



TEMA - I:

LOS SISTEMAS DE ALMACENAMIENTO DE LA INFORMACIÓN

BD - (1º DAM)

ÍNDICE

1. FICHEROS	5
1.1. INTRODUCCIÓN A LOS FICHEROS	5
1.1.1. ESTRUCTURA DE DATOS.....	5
1.1.2. SISTEMAS DE GESTIÓN DE DATOS.....	5
1.1.3. NOCIONES DE FICHEROS.....	6
1.1.3.1. CARACTERÍSTICAS FÍSICAS Y LÓGICAS.....	6
1.1.4. OPERACIONES REALIZADAS CON FICHEROS.....	7
1.2. DISEÑO DE REGISTROS	7
1.2.1. REGISTROS DE LONGITUD FIJA.....	7
1.2.2. REGISTROS DE LONGITUD INDEFINIDA	7
1.3. CLASIFICACIÓN DE ARCHIVOS	7
1.3.1. TIPOS DE ARCHIVOS SEGÚN SU FUNCIÓN O UTILIZACIÓN	8
1.3.1.1. FICHEROS PERMANENTES O MAESTROS.....	8
1.3.1.2. FICHEROS DE MOVIMIENTOS O TRANSACCIONES.....	8
1.3.1.3. FICHEROS DE MANIOBRA O TRABAJO	8
1.4. MODO DE ACCESO A LOS REGISTROS DE UN ARCHIVO	8
1.5. ORGANIZACIÓN DE LOS REGISTROS DENTRO DE UN ARCHIVO	9
1.5.1. SECUENCIAL.....	9
1.5.2. ORGANIZACIÓN SECUENCIAL ENCADENADA: PUNTEROS.....	10
1.5.3. RELATIVA.....	10
1.5.3.1. DIRECCIONAMIENTO DIRECTO.....	10
1.5.3.2. DIRECCIONAMIENTO INDIRECTO	11
1.5.4. ORGANIZACIÓN INDEXADA.....	12
1.5.5. ORGANIZACIÓN SECUENCIAL INDEXADA	12
1.5.6. ORGANIZACIÓN SECUENCIAL INDEXADA ENCADENADA	12
1.5.6.1. ÁRBOLES B ⁺	12
1.6. CRITERIOS DE SELECCIÓN DE ORGANIZACIONES FÍSICAS.....	13
2. BASES DE DATOS (BBDD)	14
2.1. CARACTERÍSTICAS DE LAS BBDD.....	14
2.1.1. ELEMENTOS QUE FACILITAN EL USO DE UNA BASE DE DATOS	14
2.1.2. ELEMENTOS DE INTEGRIDAD DE DATOS	15
2.2. VENTAJAS E INCONVENIENTES DE LAS BBDD.....	15
2.2.1. VENTAJAS	16
2.2.2. INCONVENIENTES	16

3. SISTEMA GESTOR DE BASE DE DATOS.....	17
3.1. INTRODUCCIÓN	17
3.1.1. EL SGBD COMO INTERFAZ ENTRE EL USUARIO Y LA BASE DE DATOS	17
3.1.1.1. USUARIOS INFORMÁTICOS.....	17
3.1.1.2. USUARIOS FINALES	18
3.1.2. EL ADMINISTRADOR DE LA BASE DE DATOS.....	18
3.1.2.1. FUNCIONES DEL ABD.....	19
3.1.2.2. RELACIÓN DEL ABD CON EL RESTO DEL PERSONAL	20
3.2. FUNCIONES DE UN SGBD.....	21
3.2.1. FUNCIÓN DE DESCRIPCIÓN O DEFINICIÓN.....	21
3.2.2. FUNCIÓN DE MANIPULACIÓN.....	21
3.2.3. FUNCIÓN DE UTILIZACIÓN O CONTROL	21
3.3. COMPONENTES DE SGBD.....	22
3.3.1. NÚCLEO.....	22
3.3.2. DICCIONARIO DE DATOS.....	22
3.3.3. LENGUAJES.....	22
3.3.3.1. LENGUAJES DE IMPLEMENTACIÓN.....	22
3.3.3.2. LENGUAJE DE DEFINICIÓN DE DATOS	23
3.3.3.3. LENGUAJE DE MANIPULACIÓN DE DATOS.....	23
3.3.3.4. LENGUAJE DE CONTROL DE DATOS	24
3.3.4. UTILIDADES.....	24
3.4. ARQUITECTURA DE UNA BASE DE DATOS.....	24
3.4.1. LA ESTANDARIZACION DE LAS BASES DE DATOS.....	24
3.4.2. NIVELES DE ABSTRACCÓN EN UNA BASE DE DATOS.....	24
3.4.2.1. ESTRUCTURA LÓGICA DE USUARIO (ESQUEMA EXTERNO)	25
3.4.2.2. ESTRUCTURA LÓGICA GLOBAL (ESQUEMA CONCEPTUAL)	25
3.4.2.3. ESTRUCTURA FÍSICA (ESQUEMA INTERNO).....	25
3.5. TRANSFORMACIÓN DE LOS DATOS.....	26

1. FICHEROS

- CONCEPTOS: Fichero - Registro - Registro Lógico - Registro Físico – Campo - Claves
- ALMACENAMIENTO
 - **Primario:** Es limitado - Es caro - No puede compartirse una vez en uso - Volátil
 - **Secundario** Mayor tamaño - Menor precio - No requiere flujo continuo de energía

1.1. INTRODUCCIÓN A LOS FICHEROS

1.1.1. ESTRUCTURA DE DATOS

La razón de las estructuras de datos es presentarlos al procesador lo más manipulados posible, de modo que se puedan procesar fácilmente como un conjunto homogéneo. Los tipos de datos son:

1. *BASICAS: BIT, BYTE . . .*
2. *SIMPLES:*
 - Numérico (entero, real)
 - No numéricos (carácter, booleano, usuario)
3. *ESTRUCTURAS DE DATOS COMPLEJAS:*
 - 3.1. *INTERNAS*
 - Estáticas (vectores y tablas o arrays)
 - Dinámicas (listas, pilas, colas, arboles)
 - 3.2. *EXTERNAS*
 - Ficheros (secuencial, directo, indexado)
 - Bases de Datos.

1.1.2. SISTEMAS DE GESTIÓN DE DATOS

Se denomina **DISEÑO LÓGICO DE DATOS** a la apariencia que presentan, para las personas que forman un equipo informático o para los usuarios, los datos almacenados en un ordenador.

Se denomina **DISEÑO FÍSICO DE DATOS** al modo en que se guardan los datos en un dispositivo de almacenamiento (disco, cintas , etc.).

LOS SISTEMAS DE GESTIÓN DE DATOS de clasifican, según la forma en que los recuperan y almacenan en:

- En **SISTEMAS DE GESTIÓN BASADOS EN FICHEROS DE DATOS Y PROGRAMAS**
Los programadores deben definir su organización, el modo de acceso a los registros, su estructura, etc. Lo que implica que los programas sean dependientes de los archivos y viceversa. Esto conduce a serios problemas de repetición de información, ya que, aplicaciones realizadas por distintos programadores organizarán los mismos datos de diferentes maneras, en probablemente archivos diferentes. Los lenguajes empleados para el diseño de programas eran COBOL, C, PASCAL etc.
- Los **SISTEMAS DE GESTIÓN BASADOS EN BASES DE DATOS** Para lograr una gestión más racional del conjunto de datos, surge un nuevo enfoque que se apoya sobre una **BASE DE DATOS**, en la cual los datos son recogidos y almacenados, al menos lógicamente, una sola vez, con independencia de los tratamientos.

1.1.3. NOCIONES DE FICHEROS.

- **Índice de Volatilidad:** Inserciones y Borrados durante un periodo de tiempo respecto al total de registros.
- **Índice de Actividad:** Porcentaje de registros usados durante un periodo de tiempo.

1.1.3.1. CARACTERÍSTICAS FÍSICAS Y LÓGICAS.

Para almacenar la información los SO ofrecen una estructura de datos de alto nivel llamada **archivo** que distribuye los datos en dispositivos externos de almacenamiento. Es un conjunto de informaciones estructuradas en unidades de acceso denominadas **registros**, todos del mismo tipo. De esta forma toda la información manejada por el sistema informático se almacena en archivos. Fundamentalmente, existen dos tipos de archivos: Binarios en **código máquina** y los de **datos**. Los programas en código máquina se utilizan para manipular los archivos de datos.

El **SO** ofrece el tipo abstracto de datos archivo y las operaciones necesarias para definirlo, utilizarlo y eliminarlo, encargándose de acoplar el archivo al dispositivo físico de almacenamiento (cintas, discos, etc.)

En general un **archivo** es una secuencia de bytes cuyo significado lo define su creador. Pueden ser numéricos, de texto, de líneas, de registros, de gráficos, binarios, de datos etc. Estos ficheros se almacenan en disco ocupando consecutivamente los **cluster** (unidad mínima de almacenamiento en disco) necesarios, cuyas direcciones se almacenan en la **FAT** (direcciones de los cluster que forman cada fichero). Los archivos que interesan son los de datos, que se definen como una colección de registros relacionados entre sí. Un registro lógico es una estructura de datos formada por uno o más elementos denominados **campos**, que pueden ser de diferentes tipos y que a su vez pueden estar compuestos por subcampos.

Existen dos tipos de registros:

- **REGISTRO LÓGICO:** Es una estructura de datos formada por uno o más elementos llamados campos que constituyen una unidad para un determinado proceso. Coincide con la definición de registro. El tamaño y formato lo define el programador.
- **REGISTRO FÍSICO:** es una unidad utilizada para las **operaciones de E/S** por el **S.O.** entre la memoria central y los dispositivos periféricos, de manera que el conjunto de datos transferidos en una operación de lectura o escritura se denomina registro físico o **bloque**. El tamaño de este registro depende de las características de cada computadora. Un registro físico además de los datos del usuario, contiene **información del control** del sistema. Ej. una línea de impresión, un sector del disco.

LAS CARACTERÍSTICAS MÁS IMPORTANTES DE LOS ARCHIVOS SON:

- Residen en **soportes externos**, por lo que su existencia no está limitada al tiempo de ejecución del programa que lo crea, sino que permanece cuando este termina.
- Los datos pueden transportarse de un ordenador a otro.
- Tienen capacidad de almacenamiento ilimitada ya que, aunque un soporte tiene capacidad limitada, un archivo puede distribuirse en varios soportes.

1.1.4. OPERACIONES REALIZADAS CON FICHEROS.

- **Simples:**

- Creación
- Apertura
- Consulta
- Actualización: Inserciones Borrados Modificaciones (Mantenimiento de archivos, se deteriora el espacio).
- Cierre

- **Complejas:**

- Recorrido
- Ordenación (fichero a Memoria Principal ó claves a Memoria Principal) .
- Partición.
- Mezcla.
- Renombrado.
- Copia.

1.2. DISEÑO DE REGISTROS

1.2.1. REGISTROS DE LONGITUD FIJA

- La suma de las longitudes de cada campo es siempre la misma
 - a) Igual nº de campos e idéntica longitud de campo en cada registro: *Geografía fija y longitud fija en campos.*
 - b) Igual nº de campos y distinta longitud de campo en cada registro: *Geografía fija y longitud variable en campos.*
 - c) Distinto nº de campos y distinta longitud de campo en cada uno: *Geografía variable y longitud variable en campos.*
- **Borrados:** marcar registro, insertarlo en la lista de disponibles.

1.2.2. REGISTROS DE LONGITUD INDEFINIDA

- Se emplean para optimizar la ocupación de los soportes de almacenamiento.
- Llevan asociados unos **espacios auxiliares** : control, longitud, inicio y final tanto para los campos como para los registros.
- Se pueden diseñar de dos formas, que se pueden combinar :
 - a) **Por separadores de campos o banderas :**
CAMPO 1 | Cter Esp A | CAMPO 2 | Cter Esp A | ... | CAMPO n | Cter Esp A || Cter Esp B | CAMPO 1 | ...
 - b) **Mediante indicadores de longitud :** \sum longitudes = longitud del registro
|| Longitud Campo 1 | CAMPO 1 || Longitud Campo 2 ||...
- **Borrados:** en la lista de disponibles hay que poner además la longitud del registro.

1.3. CLASIFICACIÓN DE ARCHIVOS

- Según su **CONTENIDO** pueden ser programas, archivos de texto, archivo de datos, archivo de gráficos etc. Tienen extensiones diferentes.
- Según su **ORGANIZACIÓN** de la información pueden ser secuenciales, relativos, indexados, secuenciales-indexados y de alto nivel para SGBD.

1.3.1. TIPOS DE ARCHIVOS SEGÚN SU FUNCIÓN O UTILIZACIÓN

1.3.1.1. FICHEROS PERMANENTES O MAESTROS

- Inalterables o casi inalterables.

FICHEROS DE CONSTANTES

- *Tablas matemáticas, tablas periódicas, tablas de descuentos, etc.*

FICHEROS DE SITUACIÓN

- Contienen la situación actual de la información, susceptible de sufrir frecuentes modificaciones.
- *Proveedores, cliente, etc.*

FICHEROS HISTÓRICOS

- Contienen información referente a resultado de operaciones.
- Pueden contener información resumida, o detallada para organizar.
- *Informes anuales, mensuales, etc.*

1.3.1.2. FICHEROS DE MOVIMIENTOS O TRANSACCIONES

- Contienen el resultado de transacciones o modificaciones
- Sirven para consultar o actualizar ficheros permanentes.
- Su vida es corta ; mientras dure el proceso de consulta o actualización.
- Pueden pasar a ser históricos.
- *Anotaciones de ventas, etc.*

1.3.1.3. FICHEROS DE MANIOBRA O TRABAJO

- Contienen información referente a resultados intermedios, resultados definitivos, o bien para suministrar información a otros procesos.
- Estos ficheros se confeccionan en el mismo proceso.
- En la mayoría de los casos desaparecen después del proceso.

1.4. MODO DE ACCESO A LOS REGISTROS DE UN ARCHIVO

Independientemente de cómo estén organizados los datos en un archivo, para los programas, es importante la forma en que puede accederse a los registros.

Existen básicamente tres modos de acceso: secuencial, directo y por índice. Cada organización soporta determinados modos de acceso.

Acceso secuencial

Las operaciones de lectura o de escritura se hacen sobre el registro físicamente contiguo al último que se utilizó. Este modo de acceso es consecuencia de que los primeros dispositivos de almacenamiento que eran soportes secuenciales (tarjetas perforadas, cintas perforadas cintas magnéticas).

Acceso directo

Los registros pueden leerse y escribirse directamente en la posición física que ocupan en el archivo. Para ello, antes se debe calcular dicha posición. Aparece cuando se difunden los soportes de acceso directo (tamaños y discos magnéticos).

Acceso por índice

Los dos modos anteriores son los que proporciona el sistema operativo. Sobre ellos puede implementarse el acceso por índice. Consiste en crear un índice ordenado con las claves del archivo. Para acceder a los registros, se busca secuencialmente la clave en el índice, que lleva asociada la dirección real del registro en el archivo, el cual se lee o escribe directamente.

1.5. ORGANIZACIÓN DE LOS REGISTROS DENTRO DE UN ARCHIVO

La organización de un archivo es la forma en que se disponen los datos para su almacenamiento, recuperación y procesamiento, teniendo en cuenta las características del soporte de almacenamiento (Ej. Cinta sólo secuenciales, discos los tres tipos de organización). El sistema operativo ofrece dos maneras de organizar un archivo que están directamente relacionadas con el modo en que han sido creados sus registros llamadas **secuencial** y **directa**. A partir de estas dos organizaciones y las operaciones facilitadas por los sistemas operativos, se definen otras organizaciones que facilitan la recuperación de la información de la manera más rápida posible.

1.5.1. SECUENCIAL

Con este tipo de organización, los registros se almacenan uno detrás de otro, sin dejar huecos y en el orden en que se van grabando, ocupan posiciones contiguas en la memoria de almacenamiento y sólo se puede acceder a ellos de uno en uno a partir del primero. Si los registros siguen un orden lógico por algún campo, el archivo se denomina **secuencial ordenado**; Ejemplo, *un archivo con los alumnos de una clase ordenados por apellidos*. El único modo permitido de acceso a los registros es el secuencial aunque se puede utilizar en soportes no direccionables o secuenciales (cintas magnéticas) y direccionables (discos magnéticos).

Las **ventajas** de este método radican en que no desaprovechan espacio de almacenamiento, o sea existe la compactación del archivo al no haber espacios vacíos entre los registros. Tiene una elevada **eficiencia** de recuperación de registros cuando el porcentaje de consultados es suficientemente grande.

En cambio, tiene los siguientes **inconvenientes**:

- Para acceder a un registro, hay que leer todos los que hay delante de él, que de media son la mitad de los registros del archivo.
- No se puede insertar un registro en medio de otros dos, ya que sólo está permitido añadir registros al final del archivo, lo que supone que si el archivo está ordenado, haya que reordenarlo.
- No se puede borrar un registro, sólo se marca y ocupa espacio en el dispositivo de almacenamiento.
- Para actualizar un archivo es necesario un archivo de movimientos, trasladar el maestro actualizado a uno nuevo y, posteriormente, cambiar el nombre al nuevo por el del maestro

En la actualidad se utilizan poco los archivos con organización secuencial únicamente se aconseja esta organización para archivos históricos y constantes. Un ejemplo de archivo con organización secuencial puede ser el de abonados a una compañía telefónica, en el que se almacenan los datos que aparecen anualmente en las guías.

1.5.2. ORGANIZACIÓN SECUENCIAL ENCADENADA: PUNTEROS

Una variante de la anterior es la organización secuencial encadenada. Son ficheros de organización secuencial **gestionados mediante punteros** –campos que contienen la dirección de un registro dentro del fichero. Los punteros, son un campo adicional, por lo que **aumenta el tamaño** de los registros, y forman parte integrante de la estructura del registro, indicando cuál es el siguiente o el anterior registro en secuencia lógica y no en secuencia física. Las características de los punteros son: estar en cada registro en una posición fija y definida para todos los registros y poseer una longitud constante.

En estos ficheros **la secuencia física y la secuencia lógica no coinciden**, pudiendo ocurrir que el último registro en secuencia física sea el primero en secuencia lógica y viceversa. La **adición** de registros se realiza al final, pues los registros se almacenan secuencialmente. La **eliminación** de registros se efectúa marcando el registro, de modo que el fichero mantiene su tamaño, ya que el borrado que se produce es lógico, pero no físico.

Esta organización sólo se puede emplear en soporte direccionables.

1.5.3. RELATIVA

Un archivo con organización relativa es un conjunto de posiciones contiguas de memoria secundaria que tienen longitud fija. La información se coloca y se accede aleatoriamente mediante su **posición**, o sea, indicando el lugar relativo que ocupan dentro del conjunto de posiciones posibles. El almacenamiento físico de los registros se realiza mediante el empleo de una **clave que indica la posición** del registro dentro del fichero y la posición de memoria donde está almacenado ese registro.

Sólo pueden usarse soportes direccionables. En cambio es posible acceder a los registros en los modos **secuencial y directo** (en cualquier orden y en cualquier lugar).

Si la clave no precisa ninguna transformación para corresponderse con una posición determinada del soporte, la organización se denomina **directa** o **direcciónamiento directo**. En caso contrario es **aleatoria** o **direcciónamiento indirecto**.

1.5.3.1. DIRECCIONAMIENTO DIRECTO.

Se utiliza cuando las claves tienen **rangos de valores densos**, o sea, claves primarias números consecutivos y tienen pocos huecos entre ellas. Cada valor tiene asociada una dirección en el archivo; por ejemplo, *un archivo de habitaciones de un hotel con clave el número de habitación* (suponiendo que al número de habitación no le preceda un dígito con el número de planta). A este método de transformación de claves en direcciones algunos autores lo denominan **organización directa**. Para calcular la dirección real correspondiente se aplica la siguiente fórmula:

$$\text{dirección real} = \text{valor clave buscada} - \text{clave más pequeña} + 1 \text{ (no empieza por 0)}$$

Por ejemplo, en un campo clave con rango de valores posibles entre 1111 y 5000, la dirección que le correspondería a la clave 2345 sería: dirección real = 2345 - 1111 + 1 = 1235; la clave 1111 = 1111 - 1111 + 1 = 1ª posición;

Las **ventajas** de este tipo de organización son:

- Acceso directo a los registros.
- Permite operaciones de escritura y lectura simultáneamente, ya que primero se localiza el registro y luego se realiza la operación deseada: inserción, eliminación, consulta, modificación.
- El acceso a los datos se realiza de forma secuencial o directa.

Los inconvenientes son:

- Deja gran cantidad de posiciones libres de memoria dentro del fichero, debido a que las claves de los registros pueden indicar posiciones de almacenamiento no contiguas, lo que implica una falta de aprovechamiento del soporte de almacenamiento respecto al número real de registros almacenados.
- Al realizar un acceso secuencial, en una consulta sobre todos los registros del fichero hay que recorrer todas las direcciones aunque estén vacías.

1.5.3.2. DIRECCIONAMIENTO INDIRECTO

El valor del campo clave se transforma en una de las direcciones lógicas válidas mediante un **algoritmo de conversión**. Esta técnica se denomina **hashing**. De esta forma, siempre que se desea acceder a los datos de un registro, antes hay que convertir el valor de su campo clave, aplicando el algoritmo correspondiente. Todos los algoritmos transforman el valor de la clave en un valor numérico y en algunos, el número resultante no está en el rango de direcciones posibles, por lo que hay que adaptarlo multiplicándolo por un factor de ajuste.

Además existe un problema y es cuando **dos registros son sinónimos**, o sea, que teniendo claves diferentes, el algoritmo les asigna la misma dirección, lo que produce una **colisión** entre dos registros.

MANEJO DE COLISIONES

FUNCIÓN DE TRANSFORMACIÓN

HASH PERFECTO Y MÍNIMO

Hash perfecto.

Sería un método Hash que no produzca **ninguna colisión** (hipotético porque aunque estén muy bien diseñado se van a producir). Todos los métodos Hash que se diseñen tienen que aproximar a este Hash perfecto.

Hash mínimo.

Sería un método Hash hipotético que no produciría **ningún desperdicio de memoria**. También todos los diseñados se tienen que acercar al Hash mínimo en el sentido de que produzcan una ocupación de memoria lo mas pequeña posible.

Hash perfecto y mínimo.

Conjuntando estos 2 conceptos anteriores sería un método que, por una parte, no produzca colisiones (perfecto) y tenga un tiempo de acceso mínimo; y por otro, mínimo desaprovechamiento (mínimo).

Cuando nos acercamos al Hash perfecto nos alejamos del mínimo y viceversa. Si la tabla está poco ocupada, menor es el riesgo de colisión y si esta mas ocupada hay más riesgo.

Normalmente se obtiene un buen rendimiento cuando sobredimensionamos la tabla entorno a un 10%, es decir, con un factor de carga 0'9.

1.5.4. ORGANIZACIÓN INDEXADA

La organización indexada nace para suplir las deficiencias de la secuencial (no hay acceso directo a los registros) y de la relativa (se accede directamente a los registros sólo por un campo). En muchos casos conviene recuperar la información accediendo por campos que no son clave. También puede ser necesario recuperar los registros en una secuencia distinta a la que ocupan físicamente.

Se denomina **archivo índice** a un archivo que contiene dos campos: uno con los valores del **campo** por el que se desea recuperar la información y otro con la dirección física (**puntero**) que ocupa el registro en el archivo de datos. El archivo índice se mantiene **ordenado por el campo** de datos, facilitando la búsqueda. Su función es análoga al índice de un libro.

Algunas de sus **características**:

- Un archivo de datos puede tener asociados tantos índices como sea necesario.
- Puede utilizarse una combinación de campos para formar el campo de un índice; por ejemplo, Apellidos + Nombre
- Al actualizar el archivo de datos se debe reorganizar todos sus índices para mantener la coherencia.

Operaciones con índices: Índice → RAM, RAM → Disco

- Búsqueda
- Creación
- Adicción: en el índice y en el fichero de datos
- Borrado: en el índice se pone una marca, y se borran los datos.
- Modificación
 - a)- Afecta a la clave: Borr + Adi ,
 - b)- No afecta: Registros Longitud Fija - Registros Longitud Variable.

Índices que exceden la RAM: se hacen páginas que sí entran, para esas páginas se crean índices.

1.5.5. ORGANIZACIÓN SECUENCIAL INDEXADA

ORGANIZACIÓN SECUENCIAL MAS UN ÍNDICE

1.5.6. ORGANIZACIÓN SECUENCIAL INDEXADA ENCADENADA

1.5.6.1. ÁRBOLES B⁺

1.6. CRITERIOS DE SELECCIÓN DE ORGANIZACIONES FÍSICAS.

El tipo de organización más adecuado a nuestro proceso se elegirá dependiendo de los siguientes factores:

- **VOLUMEN.** Es la cantidad de memoria auxiliar necesaria para almacenar los datos del archivo. Es decir, el espacio que ocupa el archivo en el soporte, medido en caracteres u octetos. Se puede calcular conociendo el número de registros del archivo y el espacio que necesita cada uno por término medio.

*Ej. Un fichero de 1000 registros cada uno con 3 campos de 5bytes, 9 bytes y 4 bytes. Un registro tendría 18 bytes * 1000 registros nos daría un volumen de 18000 bytes/1024 bytes = 17,6 Kilobytes.*

- **CRECIMIENTO.** Es la variación en sentido positivo del volumen del archivo. Se puede medir conociendo el aumento de registros producido por cada tratamiento, expresado en tanto por ciento respecto al total de registros del archivo. Esta medida se llama tasa de crecimiento.

$$Tc = (Altas - Bajas) / n^o \text{ registros archivo} * 100.$$

Se pueden no considerar las bajas y sería: $Tc = Altas / n^o \text{ registros archivo} * 100$

- **ACTIVIDAD.** Es el número de registros consultados o modificados, expresado en tanto por ciento con respecto al número total de registros del archivo en un período de tiempo fijo. Si este porcentaje se determina por cada tratamiento del archivo, se denomina **tasa de consulta o modificación**. Si este porcentaje se calcula por un cierto período fijo de tiempo, se llama **frecuencia de consulta o modificación**.

$$\text{Indice de actividad} = (\text{Modificaciones} + \text{Consultas}) * 100 / n^o \text{ registros archivo}$$

$$\text{Indice de actividad} = N^o \text{ registros utilizados} / n^o \text{ registros del archivo}$$

Si este índice es alto (+ 50%) conviene ficheros con **organización secuencial** porque permite acceso rápido a registros en posiciones físicas contiguas. Si sólo se realizan pocas **consultas** la organización conveniente es la **directa** porque accede de forma rápida a registros situados en cualquier posición del fichero.

Puede emplearse para decidir la organización el comparar los tiempos empleados en acceder a los registros en ambos modos. Se utilizan dos factores:

$$IA1 = NRV/NRT$$

$$IA2 = LR/(TA * VLS)$$

NRV numero medio de registros utilizados en tiempo fijo

NRT número medio total de registros del archivo

LR longitud en bytes del registro

TA el tiempo medio de acceso directo a un registro en segundos

VLS velocidad de lectura secuencial en bytes por segundo

Si $IA1 < IA2$ deberá utilizarse la organización relativa y en caso contrario organización secuencial.

- **VOLATILIDAD.** Es el número de registros dados de alta o de baja respecto al número total de registros del archivo en un período de tiempo fijo. Un archivo es **estático** si tiene un porcentaje bajo de adiciones o supresiones. Un archivo es **volátil** en caso contrario. Si este porcentaje se determina por cada tratamiento del archivo, se denomina **tasa de renovación**. Si se calcula por un cierto período fijo de tiempo se llama **frecuencia de renovación**.

*Indice de volatilidad = (registros insertados (altas) + registros eliminados (bajas)) * 100 / n^o de registros del fichero.*

Indice de volatilidad elevado (o sea más del 50%) interesa **organización directa**, pues permite insertar o eliminar registros rápidamente sin necesidad de reorganizar el fichero. Con una tasa de volatilidad de menos del 50% interesa **organización secuencial**.

2. BASES DE DATOS (BBDD)

La aparición de las BD a partir de los años 60, pretende eliminar los problemas existentes con la gestión de datos mediante archivo, así como estandarizar un modo de gestión de datos más eficaz, universal e independiente de las máquinas y sistemas operativos sobre los que implementarse, generando un sistema de almacenamiento de datos, cuya interfaz de usuario sea fácil de modificar y sencilla de consultar.

Una **BD** es una colección o depósito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real. Los datos, que han de ser **compartidos** por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción (técnicas para cada tipo de datos), han de estar almacenadas con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de **conservar la seguridad** (integridad, confidencialidad y disponibilidad) del conjunto de los datos.

A nivel funcional, una BD se puede considerar como un conjunto de datos almacenados de forma organizada y estructurada en un soporte de información que es manejado por un ordenador o más concretamente por un paquete de software (SGBD). Esta manipulación de los datos puede realizarse en modo **batch** o en modo **interactivo**. En modo batch o por lotes se hacen las operaciones de actualización de los datos (inserción, supresión, modificación). En modo interactivo o conversacional, a través de un terminal, se realizarán las operaciones de consulta e interrogación a la BD. Ambos modos de trabajo pueden realizarse simultáneamente sin problemas, o sea, que un mismo conjunto de datos puede ser compartido al mismo tiempo de varios programas usuarios.

CICLO DE VIDA DE UNA B. D.

2.1. CARACTERÍSTICAS DE LAS BBDD.

2.1.1. ELEMENTOS QUE FACILITAN EL USO DE UNA BASE DE DATOS

- **Mínima redundancia**, que exige evitar **repeticiones innecesarias** de datos en las bases de datos, haciendo que en su diseño se apliquen las reglas de **normalización**. Como resultado del proceso de normalización se crean tablas que contienen el **mínimo** número de campos posibles, relacionándose unas tablas con otras, de modo que se pueda recuperar sin pérdida toda la información disponible, haciendo mínima la redundancia.

- **Herramientas administrativas:**

- **Interfaz de alto nivel** que permite a los programadores definir y crear sus rutinas, vistas o consultas, formularios o informes sin necesidad de conocer la organización interna y externa de la base de datos.
- **Lenguaje de usuario final** que es un lenguaje de consulta de alto nivel o un conjunto de herramientas visuales o asistentes que permiten la generación automática de objetos de la base de datos, así como la consulta de los mismos por los usuarios sin emplear lenguajes específicos de programación.
- **Asistentes** que faciliten o simplifiquen tareas como la instalación, configuración, creación de bases de datos, creación de esquemas SQL, definición de objetos, creación de copias de seguridad, asistentes para la importación o exportación de datos, o para la replicación y sincronización de elementos distribuidos.

2.1.2. ELEMENTOS DE INTEGRIDAD DE DATOS

Es la propiedad de las bases de datos por la que:

- Se puede recuperar la información perdida o dañada cuando se produzcan fallos en el sistema. Hay que implementar un **control de transacciones** que almacene las operaciones sobre la base de datos en el mismo orden en que se han ejecutado. Este control de transacciones (`begin transaction`) debe permitir, cuando haya algún tipo de problema o error, deshacer las modificaciones realizadas, logrando la recuperación automática sin pérdida de datos (`rollback`). Este nivel de integridad viene integrado en los sistemas de gestión de bases de datos comerciales.
- Se mantienen los datos cumpliendo las **reglas de integridad semántica** y aserciones definidas para objeto o tabla de la base de datos (mantener todos los datos cumpliendo las reglas de validación). Hay que incluir las reglas necesarias para evitar la introducción de datos que no se ajusten a los contenidos definidos en la fase de análisis.

2.2. VENTAJAS E INCONVENIENTES DE LAS BBDD.

- **FICHEROS:** En estos sistemas de información existen una proliferación de ficheros, específicos cada uno de ellos de una determinada **aplicación**. Los datos se recogen varias veces y se encuentra repetidos en los distintos archivos. Esta **redundancia**, además de malgastar recursos, origina a menudo divergencias en los resultados.

Este planteamiento produce, además de una ocupación inútil de memoria secundaria, un aumento de los tiempos de proceso, al repetirse los mismos controles y operaciones en los distintos ficheros. Pero más graves todavía son las **inconsistencias** que a menudo se presenta en estos sistemas, debido a que la actualización de los mismos datos, cuando se encuentran en más de un fichero, no se suele realizar de forma simultánea en todos ellos.

- **BASES DE DATOS.** En ella los datos son recogidos y almacenados una sola vez, con **independencia de los tratamientos**. En este enfoque del sistema de información, los datos se organizan y se mantiene en un conjunto estructurado que no está diseñado para una aplicación concreta, sino que, por el contrario, tiende a satisfacer las necesidades de información de toda la organización. Estos sistemas orientados hacia los datos, que mejoran la calidad de las prestaciones de los sistemas informáticos y aumentan su rendimiento, van sustituyendo a los sistemas orientados hacia el proceso que, por su poca fiabilidad, falta de adecuación a la realidad y mal asegurada confidencialidad, han ido perdiendo de forma progresiva la confianza de los usuarios.

2.2.1. VENTAJAS.

Las ventajas de los sistemas de bases de datos son, entre otras, las siguientes:

Ventajas referidas a:

➤ **Los Datos:**

- ◆ **Independencia** de los datos respecto a los tratamientos y viceversa, lo que evita el importante esfuerzo que origina la reprogramación de las aplicaciones cuando se producen cambios en los datos, por ejemplo, en los caminos de acceso.
- ◆ **Mejor disponibilidad** de los datos para el conjunto de los usuarios junto con una mayor transparencia respecto a la información existente.
- ◆ **Menor redundancia**: los datos se repiten de forma mínima e inevitable.
- ◆ **Reducción del espacio de almacenamiento**: La desaparición o disminución de las redundancias, así como la aplicación de técnicas de compactación, lleva en los sistemas de BD a una menor ocupación de almacenamiento secundario.

➤ **Los Usuarios:**

- ◆ **Acceso** más rápido y sencillo de los usuarios finales.
- ◆ **Facilidad** para **compartir** datos entre usuarios.
- ◆ **Mayor flexibilidad** para atender a demandas cambiantes.

2.2.2. INCONVENIENTES.

Las bases de datos no sólo presentan ventajas, sino que también tienen posibles inconvenientes que es necesario valorar antes de tomar una decisión relativa a un cambio en la orientación del SI. Entre estos inconvenientes (referidos al caso de grandes bases de datos) es preciso destacar:

- ◆ **Instalación costosa**: La implantación de un sistema de BD puede llevar consigo un coste elevado, tanto en equipo físico (nuevas instalaciones o ampliaciones), como en el lógico (sistemas operativos, programas, compiladores, etc. necesarios para su uso), además del mismo coste de adquisición y mantenimiento del SGBD.
- ◆ **Personal especializado**: Los conocimientos que resultan imprescindibles para una utilización correcta y eficaz y sobre todo para el diseño y administración de las BD, implican una necesidad de personal especializado.
- ◆ **Falta de rentabilidad a corto plazo**: La implantación de un sistema de BD, tanto por su coste en personal y en equipos como por el tiempo que tarda en estar operativo, no resulta rentable a corto plazo, sino a medio o, incluso a largo plazo.

Cabe señalar, además, que puede encontrarse una resistencia al cambio de un sistema de información apoyado en ficheros clásicos a uno más avanzado enfocado hacia las bases de datos, lo cual se produce como consecuencia de la mentalidad de los usuarios y de la organización en sí misma.

3. SISTEMA GESTOR DE BASE DE DATOS

3.1. INTRODUCCIÓN

La base de datos, como depósito técnico de datos para toda la organización, debe ser capaz de atender las necesidades de los distintos tipos de usuarios que interactúan con ella.

Se puede definir el **Sistema de Gestión de la Base de Datos** (SGBD) como un conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios no informáticos como a los analistas, programadores, o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su seguridad, o sea permiten la administración y gestión de la información de una BD.

Hay que tener en cuenta que en una base de datos existe una gran variedad de usuarios con necesidades diversas que son susceptibles de trabajar simultáneamente con subconjuntos de esta colección de datos y cuyas necesidades varían a lo largo del tiempo. Por tanto, se pone de manifiesto que es imprescindible dotar al sistema de la adecuada **flexibilidad** para que pueda atender las exigencias de todos los usuarios y para que sea capaz de responder a los cambios a un coste no excesivo. Es decir, el SGBD ha de estar diseñado de forma que las ventajas que se han señalado como propias de las bases de datos sean una realidad.

3.1.1. EL SGBD COMO INTERFAZ ENTRE EL USUARIO Y LA BASE DE DATOS

En toda organización se suelen distinguir **tres niveles de gestión**: operacional, táctico y estratégico, de modo que el sistema de información estará integrado por tres subsistemas estructurados jerárquicamente y que se corresponden con cada uno de estos tres niveles.

La desconexión, que muchas veces se produce entre estos tres subsistemas de información (los cuales constituyen a menudo sistemas aislados o, incluso, grupos de aplicaciones sin relación alguna), aumenta el coste global de creación y mantenimiento del sistema de información y produce redundancias e incoherencias; es decir, impide una gestión racional de los datos, que debería estar encaminada a obtener el máximo rendimiento de los mismos, tal como se hace con cualquier otro recurso de la empresa.

La base de datos, como depósito único de datos para toda la organización, debe ser capaz de integrar los distintos subsistemas y aplicaciones atendiendo a las necesidades de los usuarios en los tres niveles, siendo el SGBD el que suministra la interfaz entre el conjunto de los datos y los usuarios.

Los distintos tipos de usuarios de una base de datos pueden clasificarse en usuarios informáticos y usuarios finales.

3.1.1.1. USUARIOS INFORMÁTICOS

Tienen a su cargo las tareas de creación y mantenimiento de la base de datos, así como la realización de los procedimientos y programas que necesiten los usuarios finales.

ADMINISTRADORES

El uso compartido de recursos, propio de una base de datos, obliga a la existencia de un administrador, cuya misión es la **vigilancia y gestión de los datos**. El principal recurso en una base de datos son los datos, y el administrador debe velar para que éstos no se destruyan ni se contaminen, perdiendo su **confidencialidad, disponibilidad e integridad**. Por ello, el administrador debe impedir consultas o actualizaciones no autorizadas y proteger la base de datos contra fallos del equipo lógico o físico. Será el responsable, por tanto, de establecer el sistema de autorizaciones de acceso y deberá coordinar y controlar su uso.

También el administrador tendrá a su cargo la gestión de otros recursos distintos de los datos, como pueden ser el SGBD y otras herramientas relacionadas con el mismo. Deberá ocuparse del **buen funcionamiento de todo el sistema**, sin que se produzcan paradas y de modo que se proporcionen los tiempos adecuados de respuesta.

En muchas organizaciones es **la misma persona o grupo de personas** las que tienen a su cargo las funciones de diseño y de administración, aunque se trata de papeles muy distintos que convendría diferenciar claramente.

ANALISTAS Y PROGRAMADORES

Los analistas y programadores tienen a su cargo el análisis y la programación de las tareas que no pueden ser llevadas a cabo por los usuarios finales, para lo cual han de desarrollar distintos procedimientos y programas que ponen a disposición de los usuarios finales a fin de facilitarles el trabajo.

3.1.1.2. USUARIOS FINALES

Los usuarios finales son aquellos que tienen que acceder a los datos porque los necesitan para llevar a cabo su actividad. A diferencia de los usuarios informáticos, su interés suele estar centrado en el contenido de la base de datos, es decir, en los datos. Existen también distintas clases de usuarios finales.

HABITUALES

Suelen hacer consultas y/o actualizaciones en la base de datos como parte habitual de su trabajo. Utilizan en general **menús** previamente preparados por analistas y/o programadores, de forma que se facilite su interrelación con el ordenador. En otros casos pueden usar **lenguajes sencillos** para el acceso a la base de datos, así como paquetes de programas.

Es preciso distinguir, dentro de este grupo, a los **operadores de entrada de datos**, cuya labor consiste en actualizar la base de datos para lo que se les prepara menús de actualización, aunque con la diferencia respecto a otros usuarios habituales de que tienen exigencias muy estrictas respecto a los tiempos de respuesta, además de no estar interesados en la información contenida en la base de datos.

ESPORÁDICOS

Es un tipo de usuarios muy parecido al anterior en la medida en que necesitan el ordenador a fin de que les preste una ayuda en su trabajo, pero en cambio no lo utilizan habitualmente porque el tipo de actividad que realizan no lo exige. Son, por tanto, usuarios a los que se les tiene que suministrar herramientas sencillas y, en general, potentes.

3.1.2. EL ADMINISTRADOR DE LA BASE DE DATOS

El administrador de la base de datos (ABD) (que también es conocido por las siglas inglesas DBA, Data Base Administrator) es el responsable del **diseño, control y administración** de la base de datos. Se trata en realidad de una función, por lo que sería más correcto decir **Administración de la Base de Datos**, que puede ser desempeñada por una persona o más bien por un grupo de personas dependiendo de la envergadura del proyecto.

3.1.2.1. FUNCIONES DEL ABD

A continuación, describiremos las principales funciones asignadas al ABD, sin referirnos a si deben separarse o no en grupos o personas distintas, ya que ello depende de la estructura del sistema de información (por ejemplo, características de uso, distribución, etc.) y del marco organizativo (por ejemplo, mayor o menor descentralización) de cada institución. Será, por tanto, responsabilidad del ABD:

- **La estructura de la base de datos**, en el sentido de determinar qué información va ser necesario almacenar en la misma, después de haber analizado los requisitos de los distintos usuarios (en algunos casos esta función se considera como de Administración de datos, es decir, del contenido, y se separa del resto de las funciones de administración más relacionadas con aspectos informáticos).
- **La descripción conceptual y lógica de la base de datos**, una vez especificados los requisitos de la información es preciso realizar el diseño conceptual de la base de datos para, después, adecuar la estructura conceptual a un SGBD específico y, por tanto, también a un modelo convencional de datos concreto.
- **La descripción física de la base de datos**, encontrando una estructura interna que soporte el esquema lógico y los objetivos de diseño con la máxima eficiencia de los recursos de máquina. Se trata de una labor que se extiende a lo largo de la vida de la base de datos, ya que este ajuste nunca puede darse por finalizado y, en tanto exista la base de datos, el ABD tendrá que variar parámetros, reorganizar los datos, modificar estructuras de almacenamiento, realizar nuevas distribuciones de los ficheros en los soportes, etc.
- **Las especificaciones y vistas, o subesquemas, para los programas**; del estudio de los requisitos de los usuarios se obtendrá un conjunto de necesidades en cuanto a procesos a realizar sobre los datos; con esta información será preciso definir las vistas externas y la normativa necesarias para los programas de aplicación. Aunque la función de programación de las aplicaciones no es en absoluto la competencia del administrador, sí lo es, sin embargo, el dictar las normas y especificaciones que tienen que cumplir las aplicaciones en su acceso a la base de datos.
- **La estrategia de transición** del sistema existente (informatizado o no) al nuevo sistema de información soportado en una base de datos. El ABD deberá decidir sobre la posible puesta en marcha en paralelo del nuevo sistema con el antiguo, las fases de implantación del mismo, los controles necesarios, la construcción de prototipos, etc. Todas estas decisiones habrán de tomarse en función de los objetivos marcados y de forma que se cause el mínimo trastorno a los usuarios.
- **Los procedimientos de explotación y uso**, es decir, establecer la normativa necesaria para la utilización de la base de datos, el modo de solicitar el acceso a la misma, su actualización, etc.
- **Los aspectos relativos a la seguridad**, incluidos los procedimientos de control y las auditorías.

En definitiva, el administrador interviene en todas las etapas del ciclo de vida de una base de datos representando en todas ellas un papel fundamental que a veces no se reconoce en toda su amplitud.

3.1.2.2. RELACIÓN DEL ABD CON EL RESTO DEL PERSONAL

Para cumplir estas funciones del apartado anterior, el ABD tendrá que interactuar con todo el personal de la organización:

- **Usuarios en general:** con ellos tendrá que especificar los requisitos que debe cumplir la base de datos, estudiar las necesidades que tienen y la mejor manera de satisfacerlas. Además deberá obtener su conformidad respecto a estándares y a la documentación.
- **Directivos:** que le fijarán los objetivos de la BD para que ésta responda a los objetivos generales de la empresa y se integre en los planes estratégico, táctico y operativo de la misma, proporcionando la información necesaria para la elaboración de dichos planes y para su puesta en marcha. Además, el diseño y explotación de la BD requerirán muchos recursos que deberá asignar la dirección de la empresa; ya hemos señalado la importancia de involucrar a los directivos en el desarrollo del sistema de información, e insistimos en que el apoyo de éstos es premisa indispensable para conseguir el éxito del proyecto.
- **Analistas y programadores:** a los que el ABD deberá proporcionar información acerca de la estructura de los datos, especificaciones para las aplicaciones que vayan a desarrollarse, etc. Por su parte, estos usuarios deberán informar al administrador sobre qué modificaciones o qué nuevos datos precisan para que éste estudie las medidas a adoptar.
- **Suministradores:** el ABD tendrá que estar en estrecho contacto con ellos para corregir problemas que puedan producirse en el sistema de base de datos, y, por otra parte, estar al corriente de nuevas herramientas, interfaces, programas, etc., que estén desarrollándose y que sean de interés en el entorno de la base de datos.
- **Personal de formación:** el ABD deberá estar en estrecho contacto con este personal a fin de preparar los planes de formación adecuados sobre bases de datos a todos los niveles de la empresa, completando los conocimientos de los técnicos y de los usuarios y difundiendo la filosofía de las BD.

3.2. FUNCIONES DE UN SGBD

Las funciones esenciales de un SGBD son las de descripción, manipulación y utilización, que no se corresponden de forma lineal con las distintas vistas o niveles de abstracción:

3.2.1. FUNCIÓN DE DESCRIPCIÓN O DEFINICIÓN.

Esta función debe permitir al administrador de la base especificar los elementos de datos que la integran, su estructura y las relaciones que existen entre ellos, las reglas de integridad semántica, los controles a efectuar antes de autorizar el acceso a la base, etc., así como también las características de tipo físico y las vistas lógicas de los usuarios. Esta función, llevada a cabo por el *Lenguaje de Descripción o definición de Datos (LDD)*, propio de cada SGBD, debe suministrar los medios para definir las tres estructuras de datos: externa, lógica global e interna, especificando las características de los datos a cada uno de estos niveles, o sea participan en esta función, los tres niveles de abstracción:

- A nivel físico se definen las condiciones físicas de las estructuras de datos, espacio en disco, tamaño del espacio de tablas o de páginas, unidades de disco, tablas, campos, índices etc.
- A nivel lógico permiten la creación de los objetos para los que se ha reservado espacio y definido características a nivel físico, bien desde herramientas o bien desde el LDD.
- A nivel externo se encargan de definir/eliminar usuarios, roles y sus permisos.

(*Ejemplos: CREATE, DROP, CHECK, CONSTRAIN...*)

3.2.2. FUNCIÓN DE MANIPULACIÓN

Una vez descrita la BD es preciso cargar los datos en las estructuras previamente creadas, por lo que la BD está ya dispuesta para su utilización. La función de manipulación permite a los usuarios de la base la actualización (añadir, suprimir o modificar los datos) y la recuperación (buscar), siempre de acuerdo con las especificaciones y las normas de seguridad dictadas por el administrador. La función de manipulación se llevará a cabo por medio de un *Lenguaje de Manipulación de Datos (LMD)*, que facilita los instrumentos necesarios para la realización de estas tareas.

(*Ejemplos: SELECT, INSERT, DELETE, UPDATE..*)

3.2.3. FUNCIÓN DE UTILIZACIÓN O CONTROL.

Se trata de una función que reúne todas las interfaces que necesitan los diferentes usuarios para comunicarse con la base y proporciona un conjunto de procedimientos para el administrador, entre los que se encuentra el *Lenguaje de Control de Datos (LCD)*. En la mayoría de los SGBD existen funciones de servicio, como cambiar la capacidad de los ficheros, obtener estadísticas de utilización, cargar archivos, etc.; principalmente, las relacionadas con la seguridad física (copias de seguridad, rearranque en caso de caída del sistema, etc.) y de protección frente a accesos no autorizados, las cuales se encuentran comprendidas en la función de utilización.

LCD: No aparece explícitamente definido.

(*Ejemplos: GRANT, REVOKE...*)

3.3. COMPONENTES DE SGBD.

Un SGBD está dividido en módulos que llevan a cabo sus funciones asociadas. La comunicación entre módulos se lleva a cabo a través de **interfaces**.

3.3.1. NÚCLEO.

Como interfaz de la BD con el SO le corresponde transformar las instrucciones de petición de datos que le llegan en órdenes que el S.O. sea capaz de entender y gestionar. El núcleo está soportado en un modelo de datos.

Son programas transparentes al usuario. Entre las tareas que realizan destacan:

- Controlan la integridad y seguridad
- Implementan las funciones de comunicación entre niveles
- Facilitan la independencia de los datos
- Gestionan el diccionario de datos
- Proporcionan el soporte necesario para los programas de utilidad y los lenguajes.

3.3.2. DICCIONARIO DE DATOS.

Se sitúa por encima del núcleo, ya que a este nivel se produce la transformación de las solicitudes de los clientes en instrucciones inteligibles para el servidor. Para el **control de los objetos de las BD** se definen en ésta dos elementos que pueden aparecer asociados o no, el diccionario de datos y el directorio de datos. Además se ha introducido un nuevo concepto, catálogo, para referirse al mismo. Aunque los tres son sinónimos, se refieren a conceptos diferentes.

Un **diccionario de datos (DD)** contiene información sobre los datos almacenados, o sea, sobre los elementos que conforman una BD legibles por los usuarios. Es una base que describe a otra base por eso se llama *metabase*. Al contenido de un DD se le llama *metadatos*. Al administrador de la BD le corresponde ser el responsable de su creación y mantenimiento.

3.3.3. LENGUAJES.

Las distintas funciones que ha de cumplir un SGBD hacen necesario disponer de diferentes lenguajes y procedimientos que permitan la comunicación con la base de datos; unos están orientados hacia la función (definición, manipulación o control), otros dirigidos a diferentes tipos de usuarios o de procesos que acceden a la BD y otros según el **modelo de datos** en el que se fundamenta la BD.

3.3.3.1. LENGUAJES DE IMPLEMENTACIÓN.

En el cual se escriben los procedimientos o programas de aplicación que han de utilizar los datos de la BD. Al lenguaje de utilización se le suele llamar "huésped" del de implementación o bien sublenguaje de datos; al lenguaje de implementación se le llama "anfitrión" del leng. de utilización, debido a que las sentencias del leng. de utilización suelen estar incluidas en el de implementación, generalmente como llamadas a procedimientos externos con una serie de parámetros.

En general, los usuarios informáticos, como el administrador de la base de datos, programadores, etc., requerirán medios potentes y flexibles mediante los cuales consigan definir, extraer y manipular los datos de la base en el lenguaje de programación que estén habituados a manejar (**lenguaje anfitrión**). El programador deberá, por tanto, poder hacer llamadas desde un programa recurriendo a las facilidades que le proporciona el SGBD para relacionarse con la base. Al conjunto de sentencias de manipulación del SGBD que permiten el acceso a la base se las llama sublenguaje de datos y también **lenguaje huésped**.

3.3.3.2. LENGUAJE DE DEFINICIÓN DE DATOS

(DDL Data Definition Language). El Administrador de la BD ha de disponer de instrumentos que le permitan describir los datos con facilidad y precisión especificando sus distintas estructuras; es lo que se denomina lenguaje de definición de datos. Estos son **autocontenidos** y no necesitan apoyarse en ningún lenguaje de programación.

El SGBD tendrá que facilitar medios para describir:

- La **estructura lógica global** (asignar nombres a los campos, a los agregados de datos, a los registros etc., estableciendo sus longitudes y características, así como las relaciones entre estos elementos; especificar los identificadores e indicar las restricciones semánticas que se han de aplicar a los diferentes objetos descritos. Ej.

CONSTRAINT, CHECK ...en SQL).

- Para hacer las especificaciones relativas a la **estructura interna**, volúmenes, crecimiento previsto, tipos de registros más accedidos con indicaciones sobre número medio de accesos, relación entre actualizaciones y consultas etc. Para ello el administrador especifica características respecto a la estructura física que tiendan a conseguir una máxima eficiencia en el almacenamiento (clusters, bloques, índices, tablas, hash...) y en la recuperación de los datos de la base. Para ello, tendrá que disponer de un lenguaje de definición de la estructura interna o, simplemente, deberá de dar valores a ciertos parámetros. Ej. INITIAL tamaño, NEXT tamaño, MAXEXTENTS tamaño etc. en la creación de los TABLESPACES; CREATE TABLE etc. En muchos SGBD se suministra automáticamente una estructura interna por defecto, que es la que el sistema considera más adecuada para la estructura lógica global definida.
- Y para declarar las **estructuras externas** que sean requeridas para el desarrollo de las diferentes aplicaciones. El SGBD debe poner a disposición de los usuarios medios que les permitan recuperar o actualizar los datos contenidos en la base de acuerdo con la visión lógica o estructura externa (vista) que precise cada aplicación. El lenguaje de definición de estas vistas externas podría ser análogo al de la descripción lógica global, sin embargo algunos SGBD tienen lenguajes muy distintos para estos dos niveles. Al definir una estructura externa es preciso darle un nombre e indicar qué datos y qué interrelaciones de la estructura lógica global se encontrarán en la misma.

Esta definición de los datos debe ser compilada para dar lugar a una representación orientada a la 'máquina' que es la que utiliza el SGBD en tiempo de procesamiento. La representación de los datos obtenida en este proceso de compilación es almacenada en otro componente del SGBD denominado Diccionario de Datos.

3.3.3.3. LENGUAJE DE MANIPULACIÓN DE DATOS

(DML Data Manipulation Language). Para cumplir los objetivos asignados a la función de manipulación se ha de contar con lenguajes que ofrezcan a los usuarios la posibilidad de referirse a determinados conjuntos de datos, que cumplan ciertas condiciones (criterio de selección), como que un atributo tenga un determinado valor, o que un conjunto de atributos y valores satisfagan cierta expresión lógica. Además del criterio de **selección**, a veces, es preciso indicar la estructura externa que se desea actualizar o recuperar. Ej SELECT nombre FROM tabla (directamente) o view (parte externa).

Una vez especificados el criterio de selección y los datos a actualizar (**inserción, borrado, modificación**) o recuperar, el SGBD debe ocuparse de acceder al correspondiente soporte físico de donde se extraerán los datos definidos para su transferencia a un dispositivo de salida, o en donde se insertarán, modificarán o borrarán los datos si se trata de una actualización.

3.3.3.4. LENGUAJE DE CONTROL DE DATOS

Existe un sublenguaje del DDL llamado Lenguaje de Control de Datos (LCD o DCL Data Control Language), y se utiliza para controlar el acceso a la información de la BD definiendo privilegios y tipos de acceso, así como para el control de la seguridad de los datos. *Ej. GRANT*

3.3.4. UTILIDADES.

Son aplicaciones que facilitan el trabajo a los usuarios y programadores. Tienen la característica común de tener un interfaz fácil de entender. Se basan en menús que guían al usuario para conseguir el objetivo final. La mayoría de los SGBD incluye:

Generador de menús. Diseña el interfaz de usuario de una aplicación.

Generador de informes. Presenta datos en pantalla o impresora con un formato predefinido o fácil de definir sin conocer lenguajes de BD ni de programación.

Generador de formularios. Genera pantalla de diálogos que presentan ítems y permite la introducción de información, bien por teclado, bien por botones.

3.4. ARQUITECTURA DE UNA BASE DE DATOS

3.4.1. LA ESTANDARIZACION DE LAS BASES DE DATOS

En España, **AENOR CTN 71/SC 21/ GT 3.**

La oportunidad y conveniencia de la estandarización de bases de datos es un tema controvertido, ya que una prematura fijación de estándares puede coartar posteriores desarrollos. Una **normalización a posteriori** tendría una incidencia favorable en el desarrollo de las bases de datos al no introducir cortapisas que dificulten el avance en distintas direcciones, pero será muy difícil (más bien imposible) imponer en la práctica unas normas a sistemas que han sido ya desarrollados y se encuentran en el mercado. Por el contrario, una **estandarización previa** orientará a los diseñadores y será más fácil de aplicar, pero probablemente no dejará que surjan nuevas ideas y será un freno a la imaginación de los creadores de SGBD.

3.4.2. NIVELES DE ABSTRACCIÓN EN UNA BASE DE DATOS.

El Study Group on Data Base Management Systems del ANSI/X3/SPARC (American National Standard Institute) propone la descripción de la arquitectura de un Sistema de Bases de Datos de tres niveles de representación de datos: externo, conceptual e interno.

En los sistemas de información se puede observar la existencia de dos estructuras distintas:

- **Lógica**, vista de usuario.
- **Física**, forma en la que se encuentran los datos en el almacenamiento.

En las bases de datos aparece un nuevo nivel de abstracción que se ha denominado de diversas maneras: **nivel conceptual**, estructura lógico global, esquema, etc. Esta estructura intermedia pretende una representación global de los datos que se interponga entre las estructuras lógica y física y que sea independiente, tanto del equipo como de cada usuario en particular.

Los tres niveles de abstracción en una BD se corresponden a otras tantas formas distintas de contemplar ésta. La lógica de usuario o esquema externo es la visión que tiene de la base de usuario en particular, la estructura lógica global responde al enfoque del conjunto de la empresa (lo que podría ser la visión del administrador), y la estructura física (o el esquema interno) es la forma en la que se organizan los datos en el almacenamiento físico. Una BD tiene un único esquema interno y un único esquema conceptual pero puede tener varios esquemas externos.

3.4.2.1. ESTRUCTURA LÓGICA DE USUARIO (ESQUEMA EXTERNO)

Debido a que un esquema externo es la visión que tiene de la base de datos un usuario en particular o un grupo de usuarios, en él deberán encontrarse reflejados sólo aquellos datos e interrelaciones que necesite el correspondiente usuario (son las consultas, informes, formularios). También habrán de especificarse las **restricciones** de uso, como puede ser el derecho a insertar o a borrar determinados datos o el acceso a los mismos, etc.

Habrá tantos esquemas externos como exijan las diferentes aplicaciones; un mismo esquema externo podrá ser visto por varios usuarios. Cada visión individual se denomina **subesquema** o **vista**. Al crear un subesquema, es posible mezclar campos de distintos registros, omitir campos, cambiar el orden de los campos, añadir campos que puedan ser calculados a partir de los descritos en el esquema conceptual etc. Cada usuario verá una **apariencia de BD**, esquema externo diferente, según sea el nivel de acceso que se le haya concedido. Ej. Acceder a las tablas de clientes y facturas para obtener listados de facturación por clientes.

3.4.2.2. ESTRUCTURA LÓGICA GLOBAL (ESQUEMA CONCEPTUAL)

En el esquema conceptual, que también se conoce como esquema de la BD o Modelo de Datos, por ser la **visión global** de los datos, deberá incluirse la descripción de todos los datos e interrelaciones entre éstos, las restricciones de integridad y de confidencialidad, sin importar la forma ni el lugar en el que se almacenarán y recuperarán los datos. Contiene los datos elementales (**campos**), los datos compuestos (**registros**), las **relaciones** existentes entre campos elementales y compuestos, valores válidos para cada campo, las **reglas** que rigen el funcionamiento de la empresa etc. En este nivel es donde se generan las órdenes de control de integridad y las restricciones o valores permitidos para los campos.

3.4.2.3. ESTRUCTURA FÍSICA (ESQUEMA INTERNO)

Está orientado a la forma en que se almacenan las tablas en memoria. Depende esencialmente de la memoria disponible y de los dispositivos de almacenamiento de acceso directo que se vayan a utilizar. En este momento es cuando se decide el lenguaje que se va a manejar y el equipo físico en el que se implantará. Aunque el contenido del esquema interno es muy dependiente de cada SGBD, podemos distinguir dos clases de aspectos que deben especificarse en él:

Estrategia de almacenamiento. En este apartado se incluye

- La **asignación de espacios** de almacenamiento para el conjunto de datos.
- También deberá indicarse la **estrategia de emplazamiento de los datos** que ha sido utilizada para optimizar tiempo y espacio en memoria secundaria, o sea, los archivos que contiene la información, su organización, la forma de acceder a los registros (**índices**),
- El **tipo y la longitud de los registros** y de los campos que los componen, los campos clave etc.
- Por último, deberán aparecer aspectos como el tratamiento de los **desbordamientos**, etc.

Caminos de acceso. Incluimos en este apartado la especificación de clave primarias y secundarias, así como la de índices o punteros entre registros o incluso claves de ordenación.

3.5. TRANSFORMACIÓN DE LOS DATOS

Para describir un mismo grupo de datos un sistema puede gestionar varios niveles de esquemas, para lo cual el SGBD debe poder garantizar la transferencia de los datos desde el formato correspondiente de un nivel al formato correspondiente a otro nivel; este proceso se llama transformación de los datos (**data mapping**) o mapeo, que es la integración de los tres niveles (externo, lógico y físico). Un componente del SGBD, el **sistema de control de base de datos (DBCS)** es el encargado de transformar los datos requeridos por los programas de aplicación en registros físicos a leer, hacer la petición al S.O. y cuando la información está disponible, transferirla al área de trabajo del programa que la solicitó.

En un SGBD con tres niveles existirán dos niveles de transformación:

- **Externa/conceptual:** permite el paso de una vista externa específica a la vista conceptual y viceversa.
- **Conceptual/interna:** permite el paso de la vista conceptual a la BD almacenada y viceversa.

El SGBD debe conocer las **relaciones entre los niveles** para poder efectuar de forma automática el mapeo de datos de un nivel a otro. Corresponde al DBA cuando defina los esquemas, indicar la forma en que unos esquemas se deducen de otros por medio de **las reglas de correspondencia** (mapping, rules); son los parámetros que definen los procedimientos de mapeo de los datos de un nivel de esquema a otro nivel. Estas reglas (al igual que los diferentes esquemas) se almacenan después de su compilación en el diccionario de datos con comentarios opcionales.

El SGBD es el único que tiene una visión global de la BD, por tanto los programas no leen ni graban directamente los ficheros de la BD sino que piden al SGBD que lo haga, este las convierte en operaciones de acceso por separado a los ficheros de la BD, funciones realizadas por los métodos de acceso que son rutinas del S.O. Cuando un usuario determinado (con un determinado nivel externo) emite una solicitud de acceso a la BD mediante un lenguaje de manipulación de datos específico **una consulta** por ejemplo), lo que hace el SGBD es:

1. **Un usuario solicita ver determinados datos** de una tabla creando una consulta por medio de un comando del lenguaje de manipulación.
2. **Verificación del esquema externo** para ese usuario. El SGBD intercepta la petición y la interpreta, a la vez que evalúa la capacidad del usuario para realizar la operación solicitada
3. **Aceptación del esquema externo.** Si la autorización ha sido concedida, el SGBD consulta el subesquema apropiado para ver si la petición puede ser satisfecha.
4. **Transformación de la solicitud** al nivel conceptual. Determinar los registros conceptuales implicados en la petición
5. **Verificación del esquema conceptual**
6. **Aceptación del esquema conceptual.** Determinar los registros del modelo interno asociados a los conceptuales, o sea, se manejan las tablas que contienen los datos solicitados por la consulta.
7. **Transformación de la solicitud al nivel interno.** El SGBD transforma la petición de acceso lógico en una petición de acceso físico y la pasa al S.O.
8. **Selección de la tabla** objeto de la consulta.
9. **Ejecución de la consulta**
10. **Transformación del nivel interno a conceptual.** El S.O. ejecutará los comandos oportunos sobre los dispositivos de almacenamiento físicos, tomando o depositando los datos a través del buffer.
11. **Transformación del nivel conceptual a externo.** El SGBD enviará a la zona de trabajo del usuario los datos organizados según el tipo del registro externo que se esté utilizando
12. **Se muestran los registros** correspondientes al usuario. El SGBD devolverá al usuario un código de respuesta asociado al tipo de operación realizada, informándole de si ésta ha sido llevada a cabo satisfactoriamente, o si por el contrario ha habido algún problema.

En este proceso es el **Diccionario/Directorio de datos** el encargado de las transformaciones entre las distintas vistas. El diccionario contiene las definiciones del esquema –base de datos- y de las transformaciones a realizar.

EJERCICIOS

1. Explicar en qué consisten las fragmentaciones interna.
2. Dar un ejemplo de los distintos tipos de archivo según su función.
3. Describir la búsqueda de un registro en las organizaciones de archivos secuenciales y en las organizaciones relativas.
4. Un fichero secuencial que contiene 10.000 registros se somete a 3 procesos mensuales en el que se produce una media por proceso de: 1000 altas, 2000 bajas, 1500 modificaciones. Determinar los índices de volatilidad, de actividad y la frecuencia de utilización anual.
5. Se dispone de tres ficheros A, B, C .

A tiene 50.000 registros y se efectúan 3 procesos anuales. En el primer proceso 1000 altas, 500 bajas, 300 modificaciones. En el segundo proceso hay 1500 altas, 1500 bajas y 1500 modificaciones. En el tercer proceso 500 altas, 1500 bajas, 1500 modificaciones. Tener en cuenta las bajas y altas anteriores.

B tiene 10.000 registros y hay un proceso mensual los 12 meses del año: 1000 altas, 1000 bajas, 100 modificaciones.

C tiene 80.000 registros y se actualiza una vez al año. Se dan 1000 altas, 5000 bajas y 6500 modificaciones.

Determinar cual es el más volátil por proceso y cuál tiene el índice de volatilidad anual menor.

6. Un fichero F1 se actualiza secuencialmente 2 veces al mes. Al iniciar el mes contenía 10000 registros y en la 1^a actualización se producen 1000 altas, 2000 bajas, 2000 modificaciones. En la segunda se producen 2000 altas, 1000 bajas, 3000 modificaciones. Otro fichero F2 se actualiza secuencialmente 1 vez al mes. Contiene 20.000 registros y se han producido 1000 altas, 4000 bajas y 2000 modificaciones.

DETERMINAR:

- a. mayor índice volatilidad mensual (es F1)
 - b. Mayor actividad mensual (es F1)
 - c. Mayor actividad por proceso (es F1 el segundo proceso)
 - d. Frecuencia de utilización semestral (es F1)
 - e. Tasa de crecimiento
7. Se dispone de dos ficheros F1 con 60.000 registros y 4 procesos anuales

- 1º. 1500 A / 100 B / 4000 M
- 2º. 2000 A / 1500 B / 2000 M
- 3º. 500 A / 500 B / 500 M
- 4º. 1000 A / 1500 B / 2000 M

F2 30.000 registros con 2 procesos mensuales

- 1º. 1000 A / 2000 B / 3000 M
- 2º. 1500 A / 1000 B / 2500 M

Se pide: - El más volátil por proceso

El más activo por proceso

El de mayor frecuencia de utilización mensual.