

Organización y acceso a los datos (continuación)

Bases de Datos 2 - Tecnólogo de Informática

- 1 Recapitulemos
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

- 1 **Recapitulemos**
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

- 1 **Recapitulemos**
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

¿Que hay a nivel físico?

- Discos
 - Conjunto de bloques de tamaño fijo que son direccionables a través de alguna forma de dirección por alguna instrucción específica. (Read)
- Partición
 - Subconjunto contiguo de los bloques de un disco.

¿Que hay a nivel físico?

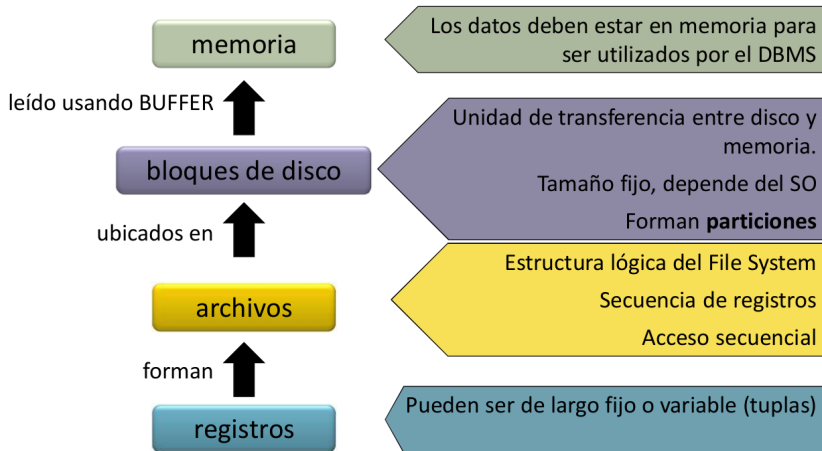
- File Systems (FS)

- Estructura de datos que facilita el acceso del sistema operativo (SO) a los distintos bloques de la partición. Típicamente se construye con comandos como Format, Newfs, mkfs, etc., dependiendo del SO.

- Archivo

- Estructura lógica que se construye sobre el FS. Es la que usualmente utilizan los programas de aplicación.
- Típicamente se ven como una secuencia de caracteres o como una secuencia de registros delimitados por algún carácter especial o de tamaño fijo y el acceso es secuencial.

Organización física de datos



¿Cómo utilizan los DBMS esto?

- Típicamente toman una partición del disco o un gran archivo y usan dicha área como su “disco”.
- Re implementan:
 - Estructuras de datos y sus algoritmos de manipulación.
 - Algoritmos de ordenación.
 - Mecanismos de buffering y paginado.
- Cuando se habla de estructuras se debe pensar en como organizar los datos para una (o más) tablas

- El acceso secuencial a archivos es bueno para un DBMS?
- Consideremos las siguientes tablas y la siguiente consulta:
 - EMPLEADOS(nombre,edad,salario,Dno)
 - DEPARTAMENTOS(Dno,nombreDpto,piso,gerente)

```
SELECT e.nombre, d.piso  
FROM EMPLEADOS e, DEPARTAMENTOS d  
WHERE e.Dno = d.Dno AND e.salario > 30000
```
- ¿Cómo se ejecutaría esta consulta si sólo contáramos con acceso secuencial a los archivos?

Formas de acceder a los registros

- Supongamos que hay un archivo de empleados y otro de departamentos:

$\pi_{\text{nombre,piso}} (\sigma_{\text{salario} > 30000} (EMPLEADOS * DEPARTAMENTOS))$

Recorrer todo el resultado de la selección y devolver ciertas posiciones de cada registro.

Recorrer todo el resultado del join y devolver los registros que cumplen la condición.

Para cada uno de los registros de uno de los archivos buscar en el otro aquellos registros que tienen igual valor en campos de igual nombre y agregarlos al resultado.

- En el Join, si M es el tamaño de empleados, y N es el tamaño de departamentos \Rightarrow se necesitan $M*N$ operaciones de acceso a disco.

Son muy costosas!!

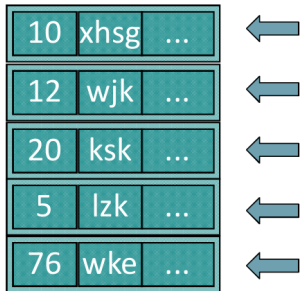
- ¿Cómo se puede mejorar?
 - Usando otras estructuras de datos sobre disco que faciliten el acceso.

INDICES

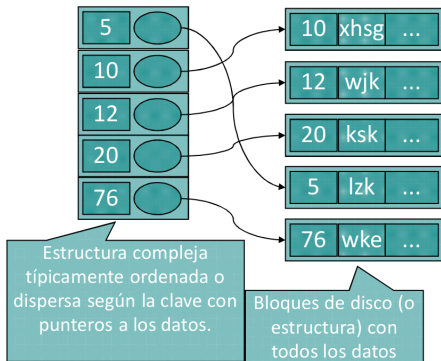
- 1 Recapitulemos
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

- 1 Recapitulemos
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

Archivo Secuencial



Archivo Indexado (clave el 1^{er} atributo)



- **Clave de ordenación** (search key), también denominado clave principal, es un atributo utilizado para buscar registros en un archivo.
- Un **archivo índice** consiste de registros, llamados **entradas de índice** (index entries), de la forma

clave de ordenación	puntero
---------------------	---------

- Comúnmente los archivo índice son mucho mas pequeños que el archivo original.

- Físicos vs. Lógicos
 - Referencias a ubicación en disco o referencias a índices físicos.
 - Consideraremos sólo los físicos.
- Ordenados vs. No ordenados
 - Requieren o no del ordenamiento de los datos por el atributo a indexar.
- Densos o no densos.
 - Hay entradas de índice para todo valor de la clave de ordenación o solo para algunos valores.
- Simples vs. Multi-nivel

- 1 Recapitulemos
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

- Índice primario (o principal)
 - Sobre clave primaria.
 - Por cada clave: dirección en el disco (bloque o bloque+offset).
- De agrupamiento (cluster index)
 - Sobre un atributo que **no es clave primaria**.
 - Por cada **valor distinto**: dirección en el disco al primer registro del grupo (bloque o bloque+offset).

Índices con datos ordenados

Índice primario

nombre	edad	salario	depto
Alberto	28	10000	Depto1
Ana	30	12000	Depto2
Juan	35	12000	Depto3
...
Lucia	40	20000	Depto3
Luis	34	12000	Depto1
...
Sandra	30	15000	Depto2

Alberto	
Ana	
Juan	
...	...
Lucia	
Luis	
...	...
Sandra	

Índices con datos ordenados

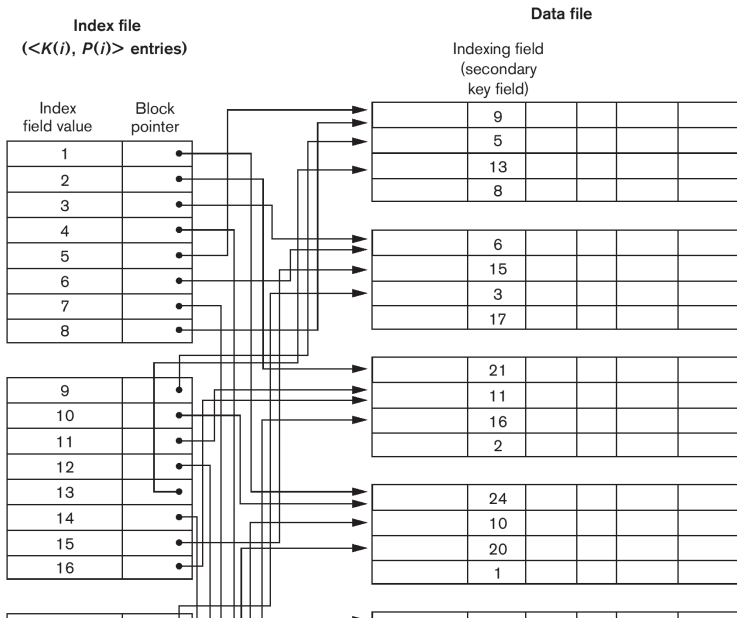
Índice cluster (por atributo *Edad*)

28	
30	
34	
35	
40	

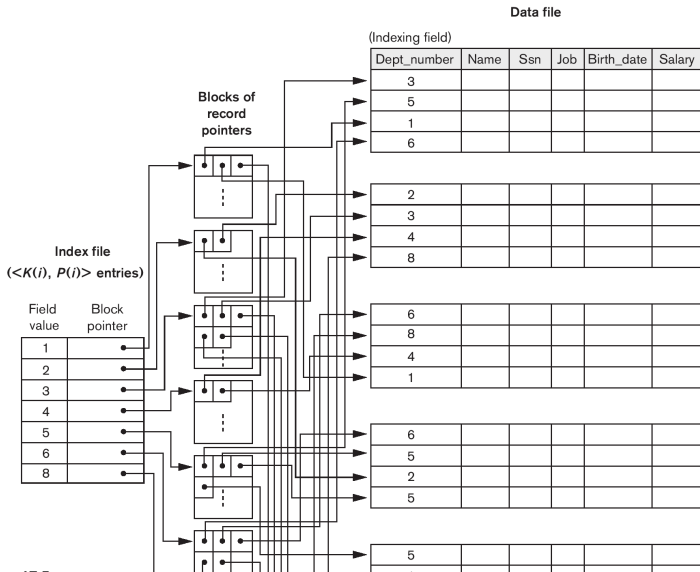
Nombre	Edad	salario	depto
Alberto	28	10000	Depto1
Ana	30	12000	Depto2
Sandra	30	15000	Depto2
Luis	34	12000	Depto1
Juan	35	12000	Depto3
Lucia	40	20000	Depto3
...
...

- Se crea una estructura auxiliar ordenada por el campo a indexar.
- Por cada valor: un puntero al bloque donde se encuentra el valor,
 - Si el valor no se repite: un solo bloque.
 - Sino, una lista. Tengo tantos punteros como bloques contengan al valor.

Índice secundario denso (con puntero a bloque)



Índice secundario denso usando un nivel de indexación (con puntero a registro)



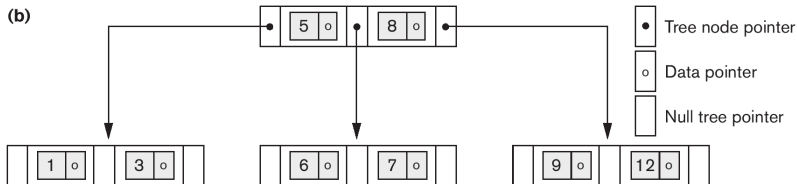
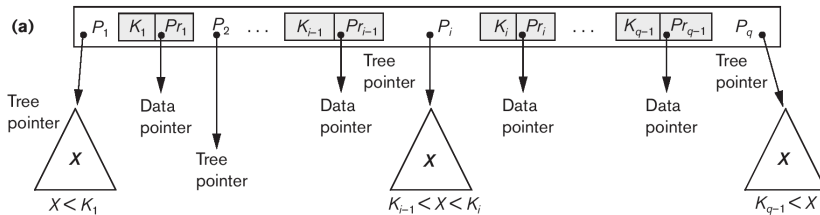
- 1 Recapitulemos
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

- Hash
 - Muy buen comportamiento en la inserción y en la recuperación por condiciones de igualdad.
 - No funciona bien para condiciones con relaciones de orden.
- Árboles B y B+
 - Buen comportamiento en recuperación tanto por condiciones de igualdad como de orden.
 - Buen comportamiento en la inserción.
 - Ocupa más disco.

- Es un árbol de búsqueda que puede estar vacío o aquel cuyos nodos pueden tener varios hijos tal como muestra el dibujo.
- Se dice que un árbol-B es de orden M (es el máximo número de hijos que puede tener) si satisface las siguientes propiedades:
 - Cada nodo tiene como máximo M hijos.
 - Cada nodo (excepto raíz y hojas) tiene como mínimo $M/2$ hijos.
 - La raíz tiene al menos 2 hijos si no es un nodo hoja.
 - Un nodo no hoja con q hijos, $q \leq M$, contiene $q-1$ elementos almacenados (y por tanto $q-1$ punteros a datos).
 - Los hijos que cuelgan de la raíz (r_1, \dots, r_m) tienen que cumplir ciertas condiciones:
 - El primero tiene valor menor que r_1 .
 - El segundo tiene valor mayor que r_1 y menor que r_2 , y así sucesivamente hasta el último hijo que tiene valor mayor que r_m .

Árbol B

EN ed7 fig 17.10

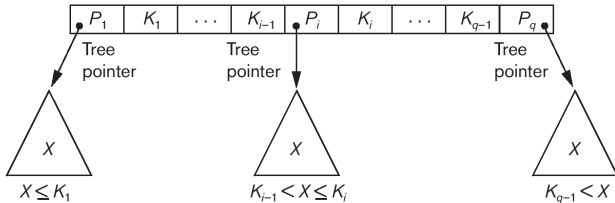


- La diferencia del Árbol B+ con respecto al árbol B es que no hay datos (puntero a) en los nodos internos. Los datos sólo están en las hojas.
- En cada nodo interno, el puntero de la izquierda de la clave apunta ahora a los menores o iguales, y el de la derecha a los mayores que la clave.
- Por otro lado, como los datos están en las hojas, entonces se puede organizar otra estructura sobre ellos que facilita la recorrida ordenada.

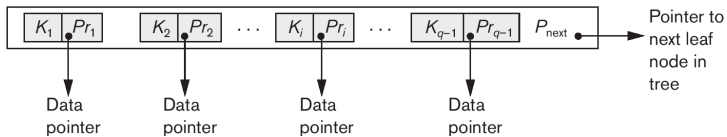
Árbol B+

EN ed7 fig 17.11

(a)



(b)



- Los DMBSs implementan diferentes estrategias para organizar los registros de una tabla:
 - Registros desordenados con acceso secuencial.
 - Registros ordenados con acceso secuencial.
 - Registros indexados por la clave primaria con hash.
 - Registros indexados por la clave primaria con árbol B+.
 - Registros indexados por otro atributo con un índice clúster.

- 1 Recapitulemos
 - Organización y Acceso a Datos
- 2 Estructuras de indexación
 - Conceptos básicos
 - Índices ordenados
 - Estructuras Hash y Árbol B/B+
- 3 Definición de índice en SQL

Definición de índice en SQL

- Crear un índice,

```
CREATE INDEX nombre_indice ON nombre_relacion  
    (lista_atributos)
```

- Utilizar **create unique index** para especificar e imponer indirectamente la condición de que la clave de búsqueda es una clave candidata.
 - No se necesita si el DBMS soporta la restricción de integridad SQL **unique**.
- Eliminar un índice,

```
DROP INDEX nombre_indice
```
- La mayoría de los sistemas de BD permiten especificar el tipo de índice.