de las siguientes funcionalidades era omitida:

- No había una planificación de la seguridad o la integridad de los datos
- La capacidad de recuperación, tanto desde un fallo de hardware como de software, era escasa o no existente
- No había capacidad de acceso simultáneo para varios usuarios del mismo departamento.

Todas estas limitaciones y desventajas hacían necesario el desarrollo de un nuevo paradigma.

3 Base de datos, sistema de gestión de bases de datos, programas de aplicación y componentes de un sistema de gestión de bases de datos (generalidades).

Todas las limitaciones descritas para los sistemas basados en ficheros provienen de estos dos factores:

- La definición de los datos está fuertemente vinculada a los programas de aplicación, en lugar de almacenarse de manera separada e independiente.
- No hay control sobre los accesos o manipulación de los datos que no sea el impuesto por los programas de aplicación.

Las limitaciones propias de un sistema fuertemente descentralizado fueron superadas mediante el desarrollo de un paradigma basado en la gestión centralizada de los datos: las bases de datos

o La Base de Datos

Según el diccionario de la RAE una base de datos es un conjunto de datos organizado de tal modo que permita obtener con rapidez diversos tipos de información.

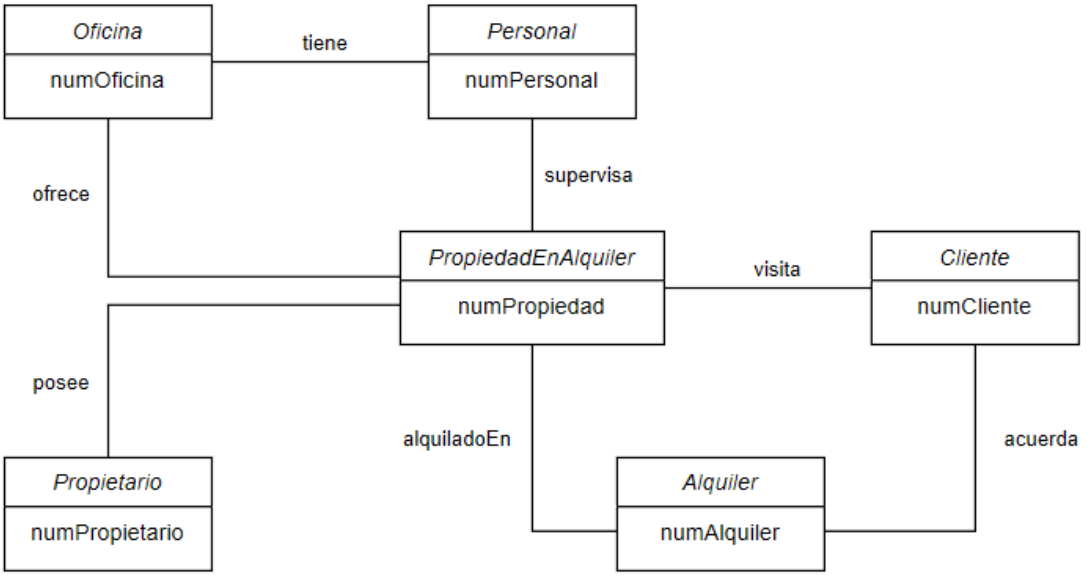
Está descripción es bastante genérica, y en ella tienen cabida la mayor parte de las estructuras de datos en las que podríamos pensar. Vamos a intentar describir de qué hablamos cuando usamos el término base de datos. Una base de datos es un almacén de datos, normalmente de gran tamaño, que puede ser utilizado de manera simultánea por distintos usuarios o departamentos de una organización. Al sustituir los sistemas basados en ficheros por un sistema centralizado se reduce la duplicidad de la información. La base de datos dejó de ser un recurso controlado por cada departamento de la organización, para convertirse en un activo de la organización. Al contrario que los ficheros, la base de datos no solo almacena los datos de la organización, sino que también almacena la descripción de la estructura de los datos. Por esto se dice que una base de datos es una *colección autodescrita de datos integrados*. La parte de la base de datos que define la estructura de los datos se conoce como **catálogo del sistema** (o también **diccionario de datos** o **metadatos**). Es gracias a esta capacidad de describir la estructura de los datos por lo que las bases de datos pueden romper la dependencia entre datos y programas.

La estrategia utilizada en los sistemas de bases de datos se parece a la seguida en la construcción del software en un paradigma orientado a objetos, donde hay una parte del código que es pública y que es utilizada por otros sistemas para utilizar los servicios que ofrece el sistema, y hay otra parte del código privada, a la que no tienen acceso otras aplicaciones, y que no es necesario conocer para utilizar los servicios, aunque sea utilizada de manera interna para llevarlos a cabo. Una ventaja de este modelo, que conocemos como **abstracción de datos**, es que es posible modificar el funcionamiento interno de un objeto sin afectar a los usuarios de dicho objeto, siempre que no haya cambios en su parte pública. De manera análoga, la base de datos separa la estructura de los datos de los programas de aplicación. Si se añaden nuevas estructuras de datos o se modifican las existentes, los programas de aplicación no se ven afectados, siempre y cuando no utilicen la parte modificada.

Otra parte de la definición de base de datos sobre la que merece la pena detenerse es la “conjunto de datos organizado”. Cuando analizamos la información utilizada en una organización identificamos entidades, atributos y relaciones. Una **entidad** es un objeto distinguible (una persona, un lugar, una cosa, un concepto o un evento) de la organización que se representa en la base de datos. Un **atributo** es una propiedad que describe alguno de estos objetos que estamos representando. Una **relación** es una asociación entre entidades. Por ejemplo, el siguiente diagrama muestra un diagrama Entidad-Relación (ER) para el ejemplo de la empresa de alquiler de viviendas. En él encontramos:

- Seis entidades (los rectángulos): Oficina, Personal, PropiedadEnAlquiler, Cliente, Propietario y Alquiler;
- Siete relaciones (los nombres cercanos a las líneas): Tiene, Ofrece, Supervisa, Visita, Posee, AlquiladoPor y Acuerda
- Seis atributos, uno para cada entidad: NumOficina, numPropiedad, numCliente, numPropietario y numAlquiler.

La base de datos representa las entidades, los atributos y las relaciones entre las entidades. Por eso decimos que es un “conjunto de datos organizado”



El Sistema de Gestión de Bases de Datos

El sistema de gestión de base de datos (SGBD) es un sistema software implementado para definir, crear, mantener y controlar el acceso a bases de datos.

El SGBD es una aplicación a la que se conectarán tanto los usuarios de la base de datos, como los programas de aplicación que accedan a esta. Normalmente contará con las siguientes características:

- Los usuarios pueden crear una base de datos, normalmente usando un **Lenguaje de Definición de Datos (LDD)**. Este lenguaje permite crear tipos y estructuras de datos, así como restricciones que deben cumplir los datos que se almacenarán en la base de datos.
- Permite recuperar datos de la base de datos, así como realizar inserciones, modificaciones y borrados, normalmente utilizando un **Lenguaje de Manipulación de Datos (LMD)**. El LMD es capaz de acceder a todos los datos de la base de datos, gracias a la gestión centralizada que realiza el SGBD, mediante un **lenguaje de consultas**. Gracias a este lenguaje de consulta el SGBD ofrece una estrategia para superar la rigidez en las consultas propia de los sistemas basados en ficheros, facilitando la actualización y la gestión del *software*. El lenguaje de consultas más popular es **SQL**, que es tanto formalmente como de facto el lenguaje de consultas más popular en SGBDs relacionales.
- Proporciona un control de acceso a la base de datos, que facilita la existencia de:
    - Un sistema de seguridad, que evita accesos no autorizados a la base de datos
    - Un sistema de integridad, que mantiene la consistencia de los datos almacenados
    - Un sistema de control de concurrencia, que permite el acceso simultáneo a la base de datos
    - Un sistema de recuperación, que devuelve la base de datos a un estado previo considerado consistente tras un error de *hardware* o de *software*
    - Un catálogo accesible a los usuarios, que contiene la descripción de los datos almacenados en la base de datos.

Programas de aplicación de bases de datos

Los programas de aplicación de base de datos son aquellos que interaccionan con la base de datos, normalmente mediante consultas de SQL.

Los usuarios suelen utilizar la base de datos a través de **programas de aplicación**, que se usan tanto para crear y mantener la base de datos, como para generar informes. Pueden ser cualquier tipo de

programas, desde programas *batch* diseñados para ejecutarse en segundo plano, o aplicaciones *web* que los usuarios manejan a través de Internet. Los programas de aplicación pueden estar escritos en lenguajes de programación generalistas, o utilizando lenguajes de 4ª generación de más alto nivel.

En el ejemplo de la compañía de alquiler de viviendas, cada departamento podría continuar teniendo sus propias aplicaciones para la gestión de las viviendas, de los clientes, etc. pero todas las aplicaciones harían uso de la misma base de datos a través de un SGBD.

Las vistas

Los SGBD proporcionan una serie de características muy potentes que facilitan el manejo de los datos de una organización. Sin embargo, para algunos usuarios puede parecer que acceder a todas esas características o a todo el modelo de datos de la organización, añade un nivel de complejidad innecesario. El **mecanismo de vistas** surgen con el propósito de reducir este problema, ofreciendo a cada grupo de usuarios una vista particular de la base de datos, que es un subconjunto especializado de la base de datos.

Las vistas reducen la complejidad aparente al permitir a los usuarios ver solamente los datos en los que están interesados. Pero además, tienen otras ventajas:

- *Proporciona seguridad*. Las vistas pueden configurarse para que oculten datos que algunos usuarios no deberían ver. Por ejemplo, los ejecutivos y el departamento de personal podrían acceder a los datos de salario e todo el personal de la compañía, pero otros usuarios tendrían una vista en la que no se mostraría esta información.
- *Permiten modificar la apariencia de la base de datos*. Por ejemplo, el departamento de contratos puede preferir que el campo Alquiler tenga un nombre más descriptivo, como AlquilerMensual.
- *Muestran una imagen consistente e invariable de la estructura de la base de datos*. Es posible que aunque se modifique la estructura de la base de datos, las estructuras ofrecidas por las vistas permanezcan invariables. Esto facilita el trabajo de los usuarios que no tienen por qué conocer la estructura interna de la base de datos.

Las características descritas son generales, y en la práctica pueden variar entre los distintos fabricantes de SGBDs. Por ejemplo, un SGBD diseñado para ordenadores personales puede no tener características para el acceso concurrente a la información. Sin embargo, los grandes sistemas comerciales ofrecen todas estas características y muchas otras. La complejidad de estos sistemas es fruto de estar diseñados para cubrir las necesidades generales de un gran espectro de usos. Estos sistemas requieren, en muchas ocasiones, un alto nivel de disponibilidad, 24 horas al día, 7 días a la semana, y ser capaces de sobreponerse ante fallos de hardware o de software. Los SGBDs están en continua evolución para satisfacer todas las nuevas necesidades de los usuarios. Algunas aplicaciones pueden gestionar tipos de datos avanzados como información geográfica, imágenes, audio, o incluso estructuras de datos.

o Componentes de un SGBD

Los principales componentes del entorno de un SGBD son el *hardware*, el *software*, los datos, los procedimientos y los humanos que interactúan con este.

▪ *Hardware*

Tanto el SGBD como los programas de aplicación necesitan un *hadrware* sobre el que ejecutarse. Puede tratarse de un simple ordenador personal, un potente servidor, o un clúster de máquinas virtuales en la nube. El tipo particular de *hardware* dependerán de las necesidades de la organización y del SGBD utilizado. Algunos SGBD solo funcionan sobre determinado tipo de *hardware* o sistemas operativo, mientras que otros funcionan en una amplia variedad de sistemas. Un SGBD tendrá unos requisitos mínimos de espacio almacenamiento, memoria RAM o capacidad de los procesadores, pero es posible que el rendimiento no sea óptimo cumpliendo estos requisitos, si el sistema desarrollado es muy grande o de muy alta complejidad.

La arquitectura cliente-servidor es una de las más utilizadas en SGBDs. El software del SGBD corre en un sistema hardware que denominamos servidor, y responde a las peticiones que los usuarios hacen

desde sus propios sistemas, bien desde la interfaz de la SGBD o desde programas de aplicación. Estos sistemas que se conectan al servidor, para realizar peticiones y diversas tareas, se denominan clientes.

- ***Software***

La componente *software* incluye el propio *software* SGBD y los programas de aplicación, junto con el sistema operativo, incluyendo el *software* de red si el SGBD se utiliza en línea. Normalmente, los programas de aplicación están escritos en un lenguaje de programación de tercera generación como C, C++, C#, Java, Visual Basic, o uno de cuarta generación, como SQL, encastrado en un lenguaje de tercera generación. El SGBD puede tener su propio lenguaje de cuarta generación para el desarrollo rápido de aplicaciones proporcionando lenguajes de búsqueda no procedurales, generadores de informes, generadores gráficos de consultas y generadores de aplicaciones. El uso de las herramientas de cuarta generación puede mejorar la productividad significativamente, generando programas más fáciles de mantener.

- **Datos**

Quizás el componente más importante en el entorno del SGBD – al menos desde el punto de vista del usuario final – son los datos. La base de datos contiene tanto los datos operacionales y los metadatos, "los datos sobre los datos". Llamamos **esquema** a la estructura de la base de datos. Los esquemas se componen de **tablas** y de **relaciones** entre ellas. Los campos o columnas de las tablas se denominan **atributos.** En este componente de datos se incluye también el **catálogo del sistema**, que es la parte de la base de datos que almacena el esquema o estructura de la base de datos.

- **Procedimientos**

Los procedimientos son las instrucciones y reglas que rigen el diseño y uso de la base de datos. Los usuarios del sistema y el personal que gestiona la base de datos necesitan procedimientos documentados sobre cómo utilizar o ejecutar el sistema. Pueden incluir instrucciones como:

- Conexión con el SGBD
- El uso de una característica específica del SGBD o de un programa de aplicación
- Arranque y parada del SGBD
- Creación de copias de seguridad de la base de datos
- Gestión de los fallos de *software* y *hardware*. Incluyendo procedimientos sobre cómo identificar el componente que produce el error, cómo arreglar el componente (por ejemplo telefoneando al equipo técnico), y, cómo, una vez arreglado el fallo, recuperar la base de datos.
- Cómo cambiar la estructura de la base de datos, reorganizar la base de datos en varios discos, mejorar el rendimiento, o archivar datos en un dispositivo secundario.

4 Ventajas y desventajas de los sistemas de gestión de bases de datos.

Los sistemas de gestión de bases de datos ofrecen considerables ventajas en el diseño, desarrollo y gestión de sistemas de información. Sin embargo, no son una solución infalible, y poseen desventajas que conviene reseñar.

o Ventajas

▪ Control de la redundancia de datos

Como vimos anteriormente los sistemas basados en ficheros pueden desperdiciar capacidad de almacenamiento al guardar la misma información repetida en más de un fichero. Por el contrario, utilizando una base de datos intentamos eliminar esta repetición mediante la integración de todos los datos de modo que no existan copias múltiples de la misma información. Este enfoque no elimina la redundancia por completo, pero facilita su control. En ocasiones se duplica la información de las claves para modelar las relaciones entre entidades; en otras ocasiones, es necesario duplicar algunos elementos para mejorar el rendimiento de la base de datos.

▪ Consistencia de datos

Al eliminar o reducir las repeticiones, disminuye el riesgo de inconsistencias en los datos. Si un dato está almacenado solo una vez en la base de datos, cualquier modificación de su valor debe realizarse en una única ubicación del sistema, y el nuevo valor está disponible de manera inmediata para todos los usuarios. Si un dato estuviera almacenado más de una vez, y el sistema tuviera conocimiento de ello, el sistema puede asegurar que todas las copias tienen valores consistentes. Desafortunadamente, muchos SGBDs no son capaces de garantizar este tipo de consistencia en copias múltiples.

▪ Más información por unidad de datos

Al integrar todos los datos operacionales de la organización, podría ser posible que la organización extrajera conocimiento adicional de las relaciones existentes entre sus datos. Por ejemplo, en el ejemplo de la compañía de alquileres de viviendas el Departamento de Contratos no tiene información sobre el propietario de una de las casas alquiladas. De igual modo, el departamento de ventas desconoce la información sobre los alquileres. Cuando integramos estos ficheros en una base de datos, ambos departamentos podrían acceder al resto de la información, pudiendo sacar nuevas conclusiones sobre el comportamiento de los clientes, etc.

▪ Datos compartidos

En los sistemas basados en ficheros, es típico que los ficheros sean propiedad o estén controlados por los usuarios o departamentos que los utilizan. Por otro lado, las bases de datos suelen considerarse un activo de la organización y puede permitir el acceso a los datos a aquellos usuarios autorizados. Así, es posible que más usuarios sean capaces de acceder a más información. Además, los nuevas aplicaciones pueden basarse en los datos existentes en la base de datos, en lugar de tener que definir de nuevo todos los requisitos de datos. Estas aplicaciones pueden también aprovechar las funciones disponibles del SGBD, como las de definición y manipulación de datos, o de control de concurrencia o recuperación, en lugar de tener que desarrollar dichas funciones en la aplicación.

▪ Integridad de datos mejorada

La integridad de la base de datos se refiere a la validez y consistencia de los datos que almacena. La integridad se suele expresar en forma de **restricciones**, que son reglas de consistencia que la base de datos debe cumplir en todo momento. Las restricciones se pueden aplicar a los registros de una tabla de la base de datos, o a las relaciones que afectan a registros de distintas tablas. Por ejemplo, una restricción puede indicar que el sueldo de un empleado no puede ser superior a 90 000 EUR o que el

código de oficina que se incluye en un registro de la tabla de personal debe corresponder con un código de oficina de la tabla de oficinas.

- **Mejoras en la seguridad**

La seguridad de la base de datos permite protegerla de accesos no autorizados. Sin medidas de seguridad adecuadas, la centralización de la base de datos hace que sea más vulnerable que un sistema basado en ficheros. Sin embargo, la integración de los datos permite que el administrador de la base de datos defina medidas de seguridad para la base de datos, que el SGBD se asegura de cumplir. Estas medidas de seguridad pueden incluir un sistema de nombre de usuario y contraseñas para identificar a los usuarios autorizados para acceder a la base de datos. Los usuarios pueden tener niveles de autorización que restrinjan las operaciones a las que tienen acceso (recuperación, inserción, borrado o modificación). Por ejemplo, el administrador de la base de datos tiene acceso a toda la información de la base de datos; el director de una oficina tiene acceso a todos los datos relacionados con su oficina; y los ayudantes de ventas podrían tener acceso a los datos de las viviendas, pero no tendrían acceso a otros datos delicados como los sueldos del resto del personal.

- **Uso de estándares**

La integración de los datos en una base de datos permite al administrador definir esquemas de datos basados en estándares, que serán seguidos en toda la organización. Estos estándares pueden ser de carácter departamental, institucional, nacional o internacional, y tratar aspectos como los formatos datos que faciliten la interoperabilidad entre sistemas, normas de nombrado de elementos, normas de documentación, procedimientos de actualización de la información, o normas de acceso.

- **Economía de escala**

Combinar todos los datos operacionales de la organización en una base de datos y construir un conjunto de aplicaciones que trabajen con este origen de datos único puede significar un ahorro de costes. El presupuesto que se destinaría a cada departamento para el mantenimiento de sus sistemas basados en ficheros podría combinarse con este fin, y posiblemente se tradujera en menores costes, debido a la aparición de la economía de escala.

- **Equilibrio entre requisitos enfrentados**

Las necesidades de los diferentes usuarios o departamentos pueden entras en conflicto con las del resto de usuarios. La arquitectura centralizada de las bases de datos, permite que el administrador de la misma puede priorizar unas necesidades sobre otras, o encontrar soluciones de compromiso que satisfagan en alguna medida todas las comunidades.

- **Mejora en el acceso a datos**

Gracias a la centralización de la base de datos, los usuarios finales pueden tener acceso a datos transversales a distintos departamentos. De este modo, el sistema puede potencialmente ofrecer muchas más funcionalidades. Gracias a los lenguajes de consulta los usuarios pueden generar consultas “sobre la marcha” y obtener cualquier información de modo casi automático en su terminal, sin necesidad que un programador haga un desarrollo específico.

- **Mejoras en la productividad**

El SGBD proporciona muchas de las funcionalidades estándares que los programadores tenían que escribir al desarrollar aplicaciones para un sistema basado en ficheros. En el nivel más básico, el SGBD proporciona todas las rutinas de bajo nivel para la gestión de ficheros más típicas en los programas de aplicación. Al no tener que preocuparse de estas funciones, los programadores pueden concentrarse en las funciones específicas de su aplicación, donde se concentra el valor añadido. Muchos SGBD, además, proporcionan un entorno de cuarta generación, con un conjunto de aplicaciones para simplificar el desarrollo de aplicaciones sobre la base de datos. Estas herramientas pueden significar un incremento en la productividad de los desarrolladores, la reducción en los tiempos de desarrollo, y ahorros en costes.

▪ Mejora del mantenimiento por la independencia de datos

En los sistemas basados en ficheros, las descripciones de los datos y la lógica para el acceso a los mismos es parte del código de los programas de aplicación, generando una dependencia entre los datos y las aplicaciones. Un cambio en la estructura de los datos puede requerir cambios sustanciales en los programas de aplicación, incluso en aquellos que no hacen uso de los datos modificados, ya que el cambio repercute en toda la estructura del fichero. Sin embargo, un SGBD separa la descripción de los datos de las aplicaciones, facilitando que estas sean inmunes a modificaciones en las descripción de los datos. Al proporcionar independencia sobre los datos se facilita el mantenimiento, tanto de las estructuras de datos como de las aplicaciones.

▪ Mejora de la concurrencia

En algunos sistemas basados en ficheros, si dos o más usuarios pueden acceder simultáneamente al mismo fichero, es posible que el trabajo de unos interfiera con el del resto, creando riesgos de pérdida de información o de integridad de los datos. Muchos SGBD son capaces de gestionar el acceso concurrente a la base de datos, y son capaces de asegurar que estos problemas no puedan ocurrir, gracias a mecanismos específicos.

▪ Servicios mejorados de recuperación y copia de seguridad

Muchos sistemas basados en ficheros responsabilizan al usuario de mantener medidas para la protección de los datos frente a fallos del sistema o del programa de aplicación. Esto puede implicar, por ejemplo, realizar copias de seguridad diarias de los datos. Así, en caso de producirse un fallo, será posible recuperar la información desde la última copia, aunque se perderán los cambios realizados en el sistema desde el instante de la copia. Por su parte, los SGBD modernos proporcionan facilidades para minimizar la cantidad de trabajo sobre la base de datos perdida en caso de uno de estos fallos.

o Desventajas

▪ Complejidad

El número de funcionalidades que esperamos de un moderno SGBD hace que estos sean aplicaciones *software* de alta complejidad. Los diseñadores y desarrolladores de bases de datos, los administradores y los usuarios finales deben comprender estas funcionalidades para ser capaces de sacarles provecho. Una mala comprensión de estos sistemas puede dar lugar a malas decisiones de diseño, con serias consecuencias para la organización.

▪ Tamaño

Por su complejidad, los SGBD son aplicaciones *software* de gran tamaño, que utiliza un gran volumen de espacio de disco y puede necesitar grandes cantidades de memoria para ejecutarse de manera eficiente.

▪ Costes del SGBD

El coste de un SGBD puede ser muy variable, dependiendo tanto del entorno como de las funcionalidades que deba proporcionar. Puede haber costes de licencia asociados al volumen de información, o al número de usuarios totales, o simultáneos. También existen SGBDs con licencias de *software* libre, que proporcionan funcionalidades comparables a las de productos comerciales. En todos los casos, es necesario tener en cuenta los costes de mantenimiento y operación.

▪ Costes adicionales de *hardware*

Los requisitos de almacenamiento en disco del SGBD y de las bases de datos puede requerir la compara de almacenamiento adicional para la información, sea en dispositivos físicos o en sistemas *cloud.* Además, para conseguir un funcionamiento eficaz puede ser necesario utilizar máquinas más

potentes que en un sistema basado en ficheros. La compra o uso en *cloud* de estos sistemas más potentes puede significar incrementos en los costes del sistema.

- **Coste de conversión**

En algunas situaciones, el coste del SGBD y cualquier ampliación de *hardware* puede ser relativamente bajo en comparación con los costes de transformar las aplicaciones existentes para su funcionamiento con el nuevo SGBD. Este coste incluirá también la formación del personal en el uso de los nuevos sistemas, y posiblemente la contratación de personal especializado para el apoyo en la conversión y manejo de los sistemas. Este coste es uno de las muchas razones por las que algunas organizaciones sigan ligadas a sus antiguos sistemas. El término **sistema legado** se utiliza para referirse a estos sistemas anticuados, y muchas veces con capacidades inferiores al estado actual de la tecnología.

- **Rendimiento**

Típicamente, un sistema basado en fichero está escrito para una aplicación determinada, por ejemplo: facturación. Como resultado, su rendimiento será muy bueno, de manera general. Sin embargo, el SGBD está escrito para ser genérico, y ajustarse a cualquier aplicación en lugar de a una sola. Esto da como resultado que algunas aplicaciones puedan no ser tan rápidas como eran en un sistema “a medida”.

- **Mayor impacto de los fallos**

La centralización de los recursos incrementa la vulnerabilidad de los sistemas. La gran dependencia sobre un la disponibilidad del SGBD por parte de usuarios y aplicaciones, puede provocar el cese en la operativa de una organización ante el fallo de determinados componentes.

8 BIBLIOGRAFÍA BÁSICA

T. Connolly y C.Begg. Database systems: a practical approach to design,implementation, and management (6th ed.). Capítulo 2.