

Índices

José R. Paramá



- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

- Un índice sobre un fichero es una estructura auxiliar diseñada para acelerar las operaciones que no son soportadas eficientemente por las organizaciones básicas de ficheros.
- Existe ya un fichero con cualquier organización de las vistas anteriormente.
- Las estructuras de índices proporcionan caminos alternativos para acceder a los registros sin afectar a la posición física de los registros en el fichero.
- Acceso eficiente a registros basándose en campos de indexación.

- Se puede construir sobre el mismo fichero, múltiples índices sobre varios campos.
- Otra de las ventajas con respecto a la técnicas de hash son las búsquedas de rango (por ejemplo, los empleados que ganen más de 20.000?).
- Estas consultas son ineficientes en el caso de los ficheros hash, ya que sólo pueden resolver eficazmente consultas por igualdad en la clave del fichero.
- Los índices ayudan en todo tipo de consultas, aunque en las consultas por igualdad en la clave del fichero, los ficheros hash son más eficientes

- **Clave de búsqueda o campo de indexación** Atributo, del conjunto de atributos, empleado para buscar registros en un archivo.
- Un archivo de índices consta de registros (denominados entradas de índice) de la forma

clave de búsqueda	puntero
-------------------	---------

- El índice almacena todos los valores del campo de indexación junto con una lista de punteros a todos los bloques del disco que contienen registros con cada valor del campo de indexación.

- **Índice primario**: Si tenemos un fichero ordenado, cuando el índice indexa la clave de ordenación del fichero de datos.
 - También denominado **índice con agrupación** si hay varios registros con el mismo valor de clave.
- **Índice sin agrupación**: que se especifica sobre cualquier campo(s) que no es el de ordenación (clave). También llamado índice secundario.

- **Índice denso**: Tiene una entrada de índice por cada valor existente en el campo(s) indexados.
- **Índice escaso (o disperso)**: NO tiene una entrada de índice por cada valor existente en el campo(s) indexados.

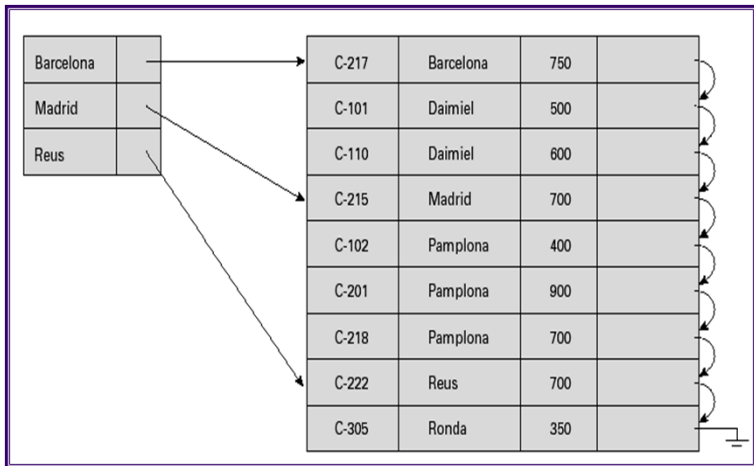
Conceptos básicos

Índice denso con agrupación



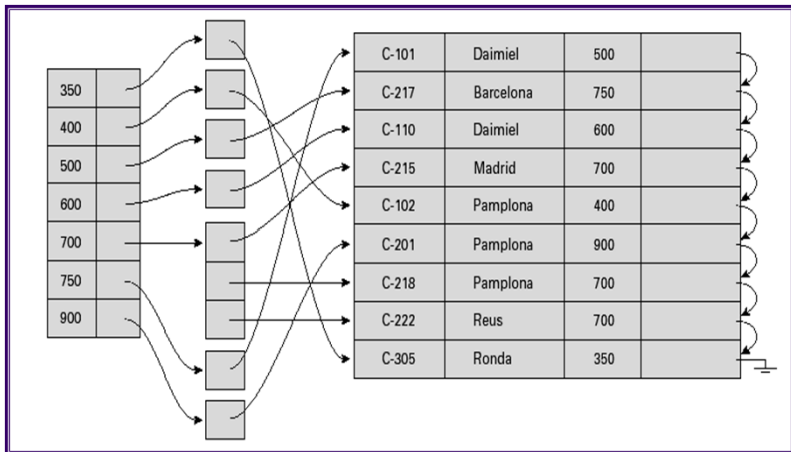
Conceptos básicos

Índice escaso con agrupación



Conceptos básicos

Índice denso sin agrupación



- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

- Tipos de acceso soportados eficientemente. Por ejemplo,
 - registros con un valor concreto en el atributo
 - o registros con un valor de atributo que se encuentra en un determinado rango de valores.
- Tiempo de inserción
- Tiempo de borrado
- Costes de espacio

- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

Índices basados en árboles

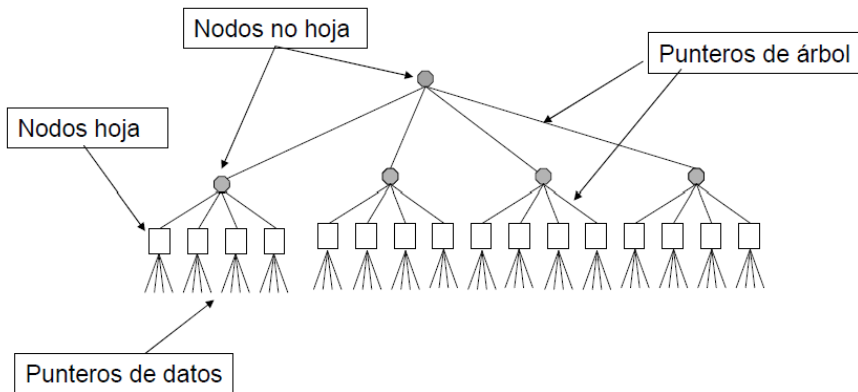
- Los índices basados en árboles proporcionan una mejora sobre la búsqueda binaria.
- La búsqueda binaria en un caso particular de búsqueda en un árbol que sólo tiene dos nodos hijo por nodo.
- Cada vez que bajamos por una rama, descartamos la mitad de los registros que quedan por buscar.
- Si aumentamos el número de hijos por nodo (**fan-out**) salen de un nodo, descartamos más nodos cada vez que seleccionamos una de ellas.

- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

Árboles heterogéneos

- Los **árboles heterogéneos** son aquellos donde cada nodo del árbol contiene sólo un tipo de punteros
- Los punteros de los nodos hoja son de distinto tipo que los de los nodos no hoja.
- Los punteros de los **nodos hoja** apuntan a los **registros** del fichero de datos.
- Los punteros de los **nodos no hoja** apuntan a otros **nodos** (en niveles inferiores) del árbol.

Árboles heterogéneos

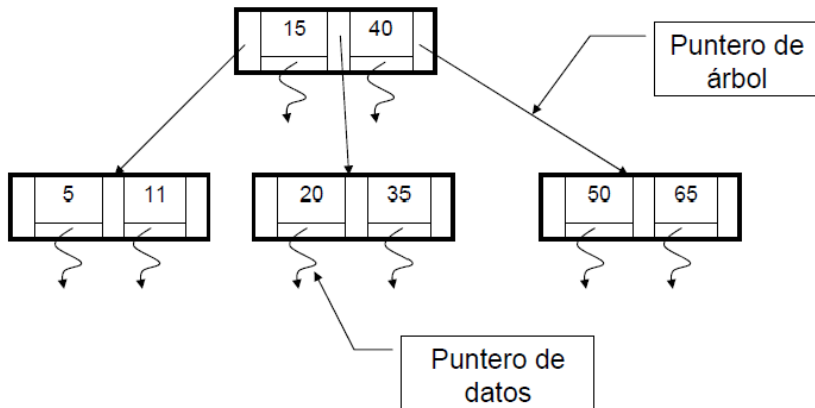


- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

Árboles homogéneos

- Los **árboles homogéneos** son aquellos en los que cada nodo contiene dos tipos de punteros, punteros a **registros (punteros de datos)** y **punteros a otros nodos (punteros de árbol)**.
- Todos los nodos son idénticos respecto a su estructura, los nodos hoja tienen punteros de árbol vacíos y punteros de datos activos.
- Los nodos no hoja, tienen ambos tipos de punteros activos.

Árboles homogéneos



- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B^+
- Claves duplicadas en Árboles B

Árboles heterogéneos y homogéneos

- La longitud media de la búsqueda será mayor en los árboles heterogéneos que en los homogéneos.
- Esto es porque en los heterogéneos la búsqueda siempre tiene que llegar a los nodos hoja, mientras que en los homogéneos puede acabar en cualquier nivel.
- El precio que se debe pagar, es el espacio necesario para el doble juego de punteros en cada nodo y unos algoritmos un poco más complejos.

- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

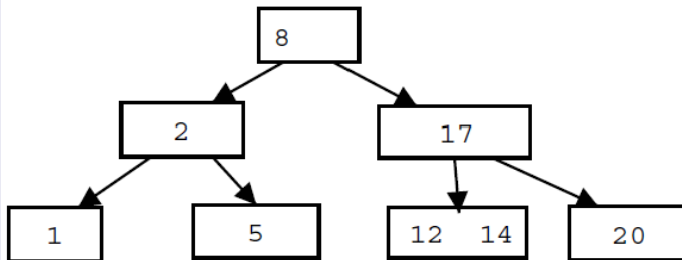
Árboles B

- Cada nodo puede alojar como mucho $2d$ valores del campo de indexación con sus punteros a datos y $2d + 1$ punteros de árbol.
- Ningún nodo, excepto el nodo raíz, puede tener menos de d valores del campo de indexación.
- Todos los nodos hoja están en el mismo nivel.

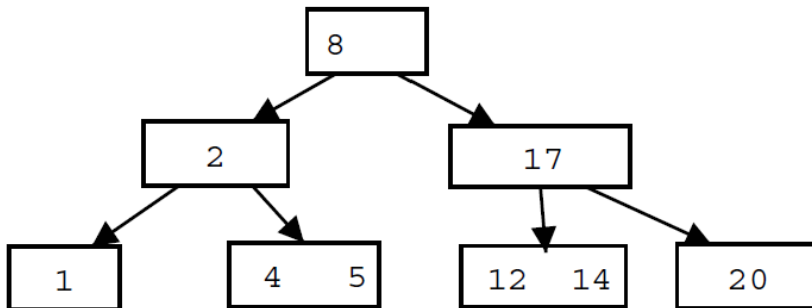
Añadir entrada al índice

- Las nuevas entradas siempre se **añaden en los nodos hoja**.
- Siempre existe la posibilidad de que el nodo que le corresponde a la entrada que queremos añadir ya esté lleno, en tal caso se produce un desbordamiento, en tal caso, el primer remedio es **redistribuir**.
- Se redistribuye las entradas entre el nodo objeto de la inserción, su padre y un **nodo hermano adyacente**.
- Si esto no es posible, el nodo se **divide** en dos nodos, con la entrada que sería el valor medio del nodo promocionando al nodo padre.

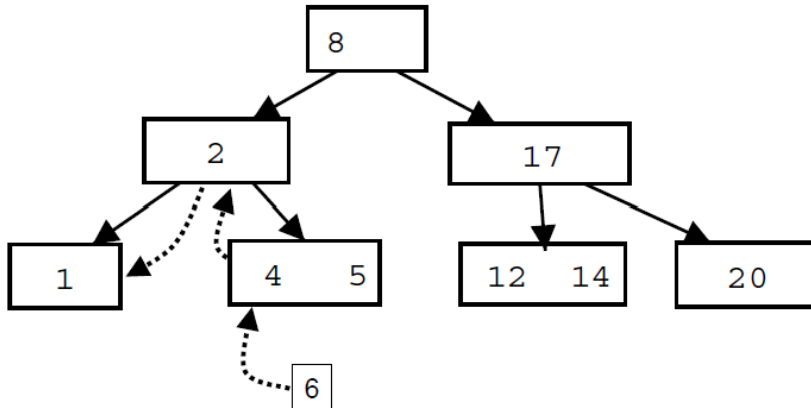
Añadir 4



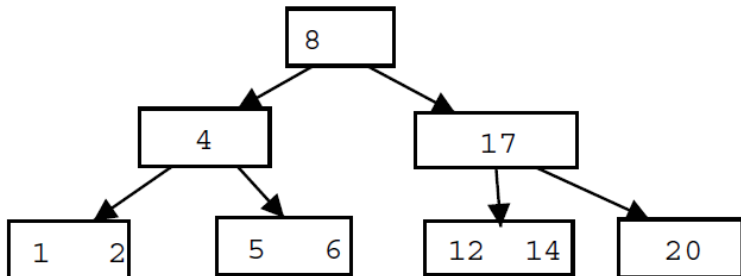
Añadir 4



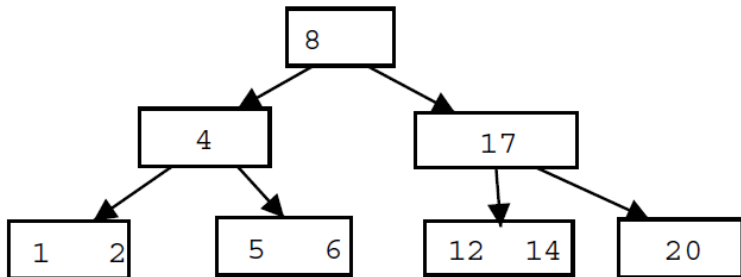
Añadir 6



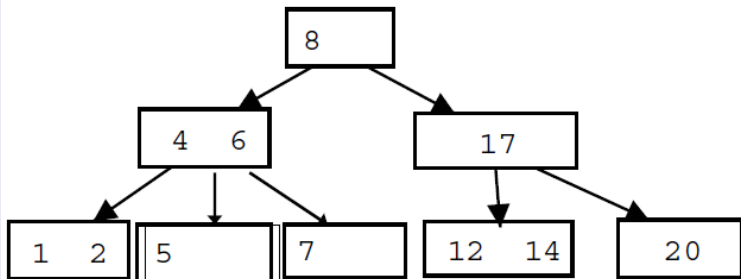
Añadir 6



Añadir 7



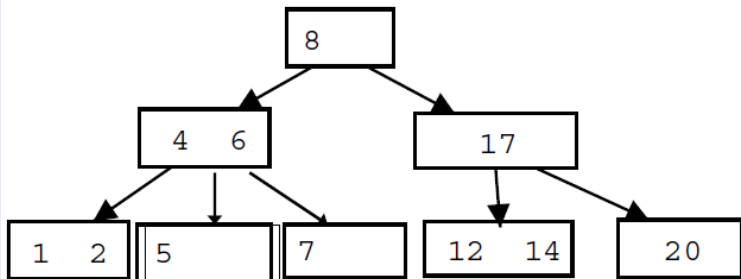
Añadir 7



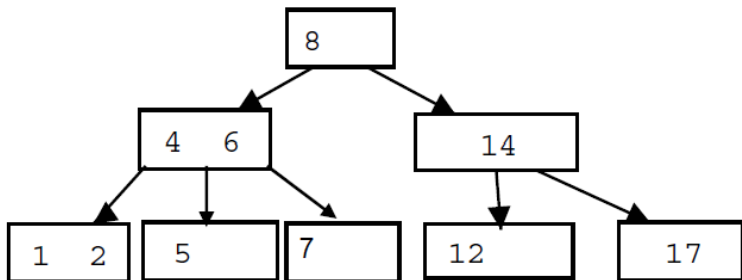
En nodos hoja

- Una vez se elimina la entrada del nodo correspondiente, se debe comprobar que el número de entradas en el nodo **no baje de d** .
- **Primer recurso** es intentar la **redistribución de claves con un nodo hermano adyacente con más de d claves**.

Borrar 20



Borrar 20

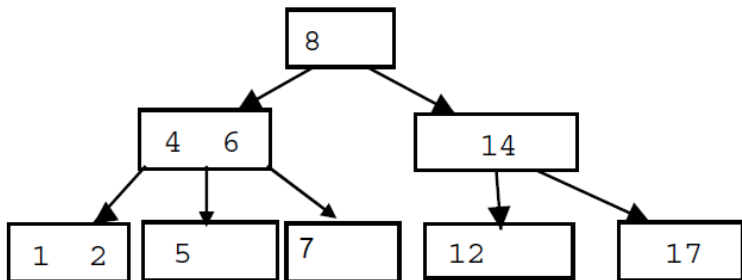


En nodos hoja

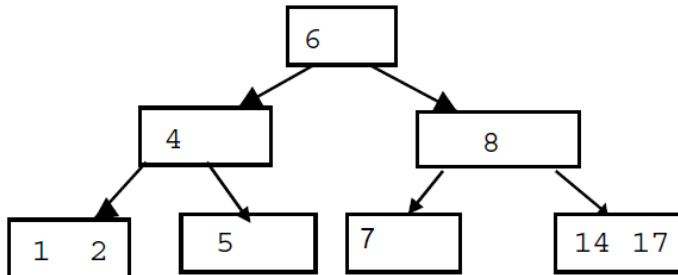
Si no es posible la redistribución

- Se **funden** el **nodo con menos de d claves**, y su **hermano adyacente**.
- Si hubiera 2 alternativas, se utilizaría una política de primero izquierdo o primero derecha.
- Se toman las claves de los **dos nodos más la entrada del nodo padre que separa a los dos nodos** y se colocan en el nuevo nodo.
- Puede que esto haga que el nodo padre pase a tener menos de d claves, con lo que se seguiría el proceso con el padre. Es decir probar primero con la **redistribución** y luego si no es posible con la **fusión** de nodos.

Borrar 12



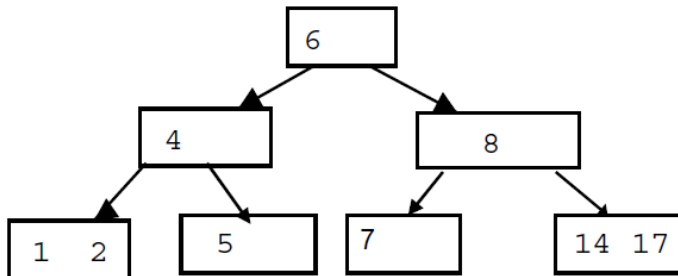
Borrar 12



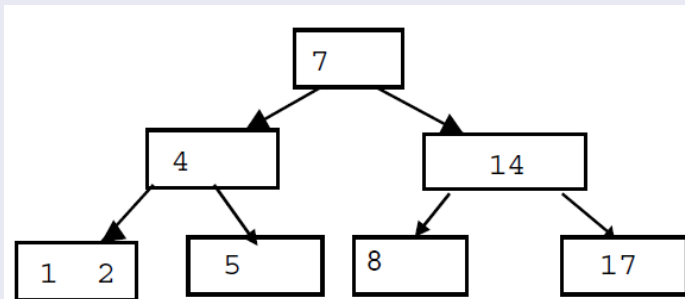
En nodos NO hoja

- Se soluciona **reemplazando** la entrada en el **nodo no hoja**, con una entrada de un **nodo hoja**.
- La entrada seleccionada, puede ser **la entrada más a la derecha del subárbol izquierdo**, o **la entrada más a la izquierda del subárbol derecho**, se utilizará una política (**primero izquierda o primero derecha**) para realizar la elección.

Borrar 6



Borrar 6



Leer un registro con una clave determinada

- El nodo raíz es el primero en ser examinado para encontrar:
 - Una **entrada que contiene el valor buscado**, y por lo tanto obtenemos el puntero al registro en el fichero de datos.
 - El **lugar en la secuencia de valores del campo de indexación** donde el valor buscado debería estar.
- Si no se encuentra, se sigue el puntero de árbol correspondiente para acceder al siguiente nodo.
- Si no hay siguiente nivel, la búsqueda termina concluyendo que el registro no está en el fichero de datos.

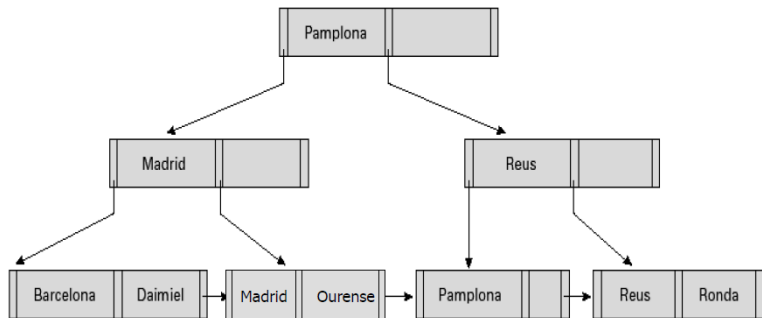
Leer todos los registros en el orden del campo de indexación

- Para leer todos los registros en orden del campo de indexación, se debe realizar un recorrido *en orden* del árbol.
- Cada vez que se lee una entrada, se accede al puntero a datos y se pasa a la siguiente entrada del índice.

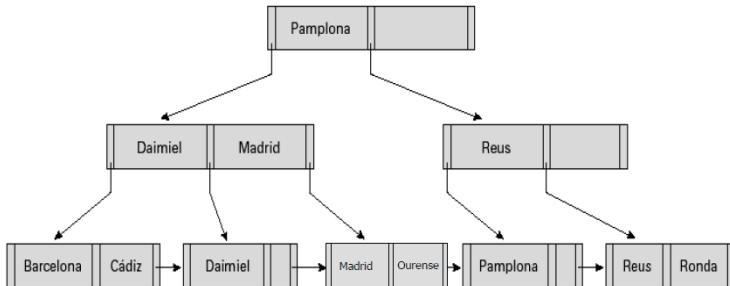
- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

- Es un **árbol heterogéneo**.
- Los valores del campo de indexación están duplicados. Algunos de ellos deben aparecer **duplicados en los nodos no hoja para guiar la búsqueda**.
- Además de los punteros a datos, cada nodo hoja tiene un **puntero al nodo hermano siguiente en la secuencia de nodos hoja**.

Árboles B+



Después de insertar "Cádiz"



- Conceptos básicos
- Métricas de evaluación de índices
- Índices basados en árboles
- Árboles heterogéneos
- Árboles homogéneos
- Diferencias entre árboles homogéneos y heterogéneos
- Árboles B
- Árboles B⁺
- Claves duplicadas en Árboles B

- Considerar las entradas duplicadas igual que las entradas normales Muchos gestores añaden un atributo adicional ficticio para considerarlas entradas distintas. Este atributo adicional tendrá un valor correlativo distinto para cada entrada con el mismo valor.
Por ejemplo, Oracle concatena el ROWID (identificador único de cada registro añadido por el propio SGBD) con el valor de la clave a indexar.
- Tener una entrada por valor, que apunta (en lugar de al registro) a una lista de punteros, que finalmente son los que apuntan al fichero de datos.