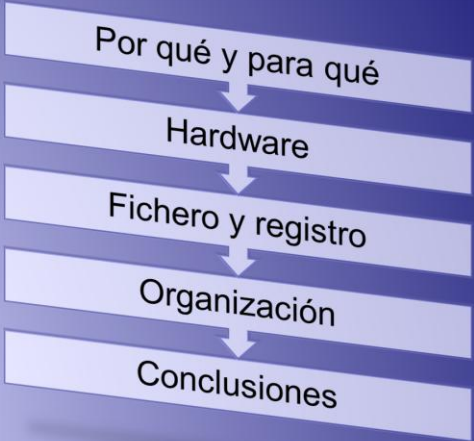



ficheros
Tema 1B



Por qué y para qué
↓
Hardware
↓
Fichero y registro
↓
Organización
↓
Conclusiones

lsi Departament d'Informàtica i Sistemes de Comunicacions Universitat d'Alacant
Universitat de Alicante

El objetivo de este tema es dar unas nociones básicas del fichero electrónico.



por qué y para qué

- Persistencia
 - almacenar información
 - para su posterior recuperación y proceso
 - el fichero ha existido siempre
 - la tecnología disponible determina el soporte
 - es la parte física de una base de datos



hardware

- En momentos de la historia se descubren
 - papel
 - tarjeta perforada
 - cinta magnética
 - disco magnético
 - flexible
 - rígido
 - óptico, flash, ...
- Hablamos de almacenamiento secundario



fundamentos de las bases de datos

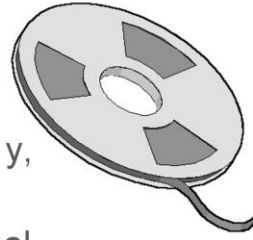
<http://ocw.unizar.es/humanidades/historia-del-proximo-orientemodulo-1/tablas2.jpg>

La idea central es que la "forma" del archivo la determina el estado tecnológico del momento.

A partir de ahora hablaremos de lo que se conoce como almacenamiento secundario. Existe, en computación, lo que se conoce como almacenamiento primario y terciario, además.

hardware: la cinta

- almacenamiento masivo
- soporte analógico
 - usado en grabaciones de audio y, posteriormente video
 - modulador-demodulador de señal
- adelante-atrás
 - hay que recorrer toda la cinta hasta encontrar lo que buscamos



From Computer Desktop Encyclopedia
© 1999 The Computer Language Co. Inc.



fundamentos de las bases de datos

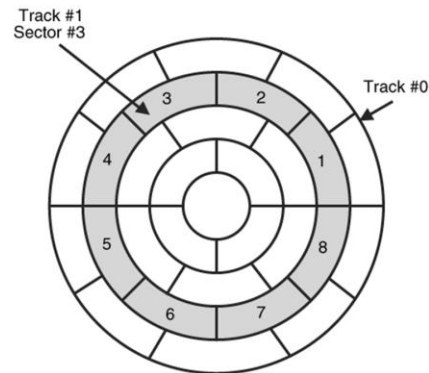
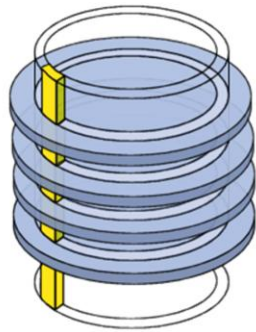


La cinta fue un gran avance respecto de la tarjeta perforada. No deja de ser un soporte analógico que necesita convertir la señal a un formato discreto que represente la información binaria.

La cinta, básicamente, va hacia adelante y hacia atrás. De otra forma, encontrar un dato implica empezar desde el principio e ir leyendo hasta llegar a lo buscado. Todo lo más, se puede ir hacia atrás y hacer lo mismo, todo depende de lo sofisticado que sea el controlador del cacharro.

hardware: el disco

- Cilindro/Cabezal/Sector
 - el direccionamiento de los datos
- bloque = {sectores}
 - tamaño mínimo de un archivo, lo que se transfiere
 - definido por el SO



fundamentos de las bases de datos

El bloque determina la cantidad mínima de información que se transfiere del disco duro a la RAM. Esto implica que cuando solicitamos un dato concreto no recuperamos solo eso sino todo el bloque y, ya en RAM, es cuando lo localizamos y usamos. Es más eficiente que hacerlo byte a byte (¿por qué?)

fichero y registro

- Guardar los datos en el ordenador
 - fichero, archivo, ... el nombre no es casual
 - registro, ficha, campo, ... tampoco

fichero



registro



campo



fundamentos de las bases de datos



fichero y registro

- Texto = registro único o sin registro
 - no estructurado
 - conocemos el principio y el final del fichero
 - **recuperar** n bytes "siguientes", **rebobinar**, ...

<p style="text-align: justify">De lo que se trata es de instalar una herramienta cliente en vuestro ordenador, no un sistema de gestión de bases de datos local ni, por supuesto, base de datos alguna. Con el cliente y la url oportuna podéis acceder a los servidores de la asignatura. Otra cosa ya **son las cosas de la red, la vuestra y la** de la universidad, ahí no se puede hacer nada. Lo bonito de los clientes que vamos a ir comentando es que la mayoría son programas poco exigentes en recursos de máquina, incluso alguna cosa hay para iPhone e iPad —yo es que lo leo tal cual me enseñaron, “i-fone” e “i-pá”. En cualquier caso, el objetivo es poder practicar desde casa. Por familiaridad, aunque ya decimos que iremos mostrando otras opciones, parece que SQL Developer es la elección más evidente.</p><p style="text-align: justify">En primer lugar, nada de susto, es muy sencillo.</p><p style="text-align: justify">SQL Developer es un cliente de acceso a servidores de bases de datos que está instalado en los laboratorios asignados a la asignatura.



fundamentos de las bases de datos

El tipo de fichero y organización más básico. De hecho, se puede decir que no hay estructura

fichero y registro

- de registros
 - Campos y tipos de datos
 - el registro representa a un objeto
 - los campos definen sus características

```
struct empleado {  
    char nombre(30);  
    char dni(9);  
    int sueldo;  
    int codtrabajo;  
    char departamento(20);  
}
```



fundamentos de las bases de datos

Ahora el fichero tiene una estructura, el registro, que a su vez se define por sus campos.

fichero y registro

- Cuestión de espacio

- longitud fija
 - necesita separadores
- longitud variable
- mixto

9		H	o	n	o	r	i	o						A	l	i	c	a	n	t	e					
10		P	e	p	e									S	a	n	V	i	c	e	n	t	e	d	e	

9	\$	H	o	n	o	r	i	o	\$	A	l	i	c	a	n	t	e	\$	10	\$	P	e	p	e	\$	S	a	n
		V	i	c	e	n	t	e		d	e			R	a	s	p	e	i	g	\$							



fundamentos de las bases de datos

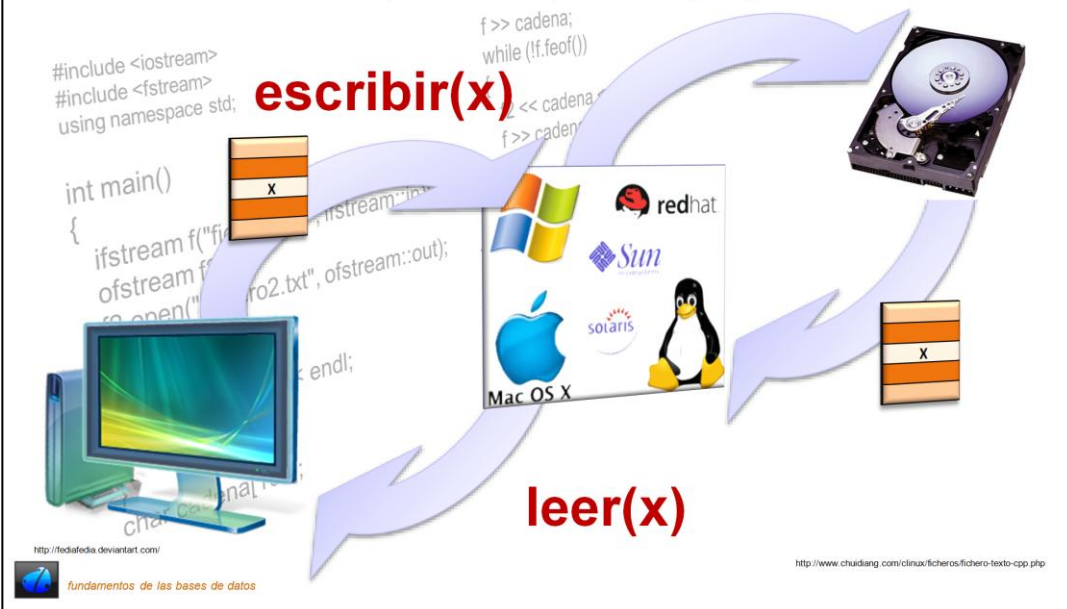
El primer ejemplo, simple: al ser todos los registros y campos del mismo tamaño es fácil recuperar la información (¿por qué?). El problema es el espacio malgastado y que, seguramente, ciertos datos no nos cabrán. Tiene más sentido incluso en un disco duro puesto que podemos pedir al sistema operativo que se posicione en un punto concreto del fichero. En cinta, es posible que también, siempre que las bobinas se muevan con cierta precisión.

En el segundo ejemplo (cuidado, se supone que el espacio de cada dato está asociado a su tipo, los ejemplos usan intencionadamente cadenas de caracteres todo el tiempo para que se vea más claro): aquí hemos redefinido el campo de la ciudad, hemos sido más generosos con el tipo de datos para que nos cupiera San Vicente del Raspeig, asumiendo que lo que gastamos de más con una definición de *string* más largo lo ahorraremos en otros registros.

Hay más formas de montar la estructura del registro, la más obvia es la mezcla de campos de longitud fija y variable.

fichero y registro: operar

- transferencia por bloques o páginas

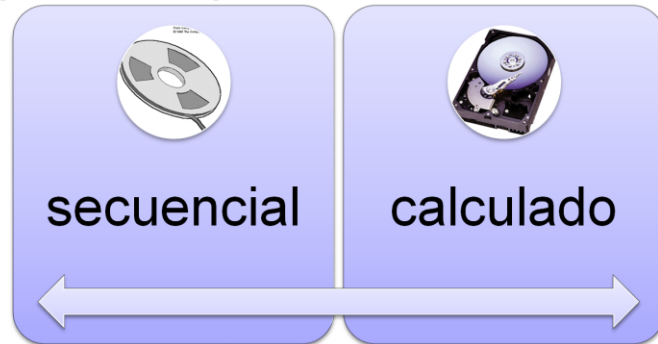


Nuestro programa solicita una operación sobre un fichero al SO, éste busca en el disco y localiza, devolviendo los datos a la RAM del ordenador, donde el programa los manipula. Las operaciones posibles son leer y escribir (borrar, puede hacerse de varias formas y no necesariamente de forma física, marcando para reorganización posterior...) El detalle importante es que el SO recupera bloques (es habitual leer datos cercanos al primero, el disco duro es el dispositivo más lento que el procesador y la RAM y se necesita dejar al mínimo los accesos a disco duro...) y no registros individuales. Es en RAM donde se accede a un registro concreto.

El proceso por bloques tiene un efecto importante en el espacio "malgastado", otro factor a tener en cuenta cuando se organiza un fichero.

organización

- La tecnología disponible marca la estructura y rendimiento del fichero
 - el almacenamiento secundario, en cualquier caso es lento
 - se busca mejorar el rendimiento mediante la organización lógica de los registros

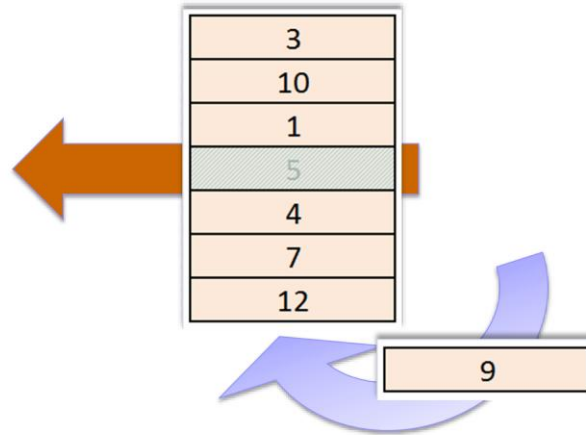


fundamentos de las bases de datos

La organización de un fichero, dentro de las posibilidades que nos da el SO, es decisión nuestra, cómo queremos mantener la información en función de si queremos optimizar la generación de listados o las altas y bajas, por ejemplo.

organización: secuencial

- No me preocupa el orden

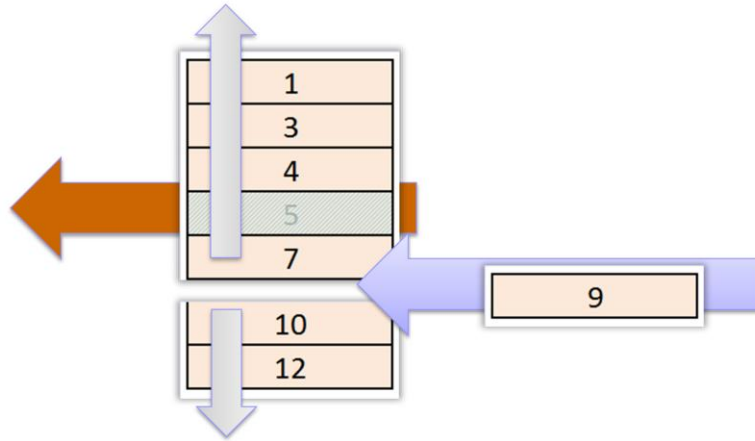


fundamentos de las bases de datos

Se inserta, simplemente, al final del fichero.

organización: secuencial ordenado

- Me preocupa el orden

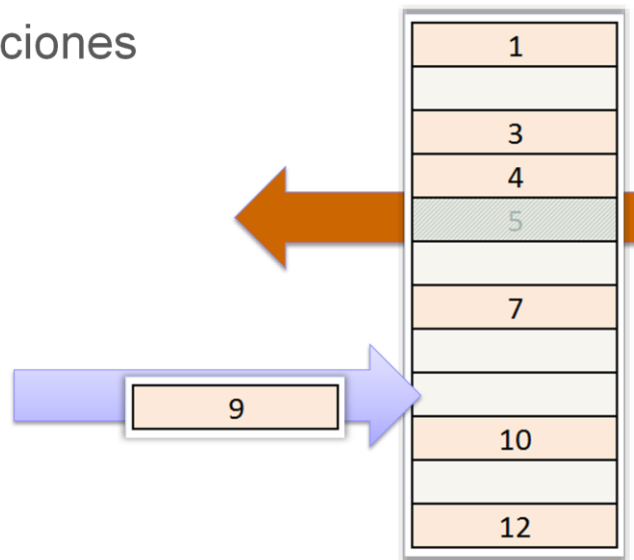


fundamentos de las bases de datos

En un fichero secuencial ordenado, cuando quiero insertar un dato, debo "hacerle sitio" en el punto correcto del fichero. Los borrados pueden seguir siendo marcas lógicas, no borrados físicos.

organización: aleatoria

- Me preocupa el orden
- Y las inserciones



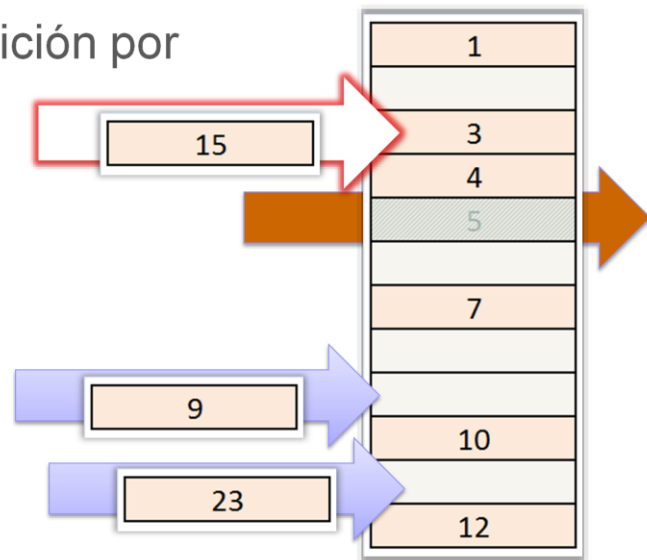
fundamentos de las bases de datos

La organización aleatoria o directa asume que hay una cantidad máxima de registros y que cada registro tiene un "sitio reservado" dependiendo de su clave de identificación.

organización: aleatoria: hashing

- El espacio
- Calcular la posición por una función

- $h = n \bmod 13 + 1$

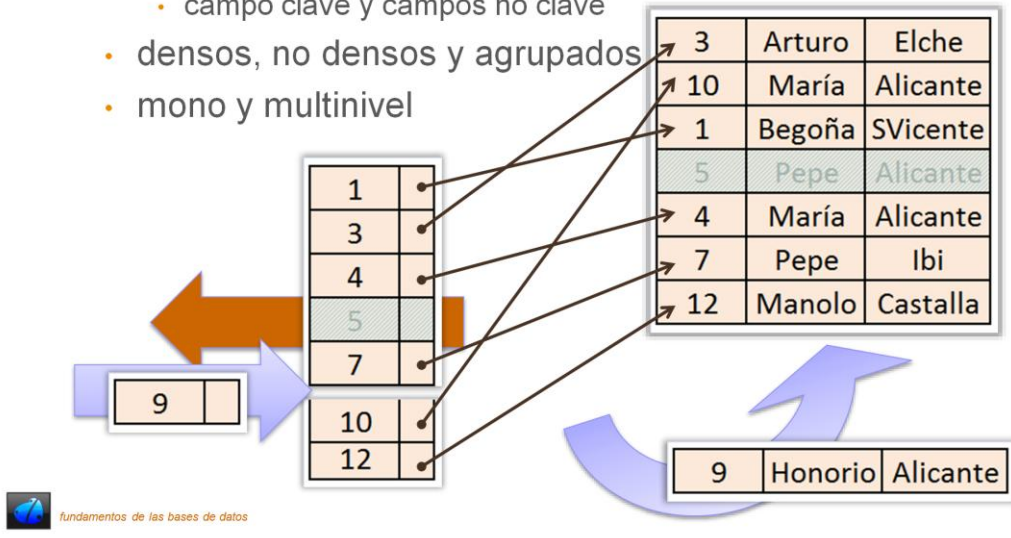


fundamentos de las bases de datos

El *hashing* es más flexible que el directo puro (que no deja de ser un *hashing* básico) porque permite almacenar más valores clave. El problema son las colisiones, para lo que hay varias técnicas que lo solucionan, aunque suponen una degradación del rendimiento del acceso y mantenimiento del fichero.

organización: aleatoria: indexada

- Eficiencia balanceada
 - primarios y secundarios
 - campo clave y campos no clave
 - densos, no densos y agrupados
 - mono y multinivel

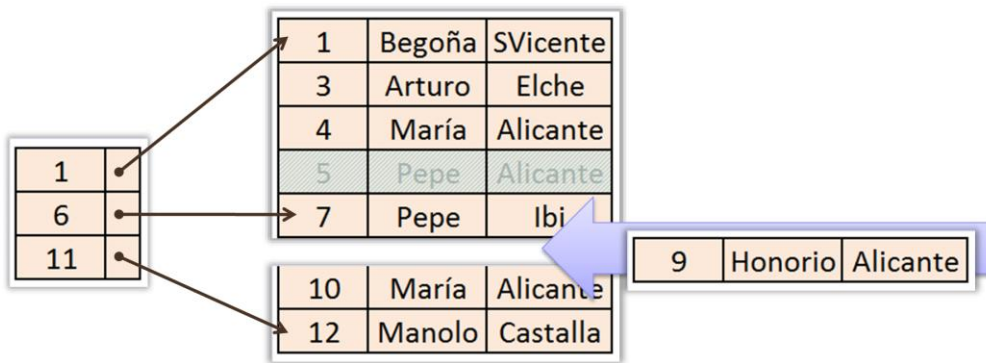


La organización indexada es más flexible. Se basa en que, aparte del fichero de datos, existe un fichero de índice que se utiliza para localizar con rapidez el registro buscado.

Hay distintos tipos de índices que favorecen las inserciones o las consultas y que determinan si el fichero de datos ha de estar ordenado o no. Téngase en cuenta, nuevamente, que desde el disco duro no recuperamos un único registro sino una página que contiene un número más o menos elevado de ellos, con la esperanza de que nuestras siguientes peticiones ya estén cargadas en RAM o en un buffer y no tengamos que "mover" nuevamente el disco.

organización: aleatoria: indexada

- Índices no densos

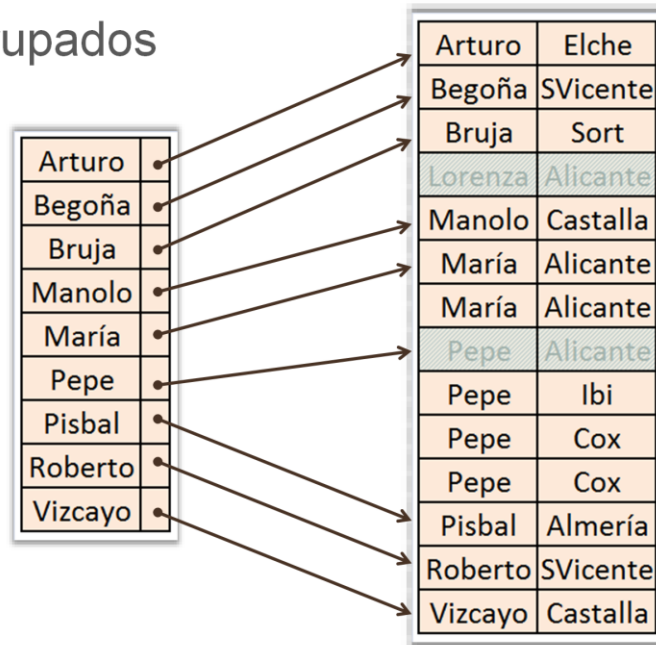


fundamentos de las bases de datos

Los índices no densos obligan a que el fichero de datos esté ordenado por una clave.

organización: aleatoria: indexada

- Índices agrupados

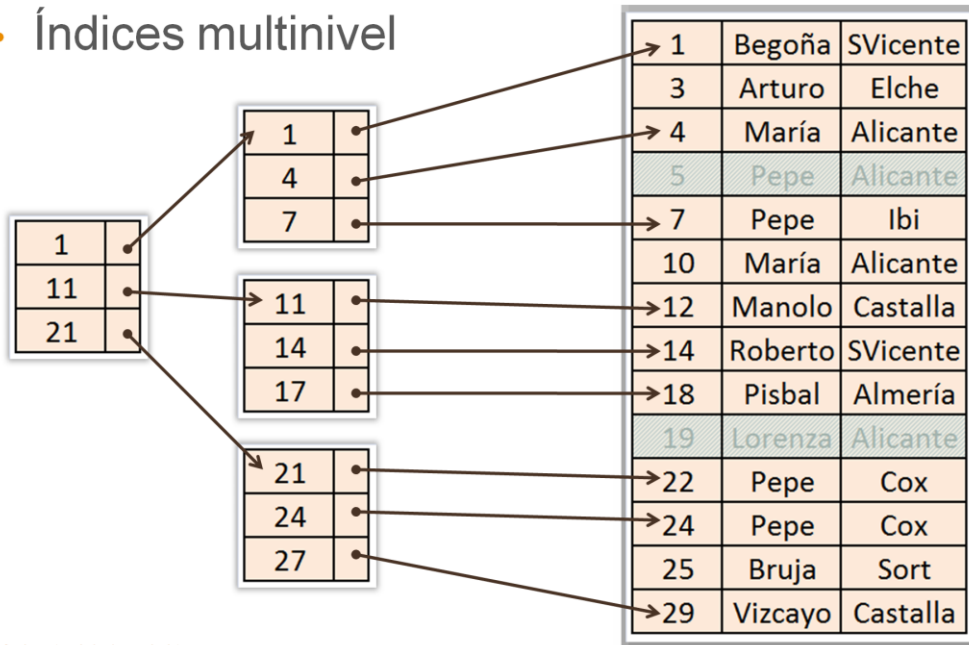


fundamentos de las bases de datos

El índice es más pequeño y la búsqueda en él más rápida.

organización: aleatoria: indexada

- Índices multinivel

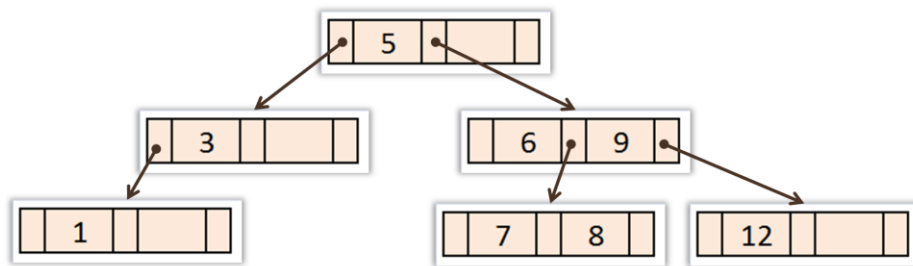


fundamentos de las bases de datos

Se busca ayudar al algoritmo de búsqueda binaria restringiéndola sucesivamente a menos datos.

organización: árboles B, B+

- Índices multinivel dinámicos
 - muy usados en los SGBD actuales



fundamentos de las bases de datos

Con algoritmos bastante más complicados de mantenimiento de esta estructura, la consulta, inserción y borrado son mucho más rápidas. Hay que situarse en un contexto de ficheros que estén almacenando miles de registros para entender que esta nueva organización de índices sea realmente rentable frente a las anteriormente vistas. En ficheros con muchas actualizaciones es necesario hacer una reorganización periódica de los índices.

conclusión

- La persistencia de datos
- Fichero = {registros} = {campos}
- La tecnología del momento marca la organización posible del fichero
- Los ficheros siguen utilizándose masivamente
 - los SGBD almacenan sus datos en ficheros
 - habituales árboles B, B+, hashing...
 - estas organizaciones no son conocidas ni importantes para el usuario (sí para el administrador)



fundamentos de las bases de datos

Esta página pretende ser una miniguía, minipista, de por donde van a ir las preguntas del cuestionario

conclusión: conceptos

- almacenamiento primario, secundario y terciario
- organización y acceso
 - ventajas e inconvenientes en la inserción-borrado-búsqueda
- lectura/escritura por bloques en disco
- fichero de datos
- fichero de índices



fundamentos de las bases de datos

Esta página pretende ser una miniguía, minipista, de por donde van a ir las preguntas del cuestionario

conclusión: referencias

- fbddocs.dlsi.ua.es/lecturas

