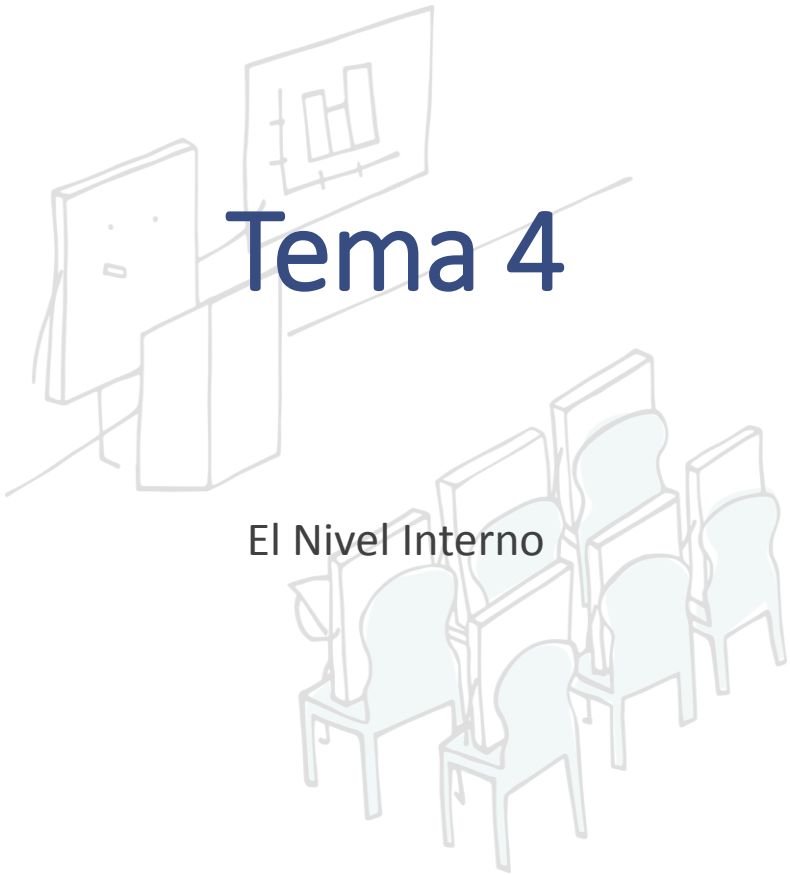
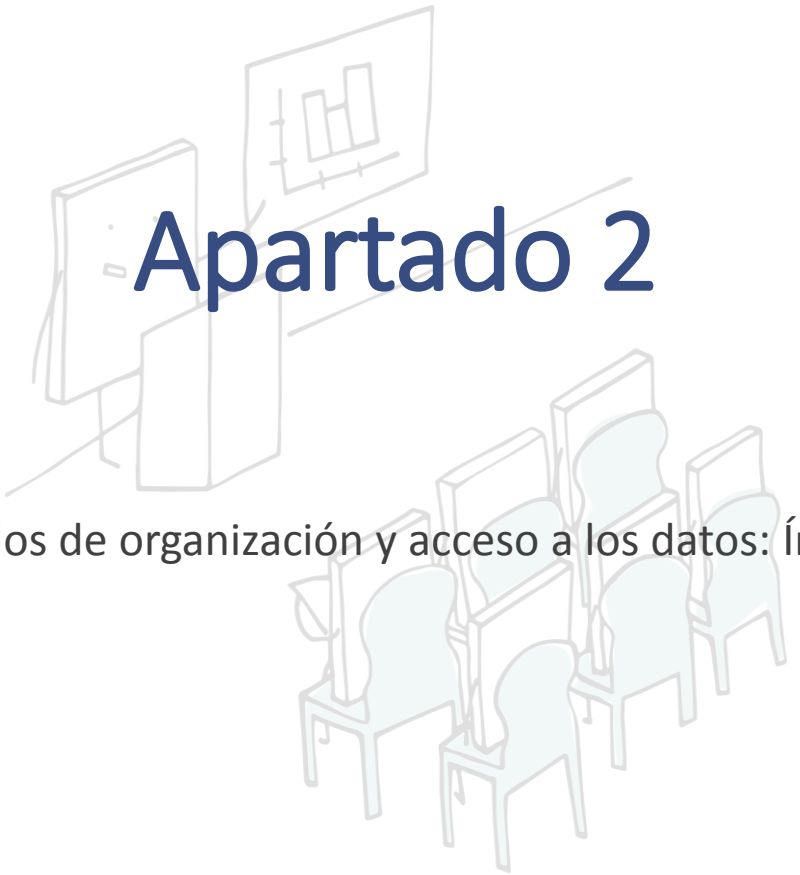


Tema 4

El Nivel Interno



Apartado 2

Métodos de organización y acceso a los datos: Índices

Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Organización y Métodos de Acceso

- OBJETIVO: Minimizar el número de accesos a disco → minimizar la cantidad de páginas de BD involucradas en una operación de BD.
  - Ninguna de las organizaciones presentadas es mejor en términos absolutos.
  - Criterios básicos para medir la "calidad" de una organización son:
    - Tiempo de acceso a los datos requeridos.
    - Porcentaje de memoria ocupada por los datos requeridos con respecto a las paginas de BD que los contienen.
  - Trabajaremos a dos niveles:
    - Organización de registros de datos a nivel de almacenamiento
    - Adición de estructuras complementarias para acelerar el acceso a dichos registros.

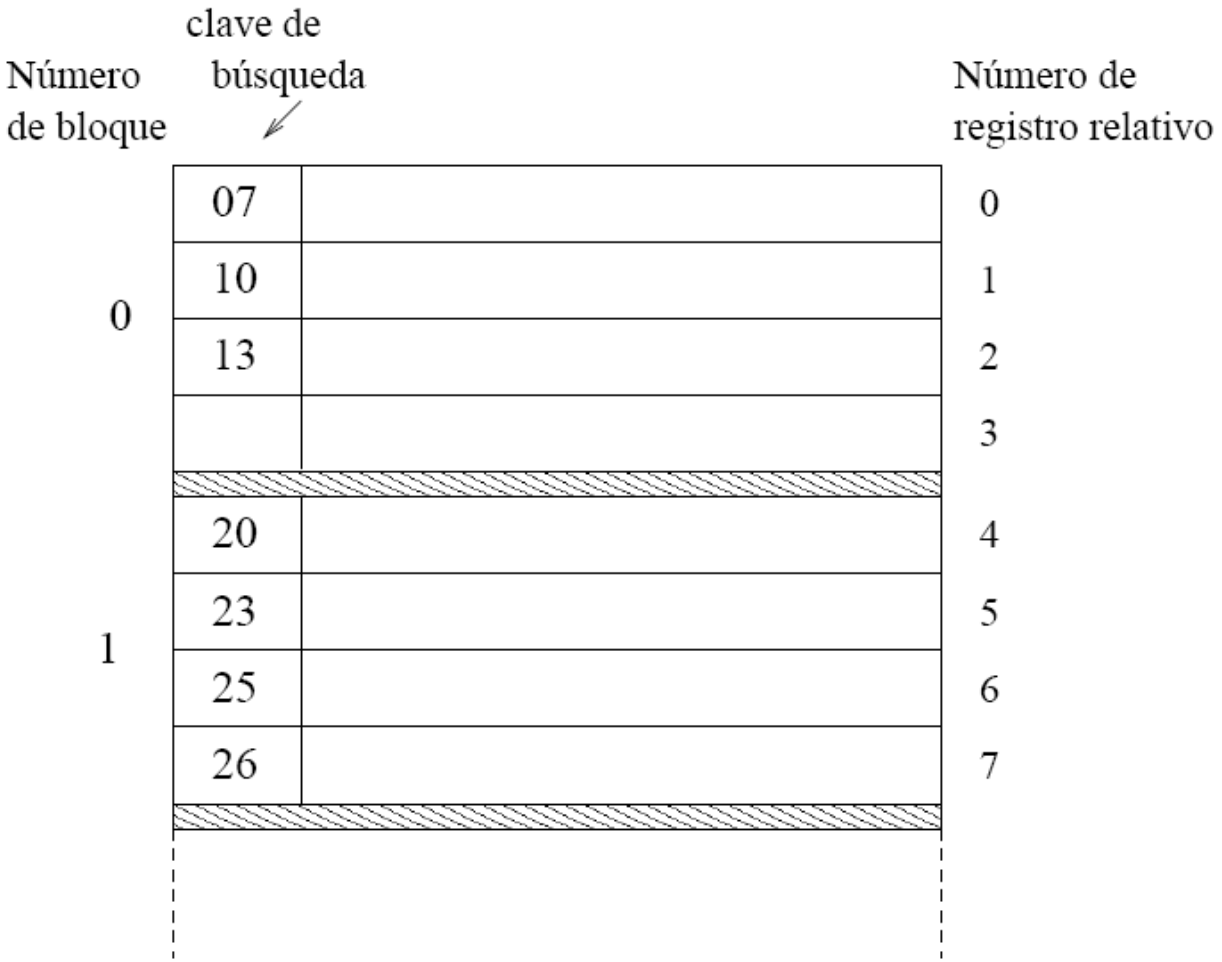
Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Organización Secuencial

- Fichero de acceso secuencial:
  - Aquél donde los registros están almacenados consecutivamente.
  - Para acceder a un registro determinado debemos pasar obligatoriamente por los registros que le preceden.
  - Los registros suelen estar ordenados por una clave (clave física).

Organización Secuencial



Organización Secuencial

- Ejemplo:
  - Mostrar la relación completa de departamentos.
  - La consulta se resolvería rápidamente si los departamentos están almacenados conjuntamente en bloques contiguos de un fichero.
  - Sin embargo:
    - ¿Qué pasa si queremos plantear consultas por valor de clave o por rango de valores?

Organización Secuencial

- El primer caso implica:
  - Recorrer uno tras otro cada uno de los registros.
  - En el caso peor (no encontrarse dicho departamento o ser el último de la lista) supone recorrer de forma completa el fichero.
  - Búsqueda es O(N)

- El segundo caso tiene un tratamiento muy parecido:
  - Se realiza la búsqueda por valor de clave de la cota inferior del intervalo.
  - Se continúa hasta alcanzar la cota superior. Si están ordenados por el valor de la clave.

Organización Secuencial

- Inserción de un nuevo registro:
  - Buscar el bloque que le corresponde.
    - Si hay sitio, se inserta el nuevo registro.
    - En caso contrario, o bien se opta por crear un nuevo bloque o bien se crea un bloque de desbordamiento.
  - Es recomendable dejar espacio vacío en los bloques para evitar los problemas de reorganización.
- Borrado de un registro:
  - Buscar el registro.
  - Puede implicar una reorganización local de los registros de un bloque.

Organización Secuencial

- En resumen, las dos operaciones suponen:
  - Escritura del bloque del registro que se inserta o borra.
  - Creación o liberación de bloques de datos en el fichero secuencial.
  - Creación o liberación de bloques de desbordamiento.
  - Reorganización de registros entre bloques contiguos, lo que implica la escritura de los bloques implicados en el desplazamiento.

Organización Secuencial

- Como puede verse, esta forma de organizar los registros no está exenta de grandes inconvenientes.
- Pueden subsanarse, al menos en parte, mediante el uso de estructuras adicionales que nos permitan:
  - Acelerar la localización de los datos.
  - Disminuir el número de bloques de disco transferidos.
- Entre las técnicas más populares se encuentran:
  - Índices
  - Acceso directo.

Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Indexación

- Tiene por objeto disminuir el tiempo de acceso a los datos por una clave de búsqueda.
- Similar a la idea de un índice en un libro.
- Existen muchas formas de llevar a cabo esta idea.

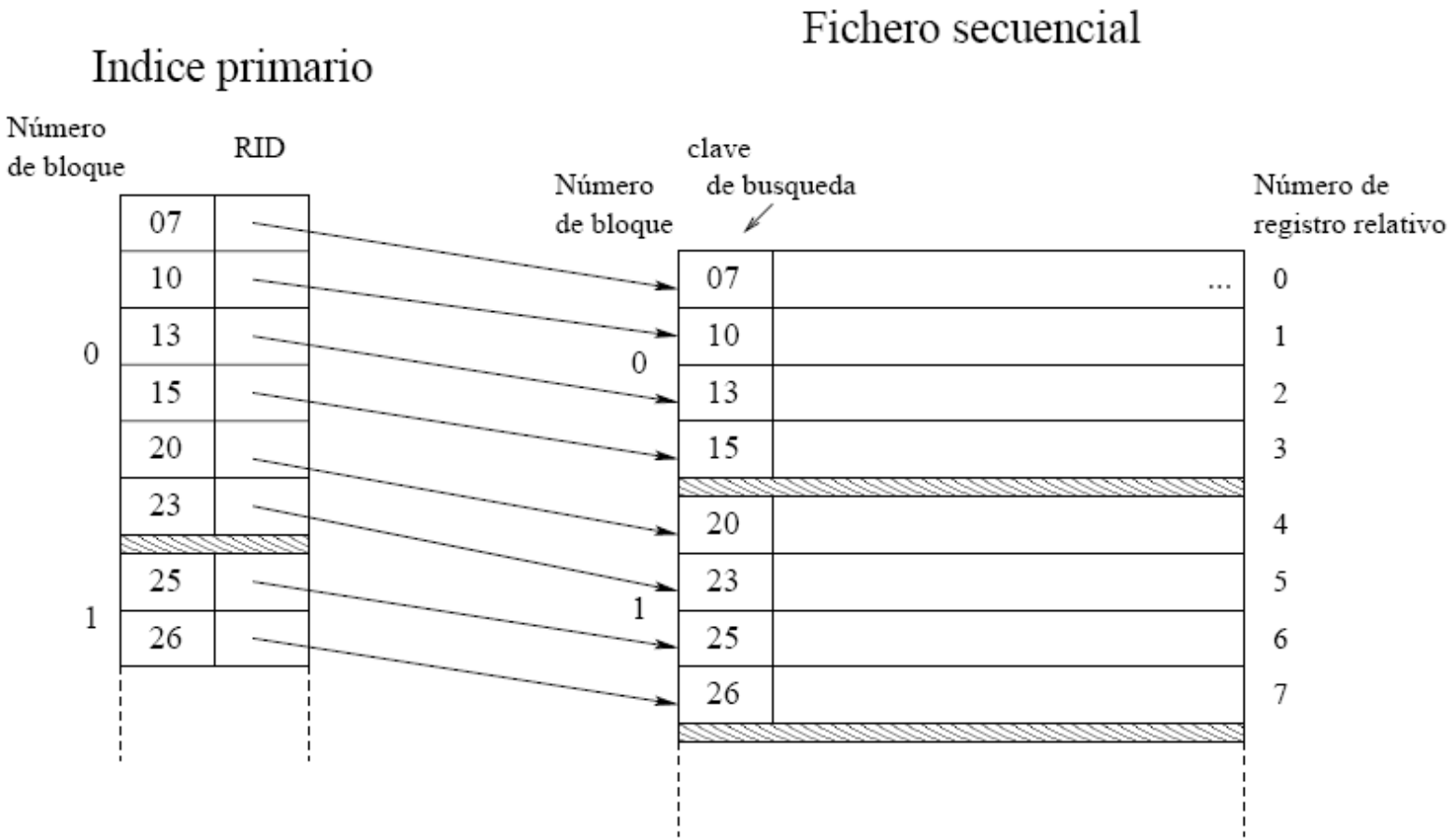
Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Ficheros indexados

- Partimos de un fichero secuencial sobre el que disponemos una estructura adicional: fichero índice
  - Sus registros poseen:
    - Campo clave (la clave de búsqueda)
    - Campo de referencia que contiene RIDs de registros.
  - Son más pequeños que los del fichero de datos, aunque el número de ellos es el mismo en ambos ficheros.

Ficheros indexados

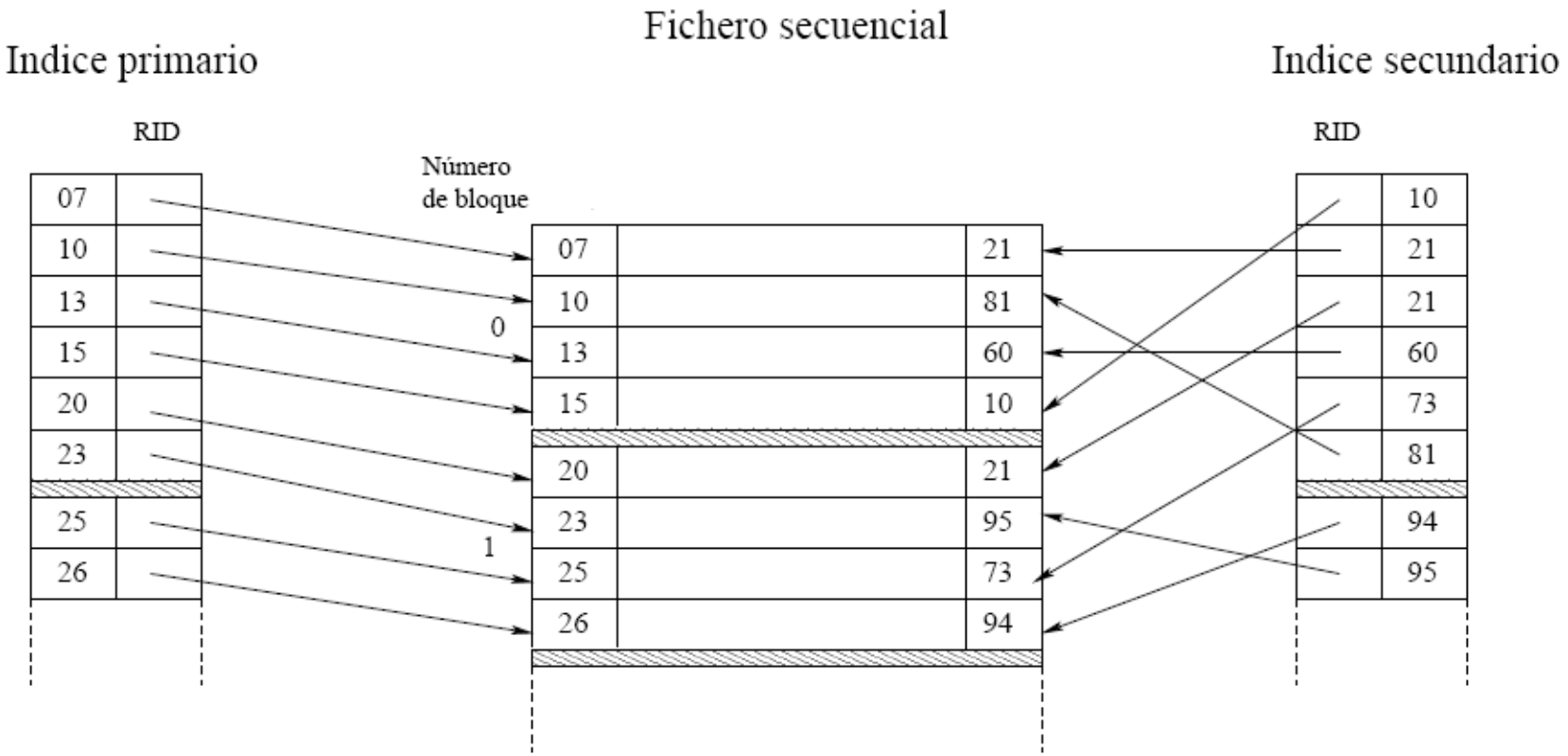


Ficheros indexados

- Índice primario:
  - La clave de búsqueda es el mismo campo clave (clave física) por el que está ordenado el fichero secuencial de datos.

- Índices Secundarios:
  - Construidos sobre otros campos que no sean la clave física del fichero de datos.

Ficheros indexados



Ficheros indexados

- Proceso de consulta
  - Consulta por un valor de la clave
    - Sobre el índice localizamos la clave (recorrido secuencial).
    - Obtenemos el RID del registro requerido.
    - Vamos a disco para recuperar el bloque de datos donde se encuentra el registro señalado por el RID.
    - La búsqueda en el índice es más rápida.
  - Consulta por rango de valores:
    - Búsqueda en el índice por valor de clave de la cota inferior
    - Recorrido de las entradas del índice que están en el intervalo, recuperando los registros correspondientes gracias a su RID.

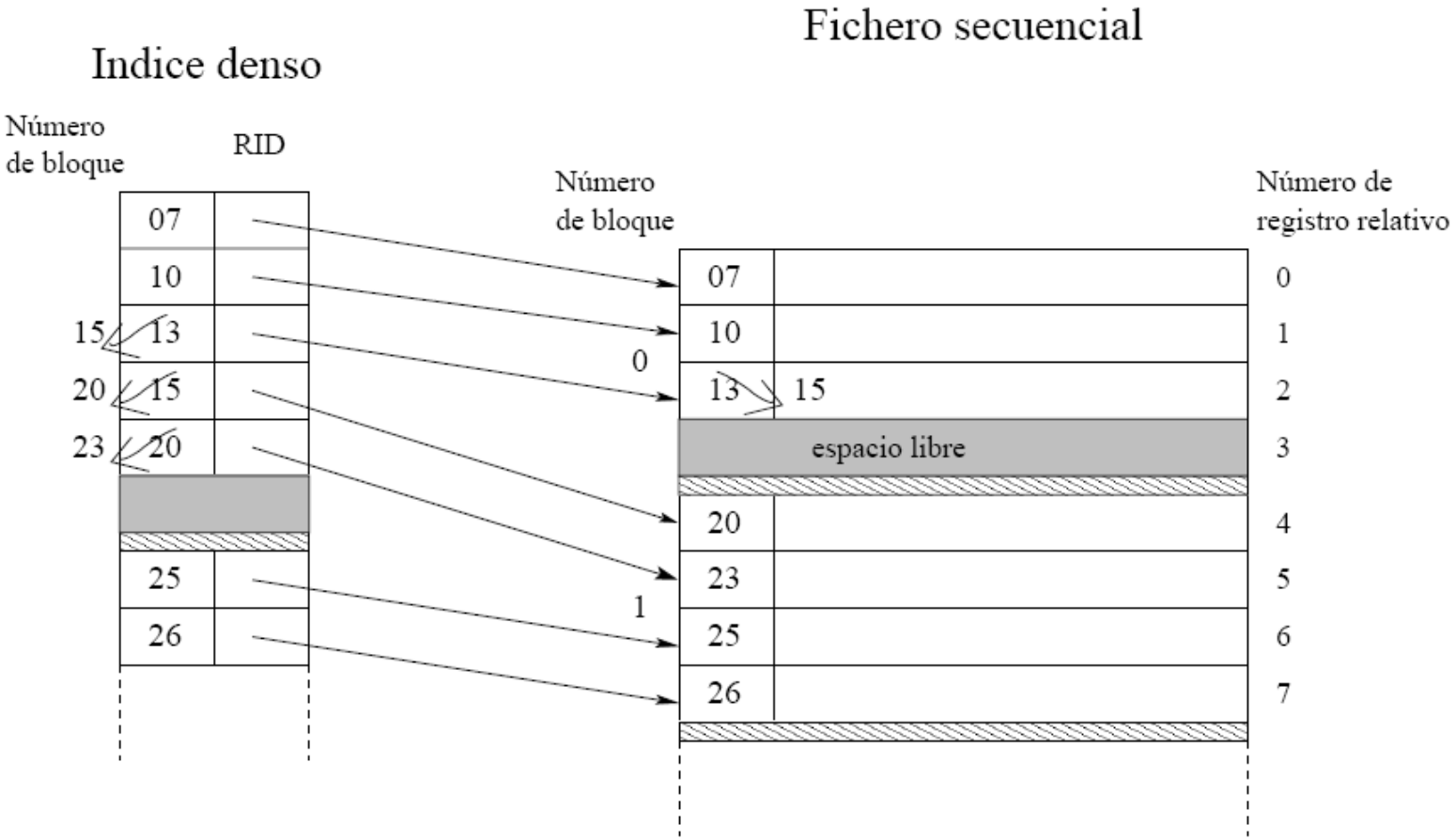
Ficheros indexados

- Inserción de un nuevo registro:
  - Las mismas operaciones que en el fichero secuencial.
  - Hay que actualizar también el índice (inserción en un fichero secuencial).

- Borrado de un registro:
  - Borrado de un registro en el fichero de datos.
  - Borrado de una entrada en el índice.

Ficheros indexados

- Ejemplo: Borrado del registro de clave 13



Ficheros indexados

- Se puede montar un índice sobre más de un campo de un registro.
  - Clave: concatenación de los campos indicados.

- Hay que tener cuidado:
  - Un índice sobre nombre-alumno y DNI.
  - Útil para consultas que involucran:
    - Nombre
    - Nombre y DNI
  - No es útil para consultas sobre el DNI.

Ficheros indexados

- Conclusiones:
  - Los índices:
    - Aceleran el acceso a los datos.
    - Ralentizan las otras operaciones.
      - Hay que mantener el índice.
  - Por tanto:
    - Hay que considerar la conveniencia de crear cada índice.
      - Frecuencia de las consultas.
      - Frecuencia de las operaciones de mantenimiento de los datos.

Contenidos

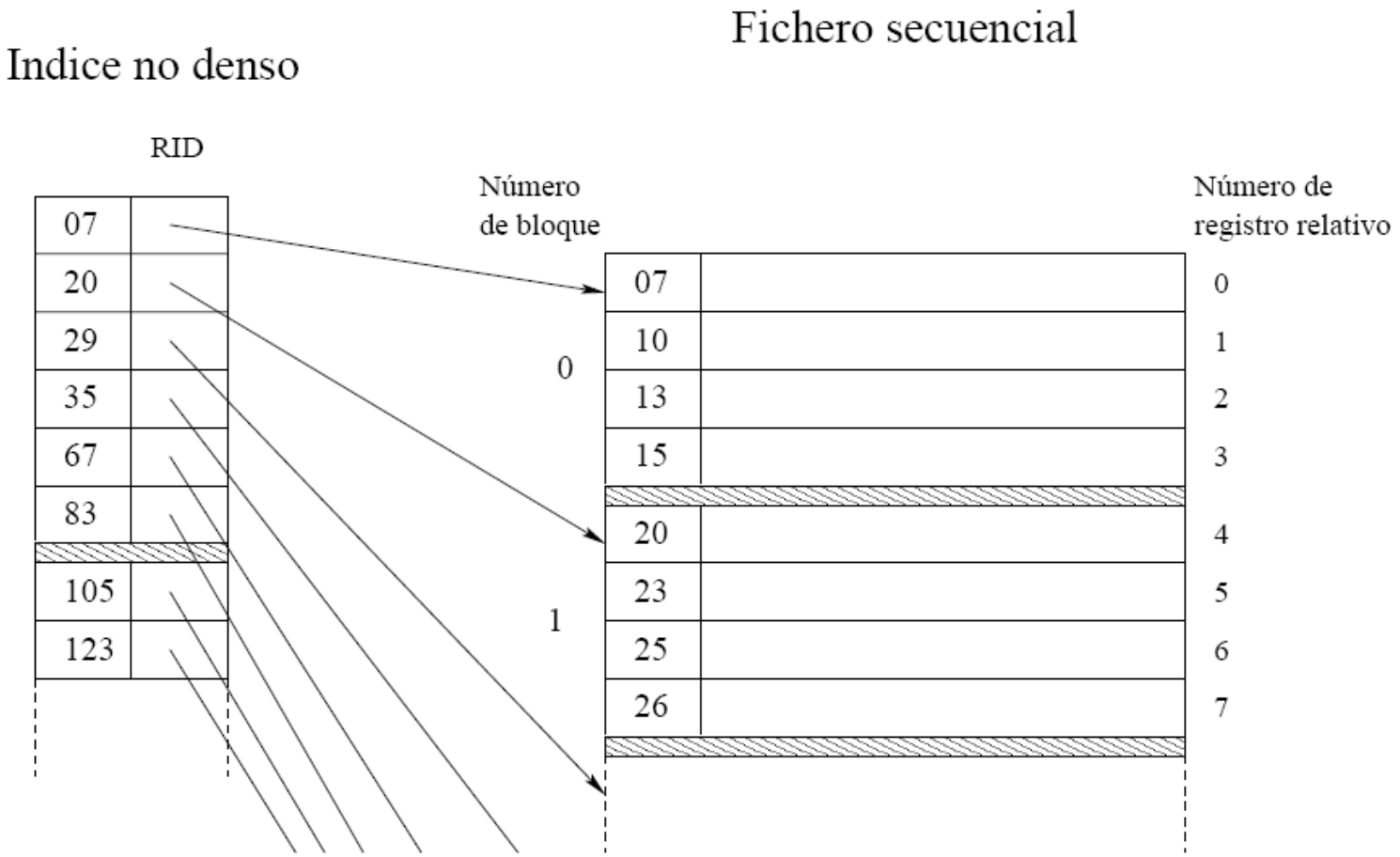
- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Índices no densos

- Ideal: Mantener el índice en memoria principal.
- Realidad: los índices siguen siendo muy grandes, porque contienen todos los registros del fichero que indexan.
  - Son densos.

- Para reducir el tamaño aparecen los índices no densos:
  - Registros compuestos por:
    - La clave de búsqueda
    - La dirección de comienzo del bloque donde puede encontrarse el registro deseado
  - El número de registros se reduce al número de bloques del fichero de datos
    - El acceso secuencial al índice no denso se acelera.

Índices no densos



Índices no densos

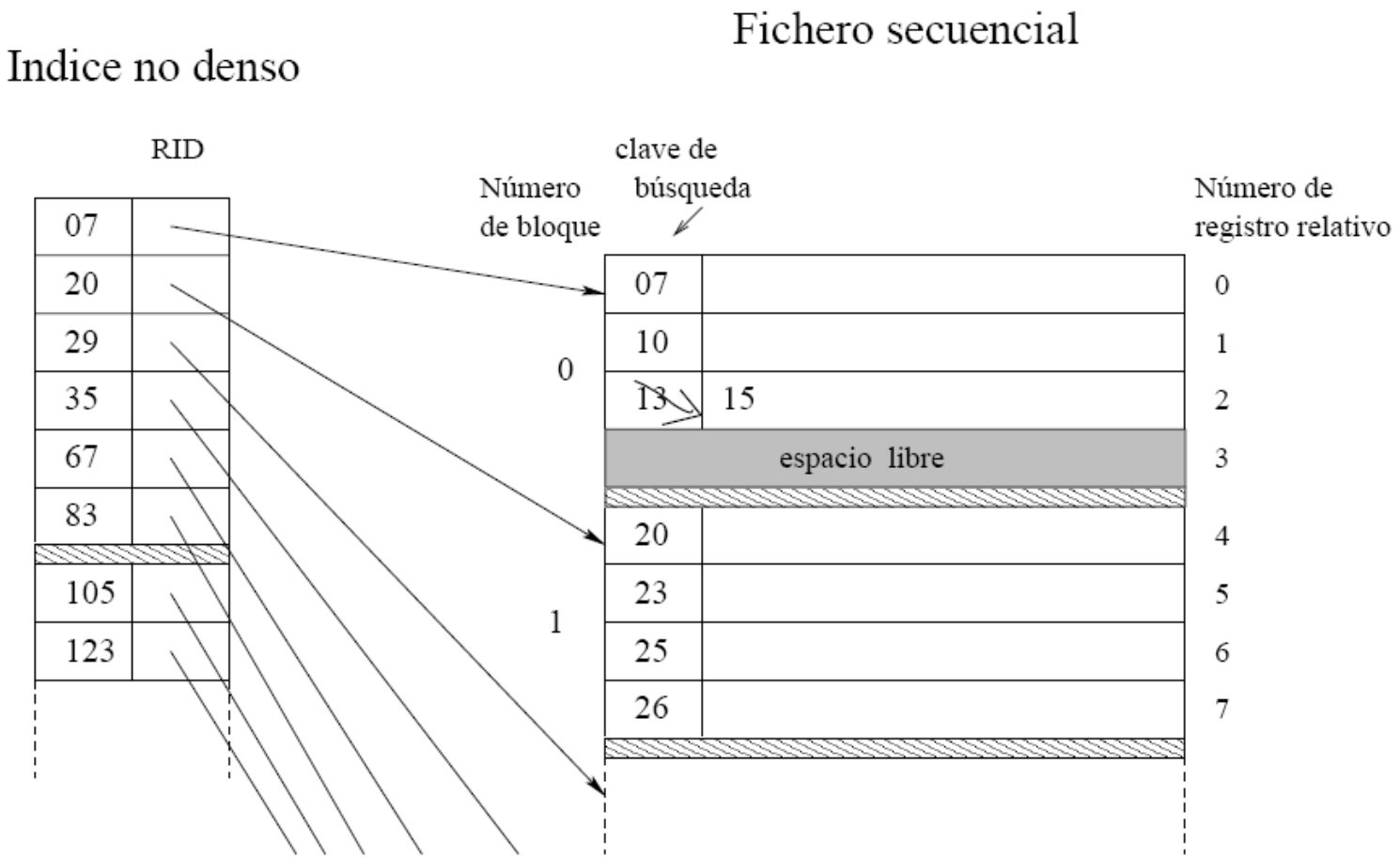
- Diferencias en el proceso de búsqueda:
  - Una vez encontrado el bloque donde podría encontrarse el registro:
    - Hay que cargarlo en memoria.
    - Hay que hacer una búsqueda secuencial.
      - No tiene costes en términos de acceso a disco.
  - No se tiene garantía alguna de encontrar el registro deseado hasta consultar el bloque de datos leído.

Índices no densos

- Los índices no densos sólo se pueden definir sobre la clave física.

- El mantenimiento de un índice no denso es menos costoso:
  - Inserción y borrado menos frecuentes
  - Sólo ocurren cuando la operación afecta al valor representativo del bloque.

Índices no densos



Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Índices jerárquicos

- Volvemos al objetivo de disminuir el tiempo necesario para recorrer el índice en busca de un registro:
  - Idea: crear índices sobre índices.
  - Varios niveles en el acceso a los datos.

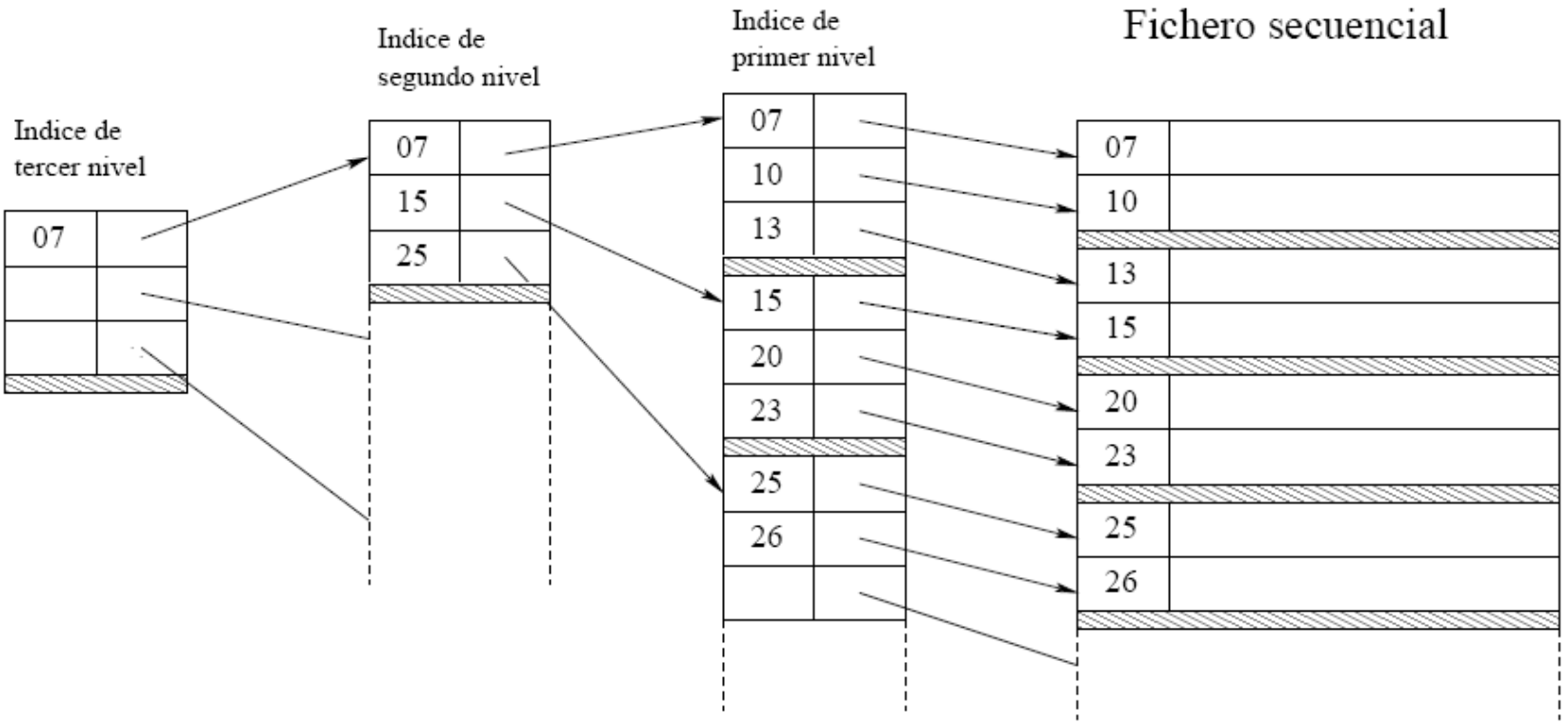
- Un índice multinivel está formado:
  - Un índice de primer nivel sobre el fichero de datos.
    - Puede ser denso o no dependiendo de la clave.
  - Otros índices, no densos, construidos sucesivamente unos sobre otros.

- El tamaño de los bloques se establece con la idea de optimizar cada una de las operaciones de acceso al disco físico.

- Se reduce el número de accesos a disco para localizar un registro:
  - En el peor caso: tantos como niveles

- Se complica el mantenimiento del índice.

Indices jerárquicos



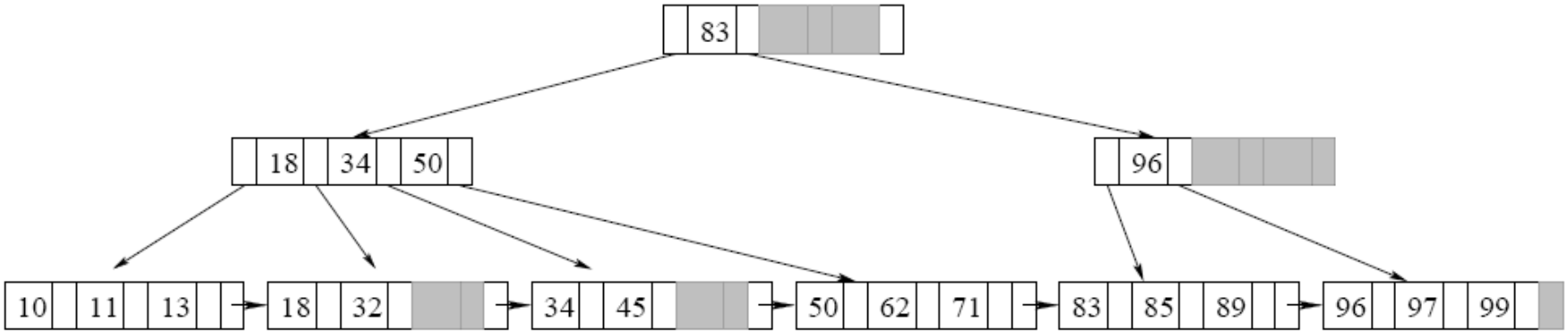
Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Árboles B+

- Propuestos en 1972 por Bayer y McCreight son una generalización de los árboles binarios balanceados en la que los nodos pueden tener más de dos hijos.
- Todos los valores de la clave se encuentran almacenados en los nodos hoja.
- Un Árbol B+ de orden M (el máximo número de hijos que puede tener cada nodo) es un árbol con la siguiente estructura:
  - Nodo de nivel superior: raíz del árbol
  - Nodos del nivel inferior: hojas.
  - Cada nodo distinto de las hojas tiene como máximo M hijos.
  - Todos los nodos hoja aparecen al mismo nivel.
  - Las claves contenidas en cada nodo nos guiarán hasta el siguiente nodo del nivel inmediatamente inferior.
  - Un nodo no hoja con n hijos contiene:
    - n-1 valores de clave almacenados.
    - n punteros P_i que apuntan a un nodo hijo

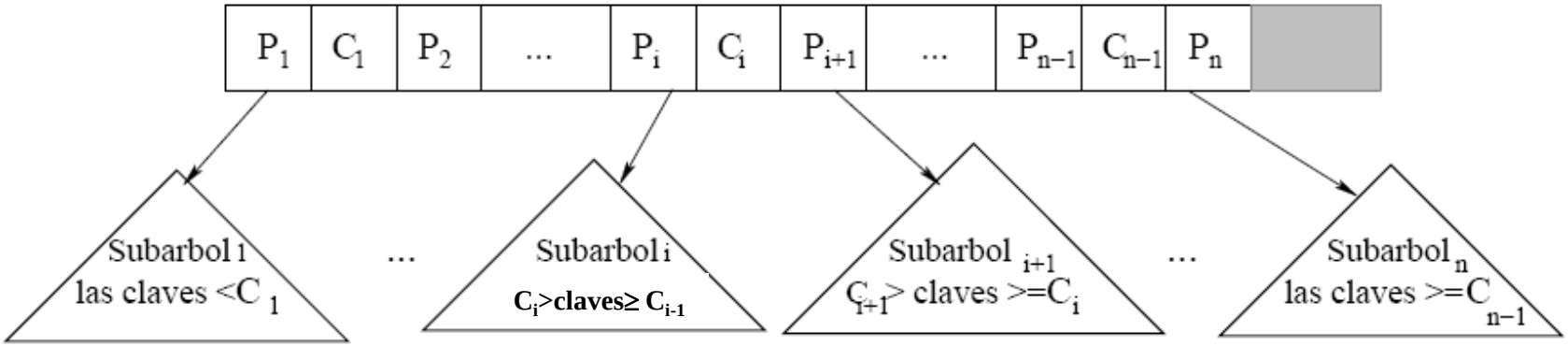
Árboles B+



Herramienta interactiva:
https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html

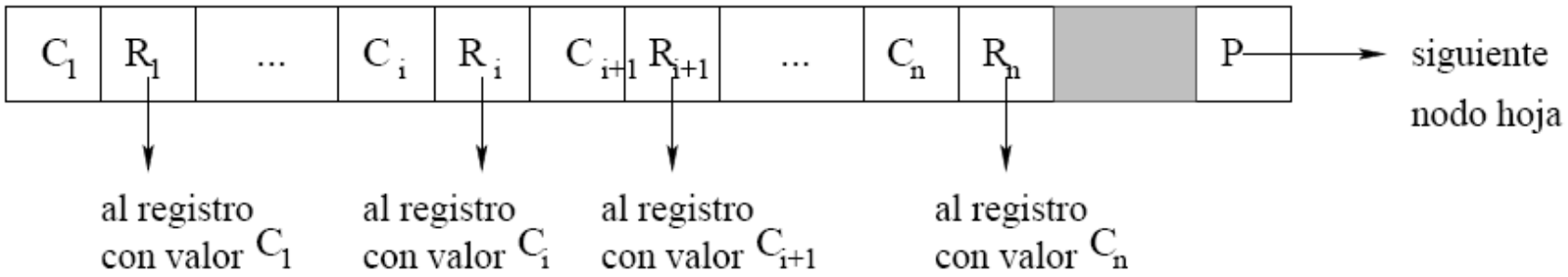
Árboles B+

- Restricciones dentro de los nodos:
  - Los valores de clave Ci están ordenados dentro del nodo.
  - Los valores x del subárbol apuntado por P_i cumplen:
    - $C_{i-1} \leq x < C_i$
    - Excepto para:
      - i = 1, donde $x < C_1$
      - i = n, donde $x \geq C_{n-1}$



Árboles B+

- Nodos hoja:
  - Tienen una estructura diferente
    - Parejas clave – RID
    - Punteros al siguiente nodo hoja
    - Algunas variantes también tienen punteros al nodo hoja anterior.
  - La lista concatenada de nodos hoja (conjunto secuencia), tiene gran utilidad a la hora de hacer consultas por intervalos.



Árboles B+

- Restricciones en nodos hoja:
  - Las claves aparecen ordenadas en cada nodo.
  - Todas las claves han de ser menores que las del siguiente nodo en el conjunto secuencia.
  - Todos los nodos hoja se encuentran en el mismo nivel.
    - Árbol equilibrado.
    - Todos los caminos desde la raíz a un nodo hoja tienen la misma longitud.

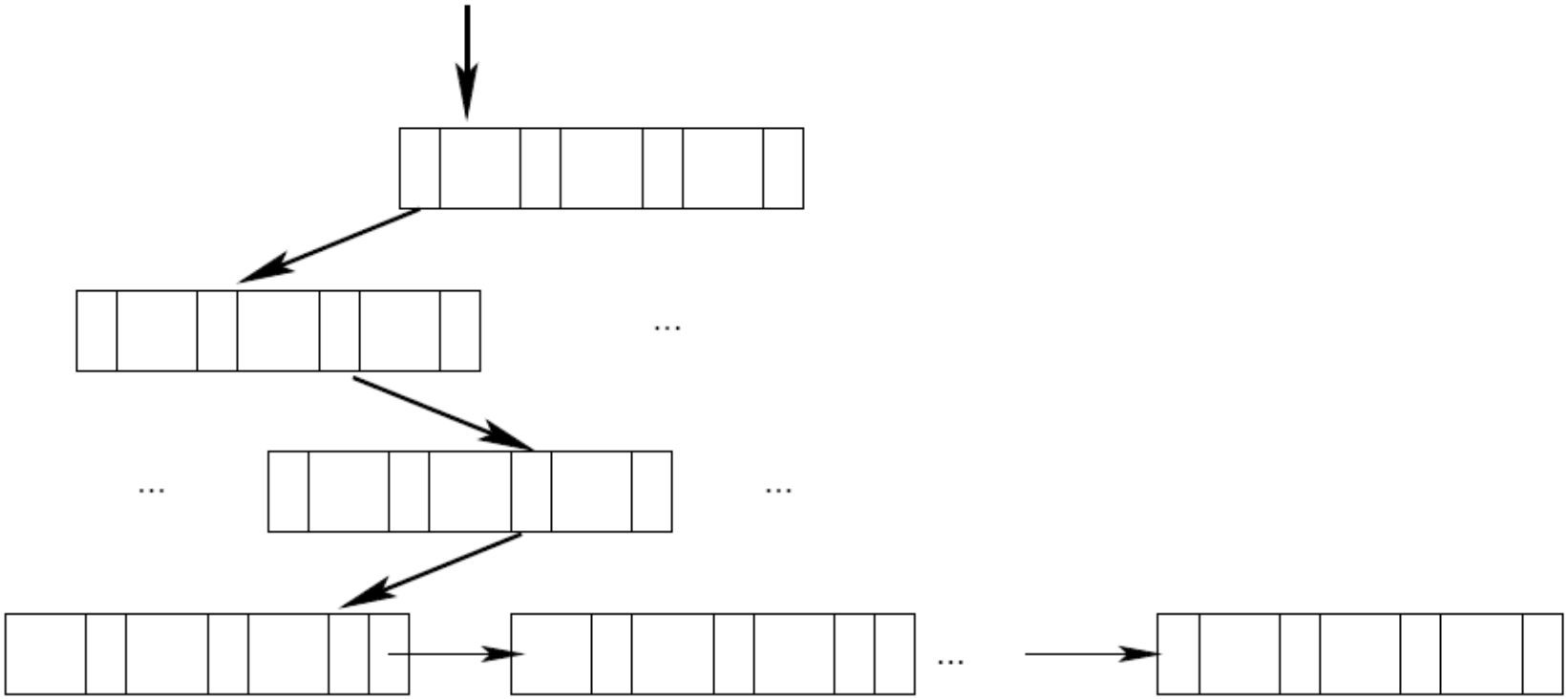
Árboles B+

Nodo	Restricción	Min	Max	Ejemplo M=5
Raíz (cuando no es nodo único)	Número de hijos	2	M	2-5
Interno	Número de hijos	$\lceil M/2 \rceil$	M	3-5
Hoja	Número de claves	$\lfloor M/2 \rfloor$	M-1	2-4

Árboles B+

- Proceso de consulta
  - Localización de un registro:
    - Navegamos desde la raíz, bajando niveles.
    - Buscamos el registro en el nodo hoja y, en su caso, recuperamos el registro del fichero de datos gracias al RID.
  - Consultas por rango:
    - Se localiza el nodo hoja que contiene el valor inferior.
    - Se recorren los nodos hoja hasta alcanzar el superior, recuperando los registros pertinentes del fichero de datos
- Inserción y borrado
  - Se utilizan algoritmos que garantizan que el árbol resultante sea equilibrado.

Árboles B+



Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Árboles B

- Los Árboles-B (B-Tree) son una variante de B+Tree en la que no se almacenan todos los valores de la clave en los nodos hoja, sino que algunos valores se van almacenando en los nodos intermedios conforme se crea el árbol.

Herramienta interactiva:

https://www.cs.usfca.edu/~galles/visualization/BTree.html

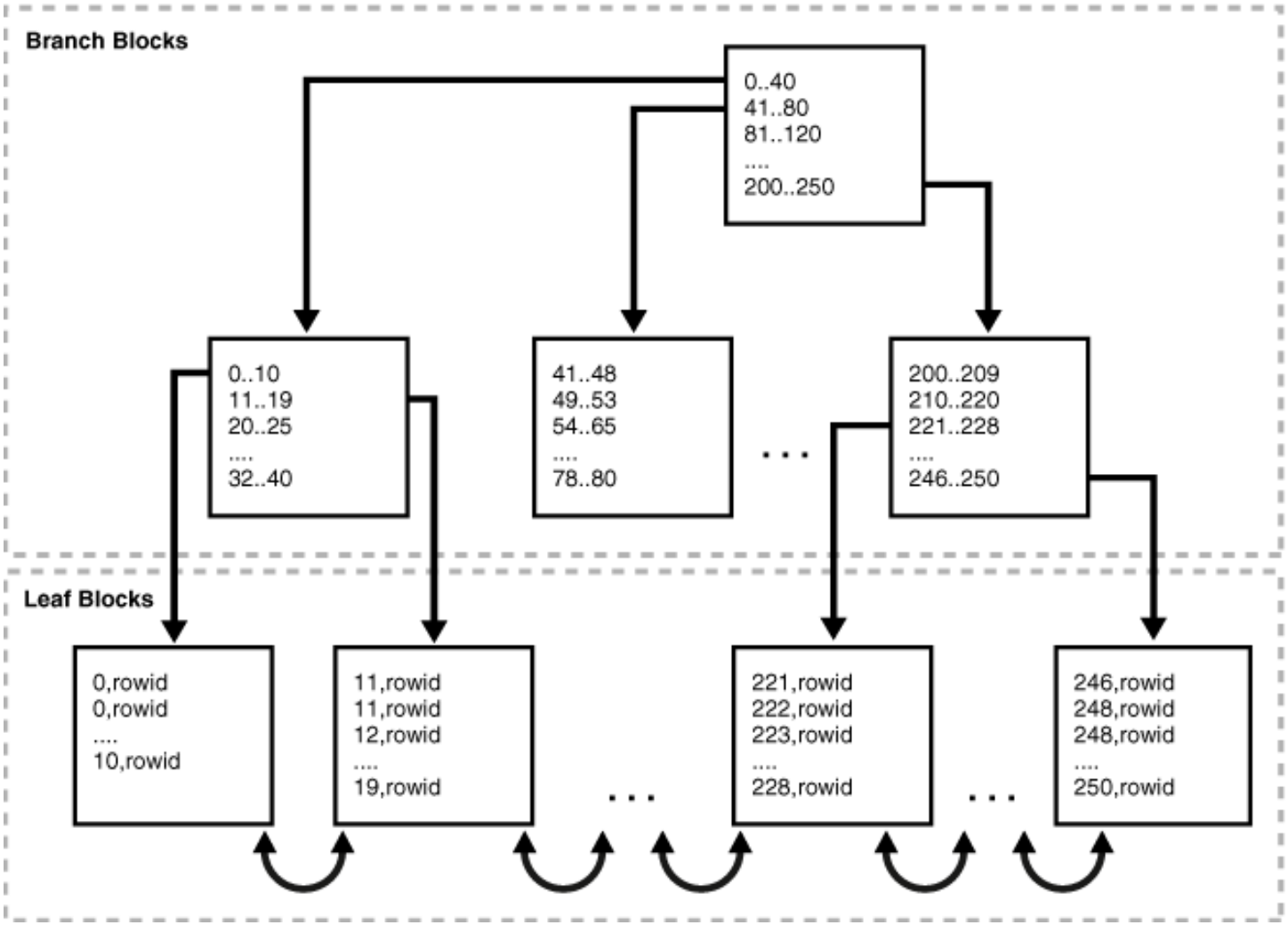
Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Árboles B+ en Bases de Datos

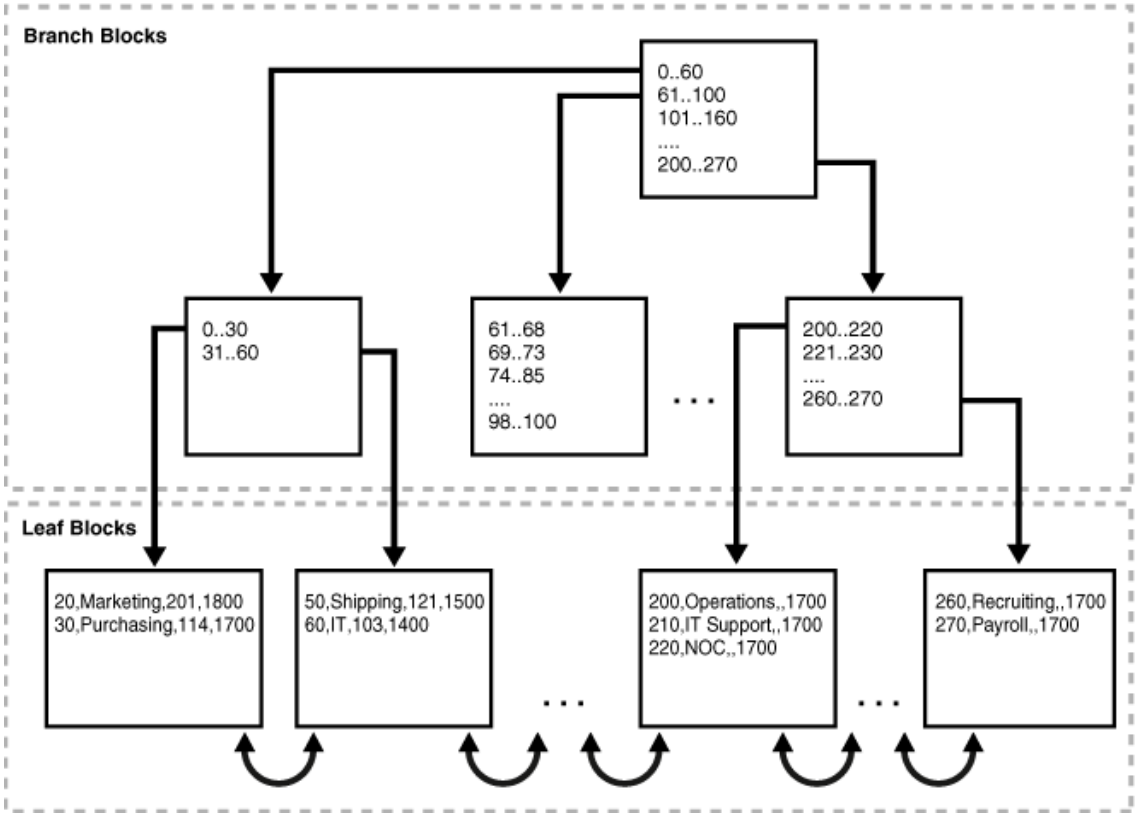
- Son variaciones del Árbol B+, de orden elevado, en la que se procura que cada nodo tenga una capacidad de almacenamiento similar al tamaño de un bloque de datos.
- Esto reduce los accesos a disco que suelen ser los que determinan el rendimiento de las búsquedas en BD.
- En los nodos intermedios solo están los rangos de los valores de la clave y los punteros a los nodos hijo correspondientes.
- En los nodos hoja se encuentran todos los valores de la clave ordenados junto con los RIDs(rowid) que apuntan a las tuplas que contienen ese valor de la clave.
- Los nodos hoja, que forman el conjunto secuencia, se encuentran enlazados para poder recuperar por búsquedas secuenciales, a veces se encuentran doblemente enlazados, para facilitar búsquedas ascendentes y descendentes por el valor de la clave.

Árboles B+ en Bases de Datos

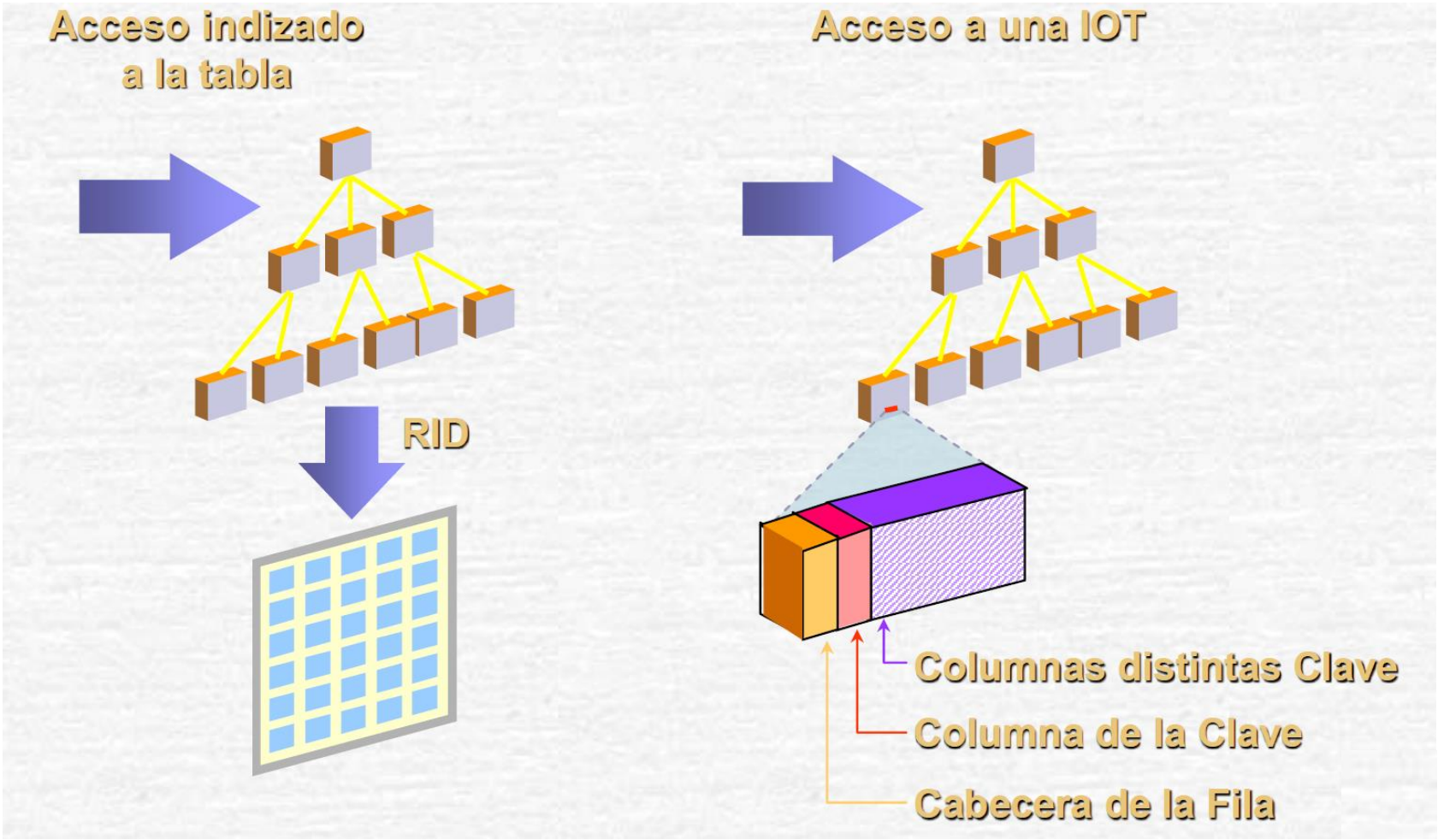


Árboles B+ en Bases de Datos

- Tablas Organizadas por Índice (IOT). Las hojas contienen las tuplas en lugar del RID. Una IOT sólo puede estar organizada de esta forma mediante una clave, aunque se pueden definir índices adicionales basados en otras claves.



Árboles B+ en Bases de Datos



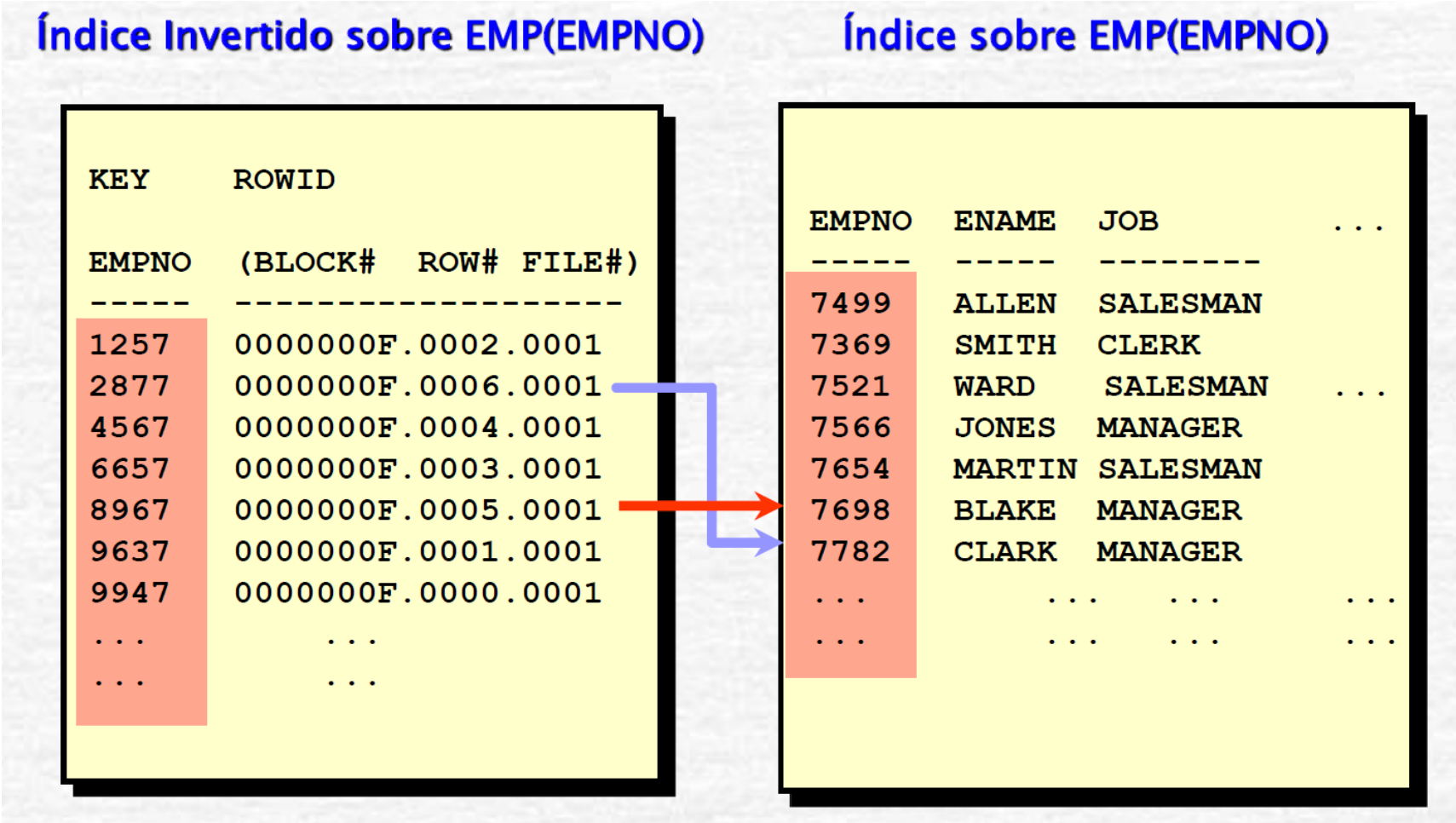
Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Índices por Clave Invertida

- Índice por Clave Invertida (reverse index). Invierte los datos del valor de la clave.
  - Para el empleado 7698 almacena 8967.
  - Este índice es adecuado para búsquedas basadas en predicados =.
  - Con este índice se reducen los embotellamientos (retención de bloques de BD) en el índice cuando se están introduciendo los datos de forma ascendente para los valores de la clave, puesto que todos irían a la misma entrada de índice.

Indices Clave Invertida



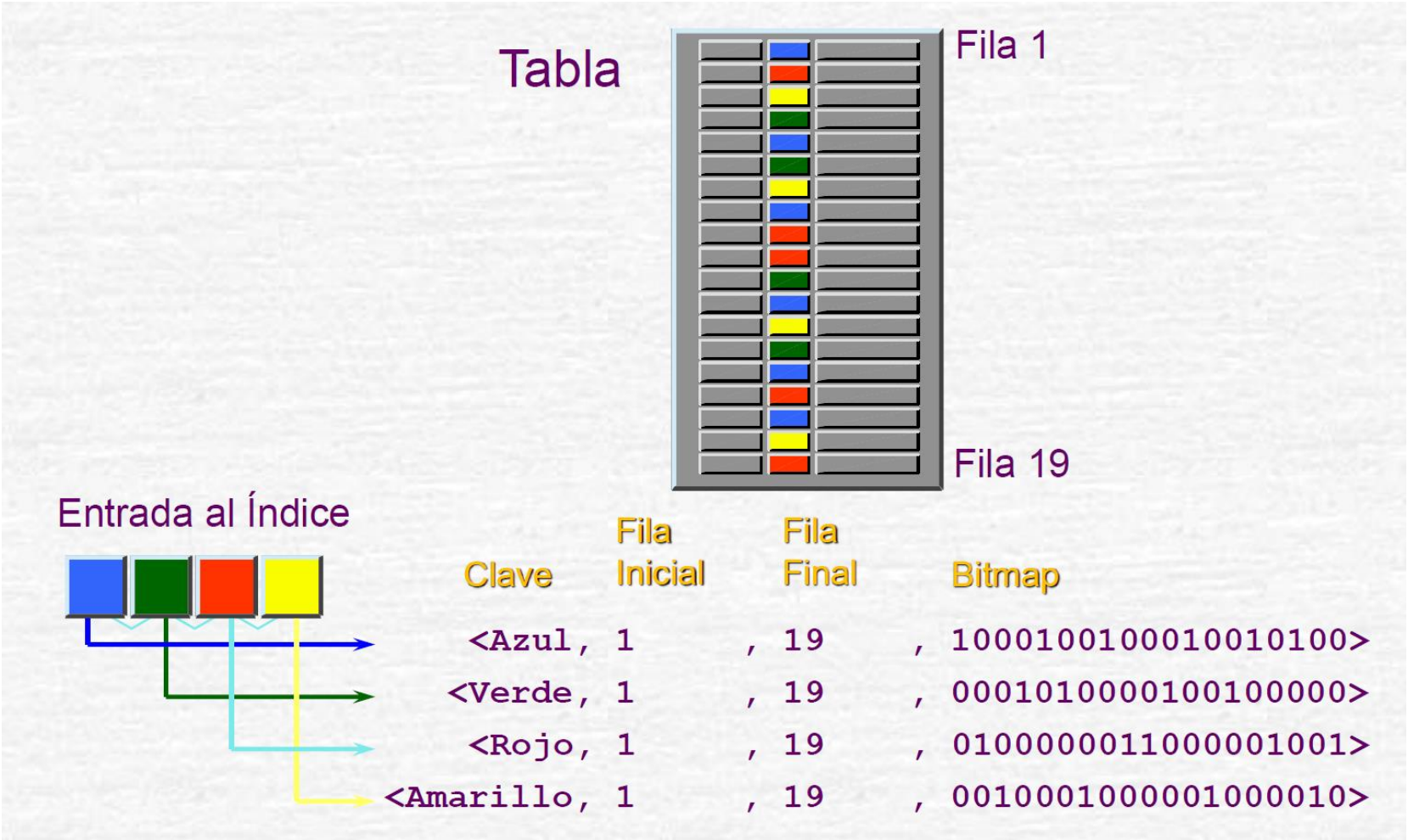
Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Indices BITMAP

- Indices BITMAP
  - Para cada valor que toma la clave almacena una secuencia de bits (tantos como tuplas contenga la tabla):
    - El bit 1 indica que ese valor está presente en la tupla
    - El 0 que no lo está.

Indices BITMAP



Indices BITMAP

B-tree	Bitmap
Adecuado para columnas que tomen muchos valores	Adecuado para columnas que tomen pocos valores
Actualizaciones sobre las claves no muy costosa	Actualizaciones sobre las claves muy costosa
Ineficiente para consultas usando predicados OR	Eficiente para consultas usando predicados OR

Contenidos

- Organización y métodos de acceso
- Organización secuencial
- Indexación
- Ficheros indexados
- Índices no densos
- Índices jerárquicos
- Árboles B+
- Árboles B
- Árboles B+ en BD
- Índices clave invertida
- Índices BITMAP

Imágenes de portada y cabecera tomadas de https://pixabay.com/