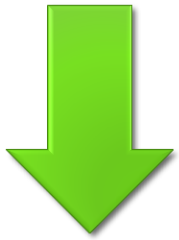
ÍNDICES EN BASES DE DATOS

Diana Benavides
José Abásolo

Basada en: Ma. Del Pilar Villamil

ÍNDICES

Mecanismo eficiente para encontrar información en la base de datos, mediante una **llave de búsqueda**.



Estructura de datos que mantiene, en orden, el resumen de información almacenada en una base de datos.

¿Ejemplos de índices en la vida cotidiana?

ÍNDICES

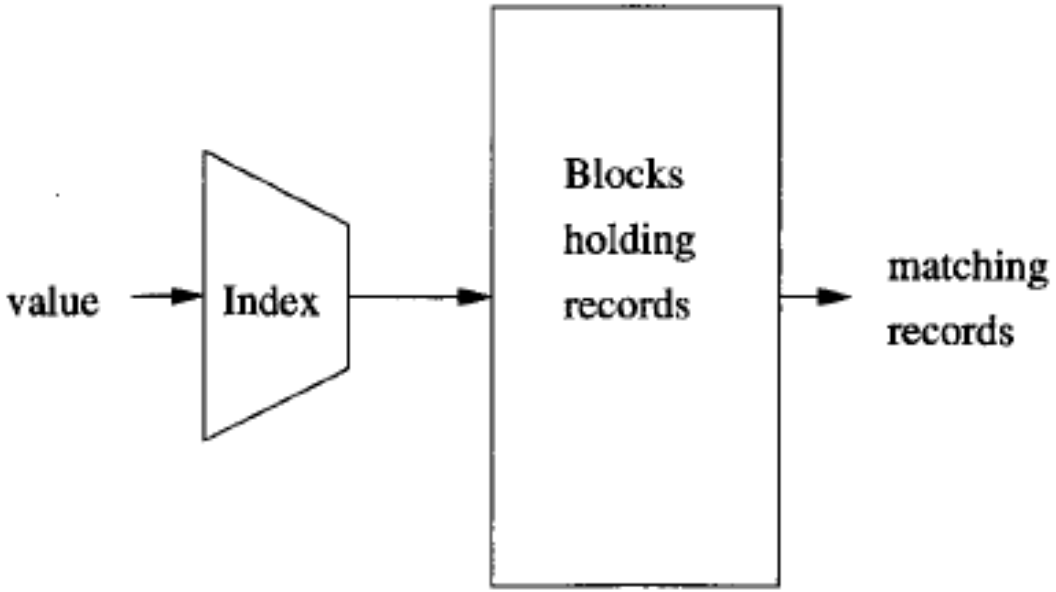


Figure 14.1: An index takes a value for some field(s) and finds records with the matching value

ÍNDICES: GENERALIDADES

Tipos de búsquedas que se pretenden facilitar con índices:

- Igualdad (exactas)
- Rangos
- Partial-key (importa el orden del índice)

ÍNDICES: GENERALIDADES

Los índices pueden almacenarse:

- **Integrados**: La misma estructura de datos mantiene los índices y los datos mismos.
- **En un archivo de índice**: Hay una estructura de datos separada para mantener los índices.

¿Implicaciones?

ÍNDICES: GENERALIDADES

Los índices pueden ser:

- **Simples**: Un solo atributo.
- **Compuestos**: Varios atributos → el orden importa.

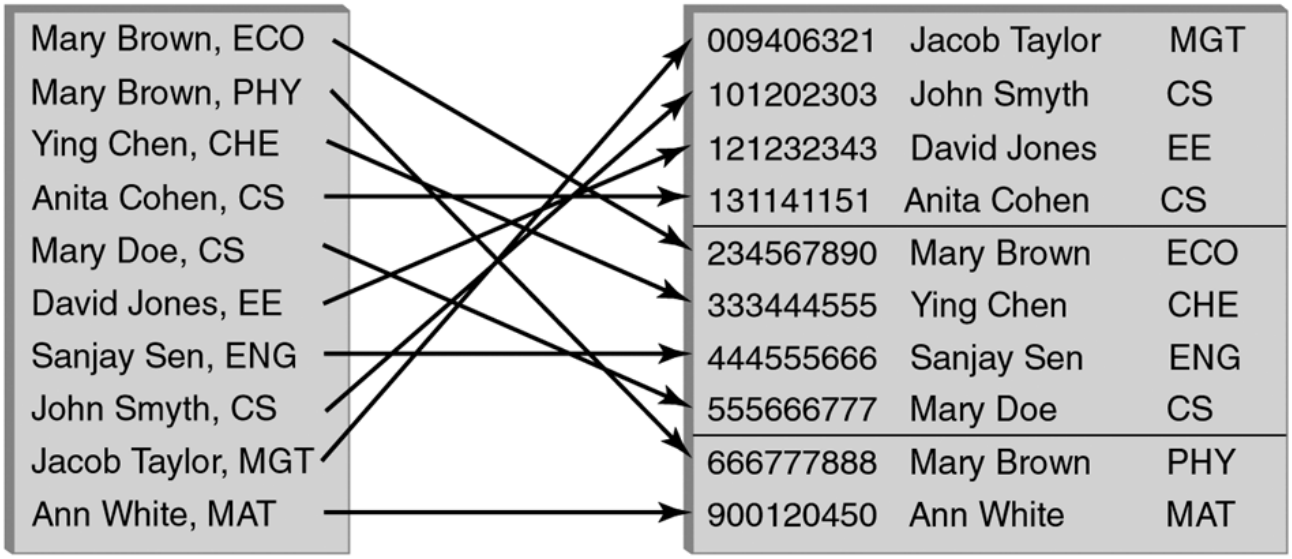


FIGURE 9.12 Dense index on PROFESSOR with search key `Name, DeptId`.

ÍNDICES: GENERALIDADES

Los índices pueden ser:

- **Densos**: Una entrada en el archivo de índice por registro de la tabla.
- **Dispersos**: Una entrada en el archivo de índice por bloque.

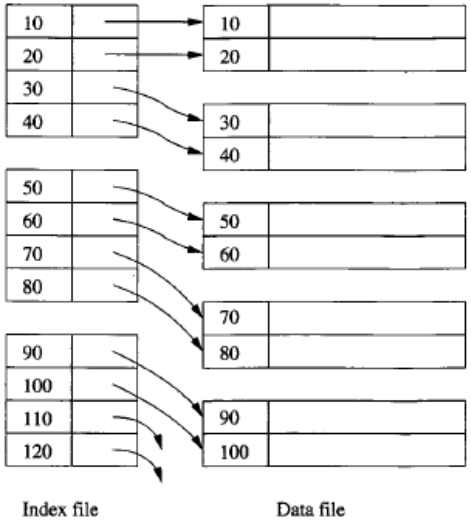


Figure 14.2: A dense index (left) on a sequential data file (right)

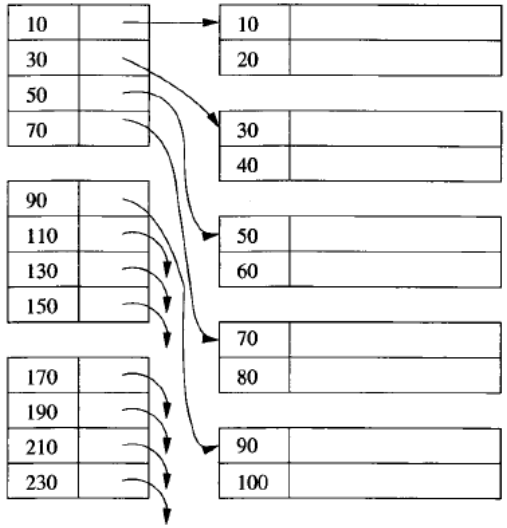


Figure 14.3: A sparse index on a sequential file

¿Cuál será “mejor”?

ÍNDICES: EJEMPLO

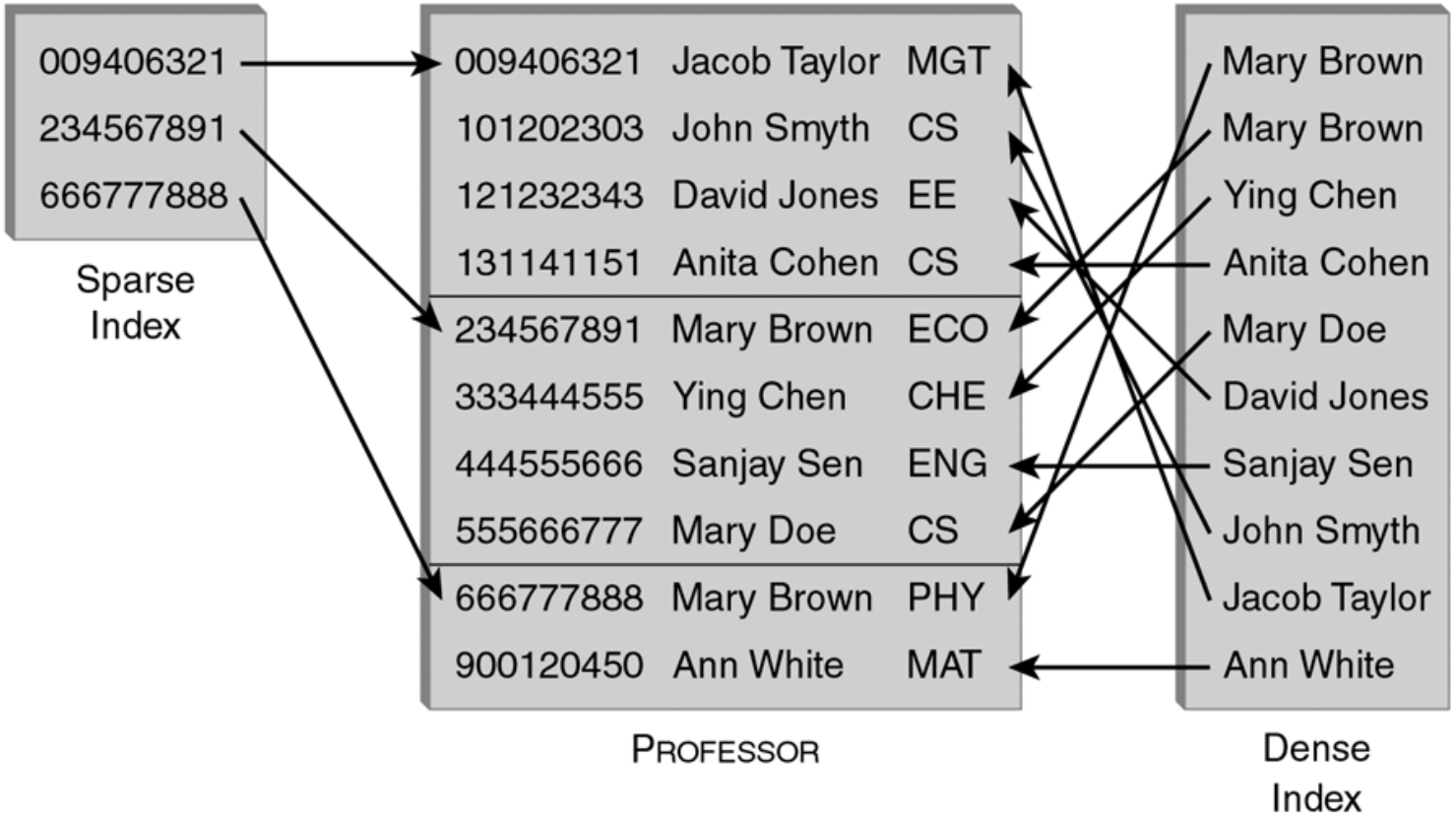


FIGURE 9.10 The index entries of (left) a sparse index with search key `Id` and (right) a dense index with search key `Name`. Both refer to the table PROFESSOR stored in a file sorted on `Id`.

ÍNDICES: GENERALIDADES

Los índices pueden ser:

- **Primarios (clustered)**: Los registros están ordenados físicamente por la clave del índice.

- **Secundarios (unclustered)**: Los registros **no** necesariamente están ordenados físicamente por la clave del índice.

ÍNDICES: GENERALIDADES

- **Índices primarios (clustered)**: Los registros están ordenados físicamente por la clave del índice.

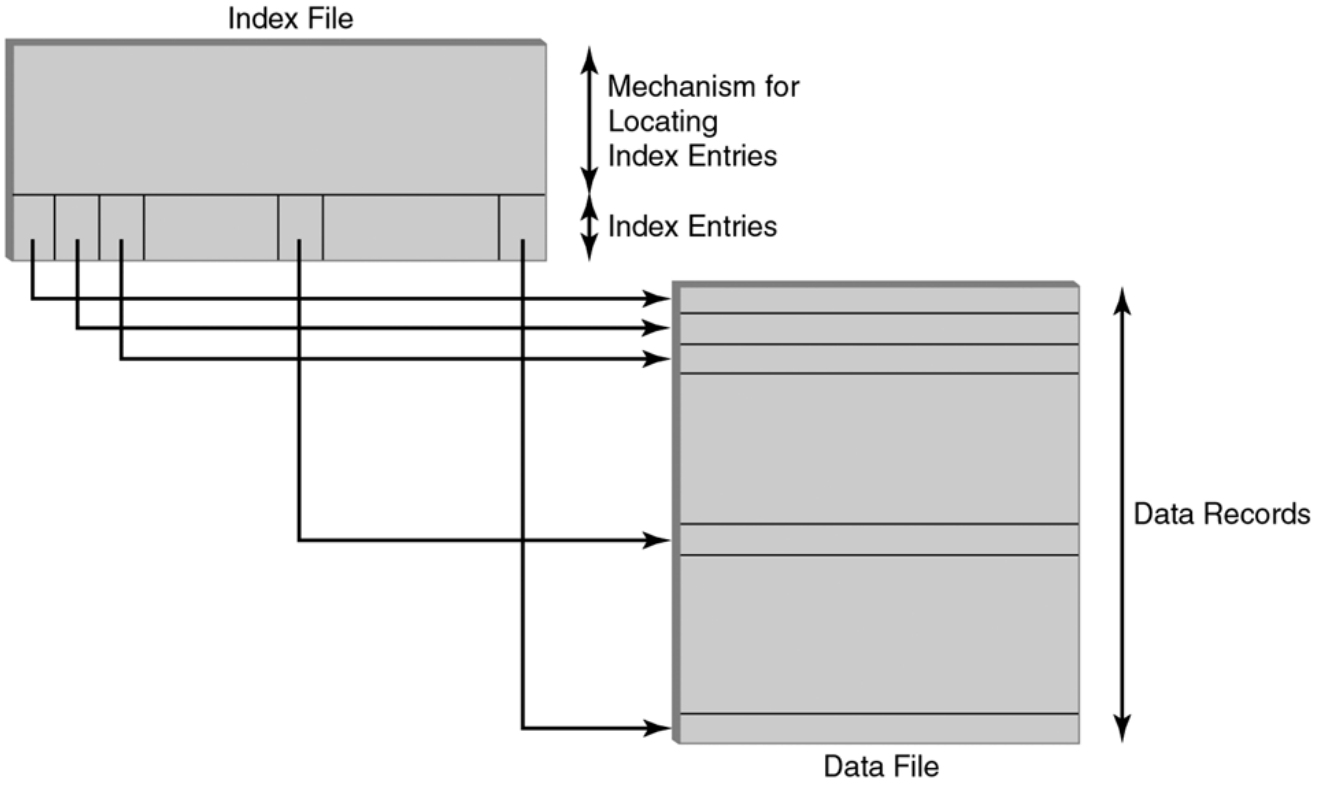


FIGURE 9.8 A clustered index that references a separate data file.

ÍNDICES: GENERALIDADES

- **Índices secundarios (unclustered)**: Los registros **no** necesariamente están ordenados físicamente por la clave del índice.

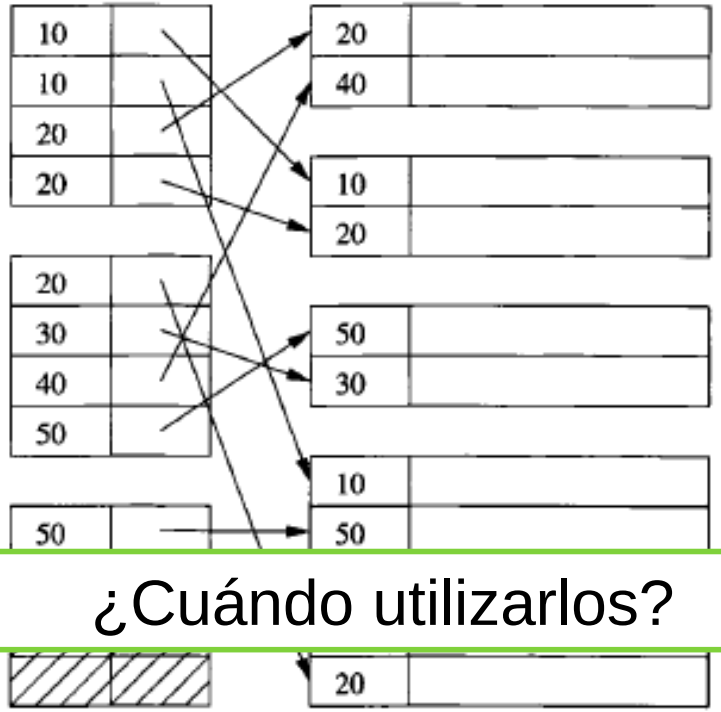


Figure 14.5: A secondary index

¿Cuándo utilizarlos?

ÍNDICES: GENERALIDADES

- **Índices secundarios (unclustered)**: Los registros **no** necesariamente están ordenados físicamente por la clave del índice.

Índice en el código del presidente del estudio

Ejemplo de uso: “Clustered file”

Índice secundario en películas

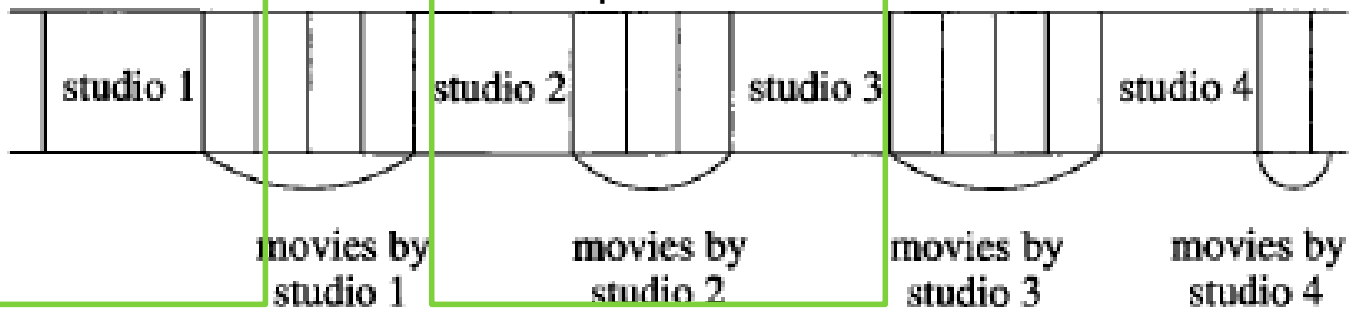
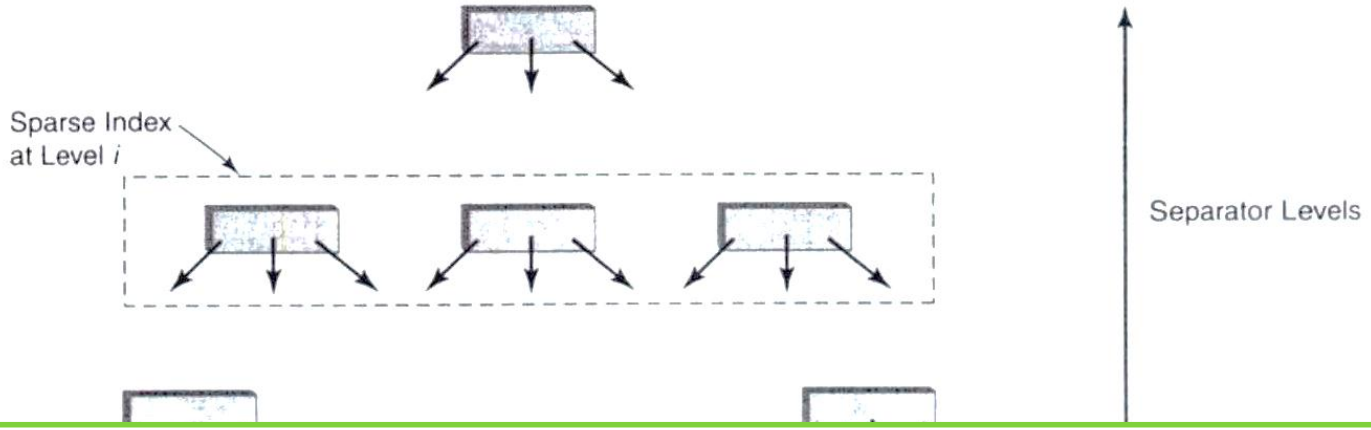


Figure 14.6: A clustered file with each studio clustered with the movies made by that studio

ÍNDICES: GENERALIDADES

Los índices pueden tener:

- **Un nivel**.
- **Múltiples niveles:** Indexar, a su vez, la lista de índices.



¿Ventajas, implicaciones?

ÍNDICES MULTINIVEL: ISAM

- Índice principal (clustered), integrado.
- Cada nivel raíz o intermedio del índice tiene índices dispersos.
- Los nodos hoja están ordenados.
- Son estáticos (los niveles intermedios no cambian), no muy apropiados para datos que cambian de forma dinámica. Generalmente no son usados en los DBMS debido a esto.

ÍNDICES MULTINIVEL: ISAM

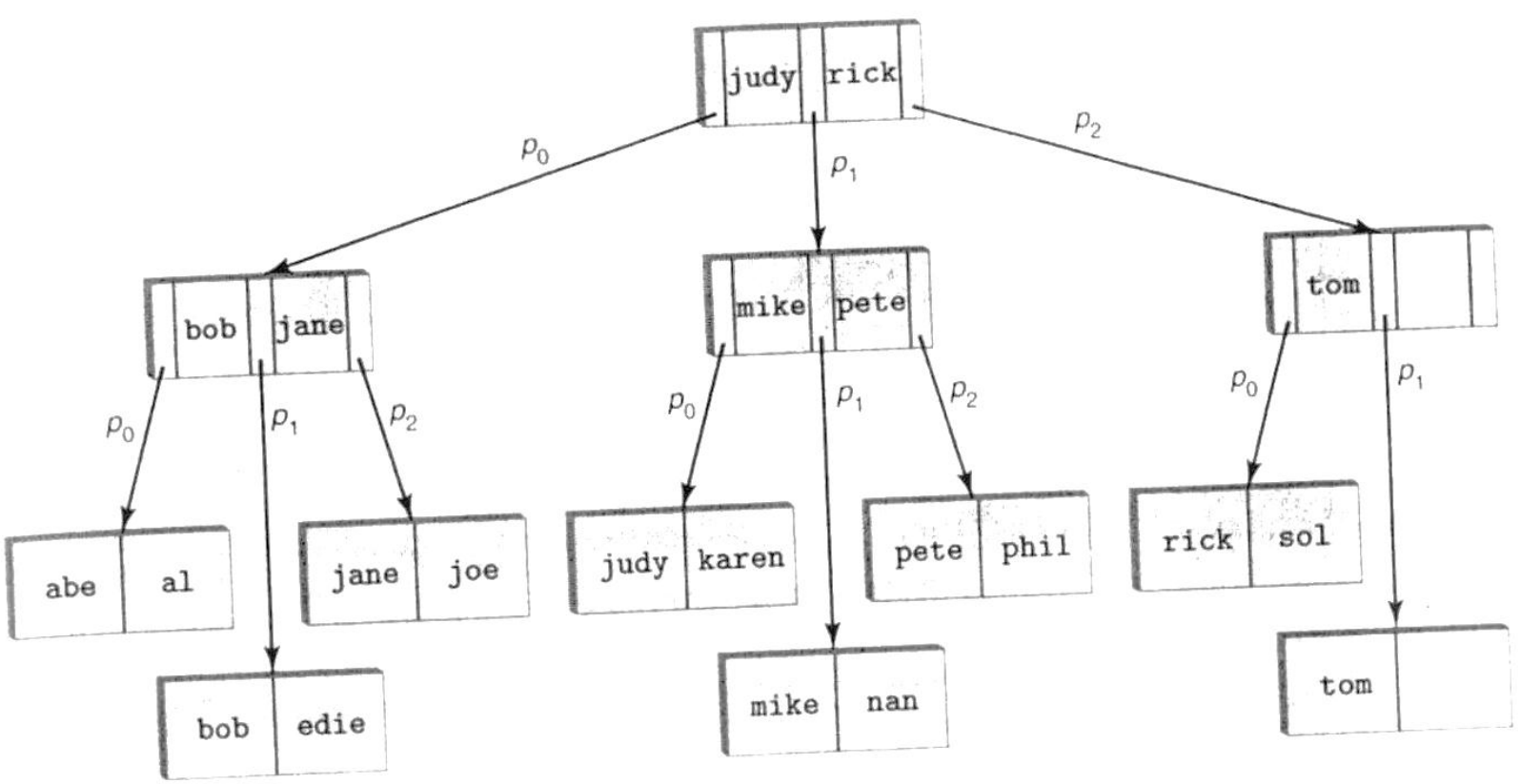
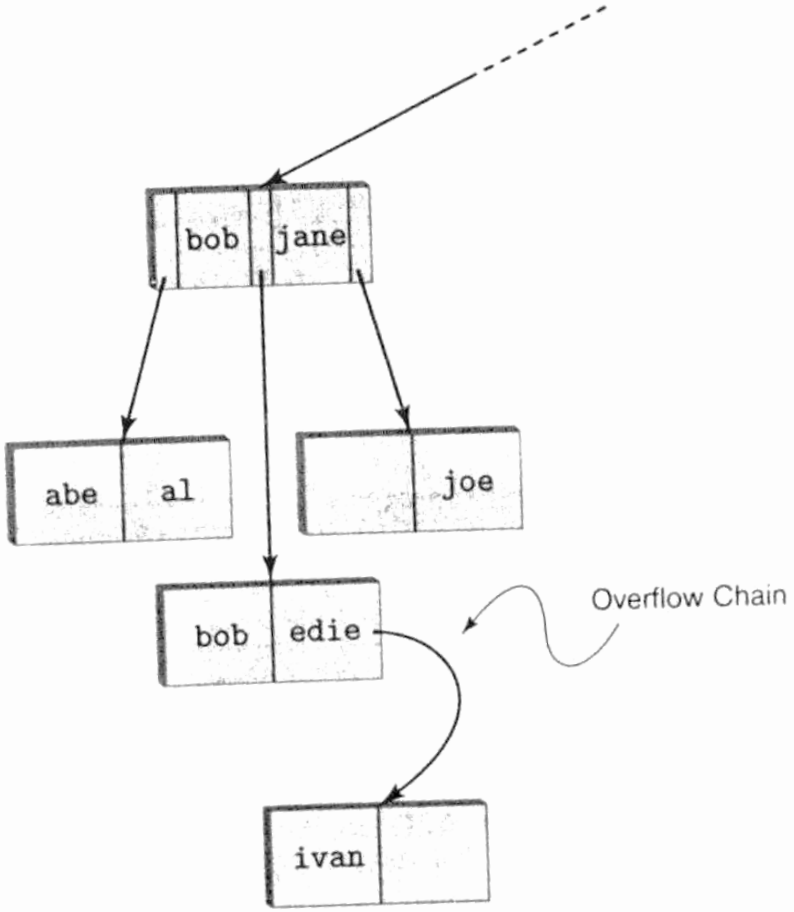


FIGURE 9.16 An example of an ISAM index.

ÍNDICES MULTINIVEL: ISAM

FIGURE 9.17 Portion of the ISAM index of Figure 9.16 after an insertion and a deletion.



ÍNDICES MULTINIVEL: ÁRBOLES B+

- Árbol balanceado.
- Puede ser integrado (los nodos hoja pueden o no contener los datos mismos) o no.
- Índice principal (clustered) si es integrado, principal o secundario (unclustered) si no lo es.
- Dinámico (los nodos intermedios pueden reorganizarse).
- Maneja, además, apuntadores entre hojas “hermanas”.

ÍNDICES MULTINIVEL: ÁRBOLES B+

- Reglas:
  - **Número índices y apuntadores:** En cada nodo o bloque, pueden haber máximo **n** índices y **n+1** apuntadores a bloques o nodos del siguiente nivel.
  - El nodo raíz tiene al menos dos apuntadores.
  - El último apuntador de cada nodo hoja apunta al primer índice del nodo siguiente.
  - **Apuntadores de nivel intermedio:** Al menos techo (n+1)/2 apuntadores de cada nodo intermedio apuntan a bloques o nodos del siguiente nivel.
  - **Apuntadores de hojas:** Al menos piso (n+1)/2 apuntadores de cada nodo hoja apuntan a registros de datos.
  - Todos los registros de cada bloque o nodo utilizados aparecen al inicio del bloque.

ÍNDICES MULTINIVEL: ÁRBOLES B+

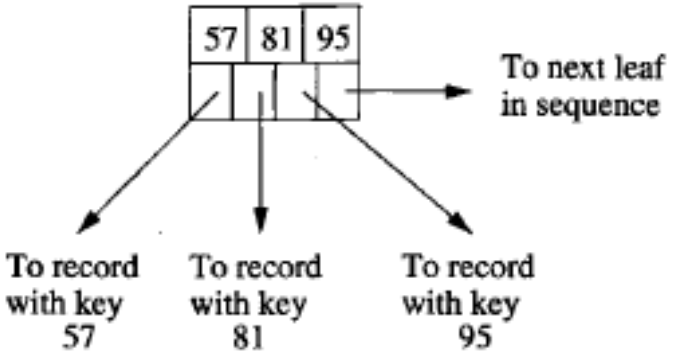


Figure 14.11: A typical leaf of a B-tree

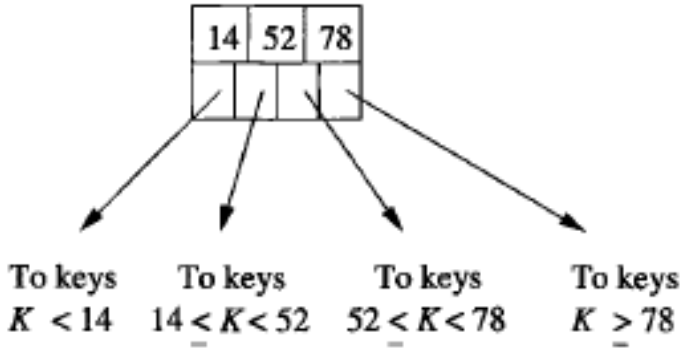


Figure 14.12: A typical interior node of a B-tree

ÍNDICES MULTINIVEL: ÁRBOLES B+

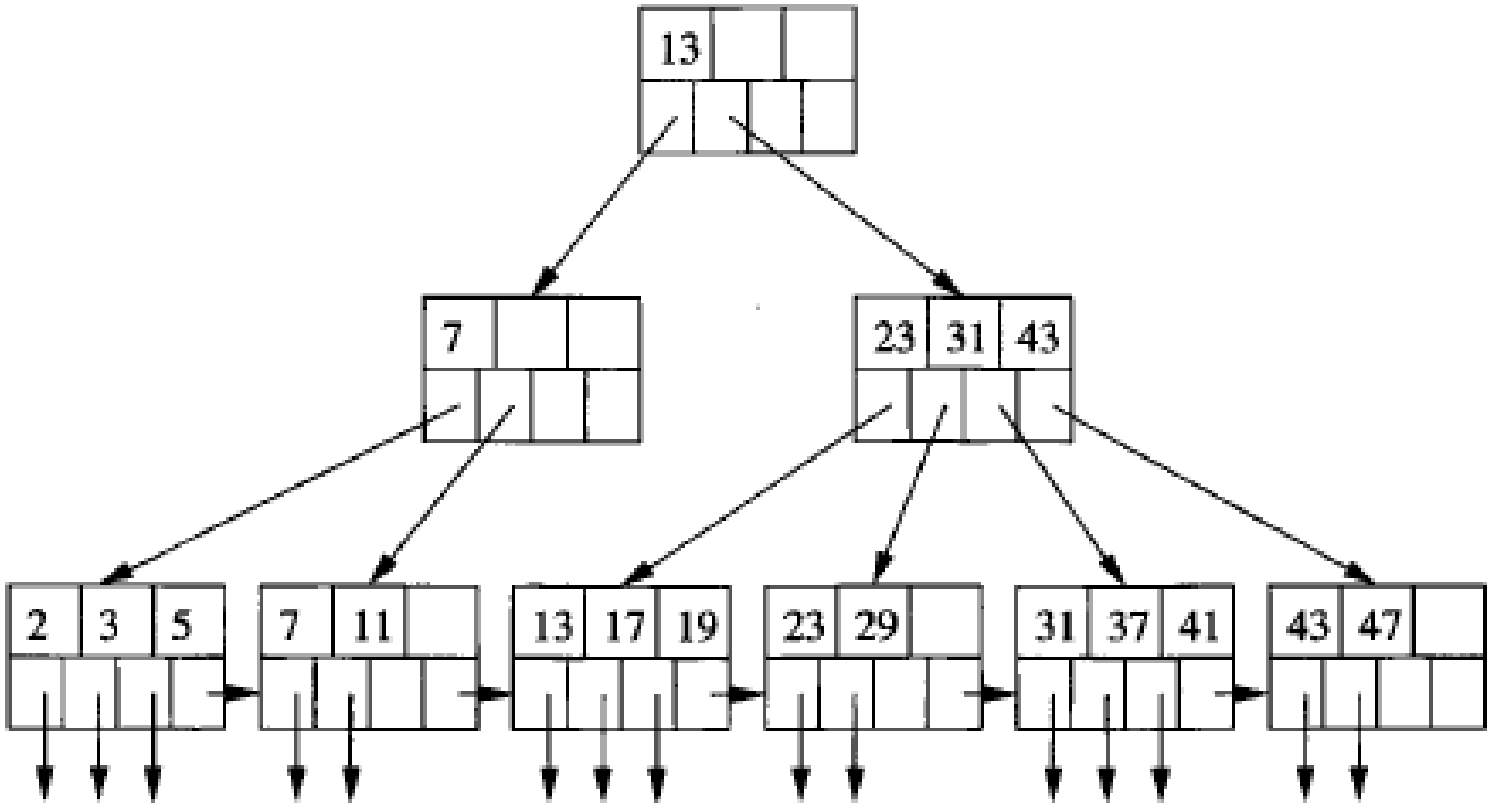


Figure 14.13: A B-tree

ÍNDICES MULTINIVEL: ÁRBOLES B+: EJEMPLO CADENAS

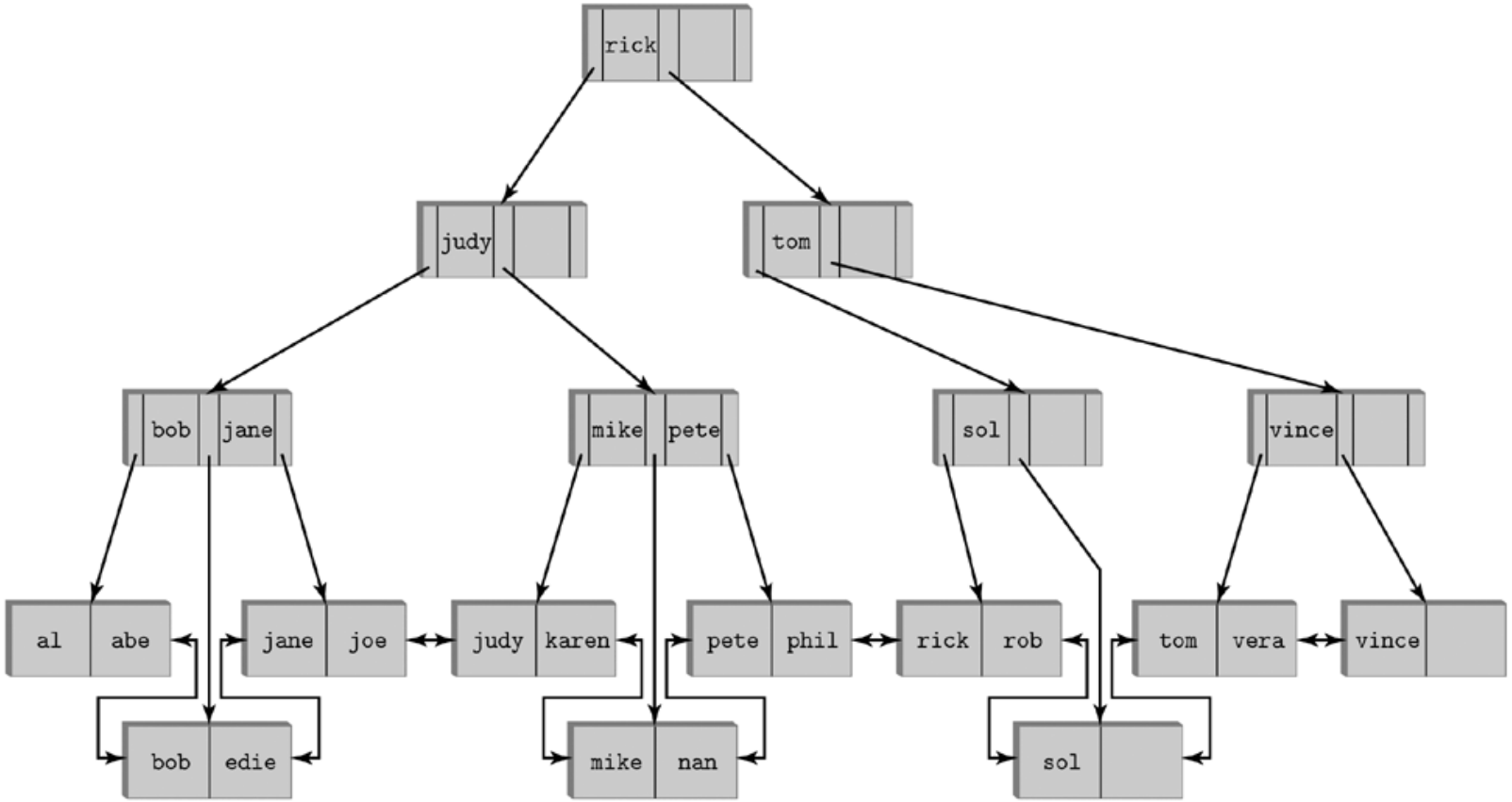


FIGURE 9.22 B+ tree that results from the insertion of `vince`, `vera`, and `rob` into the index of Figure 9.16.

ÍNDICES MULTINIVEL: ÁRBOLES B+ - BÚSQUEDA

Buscar 37

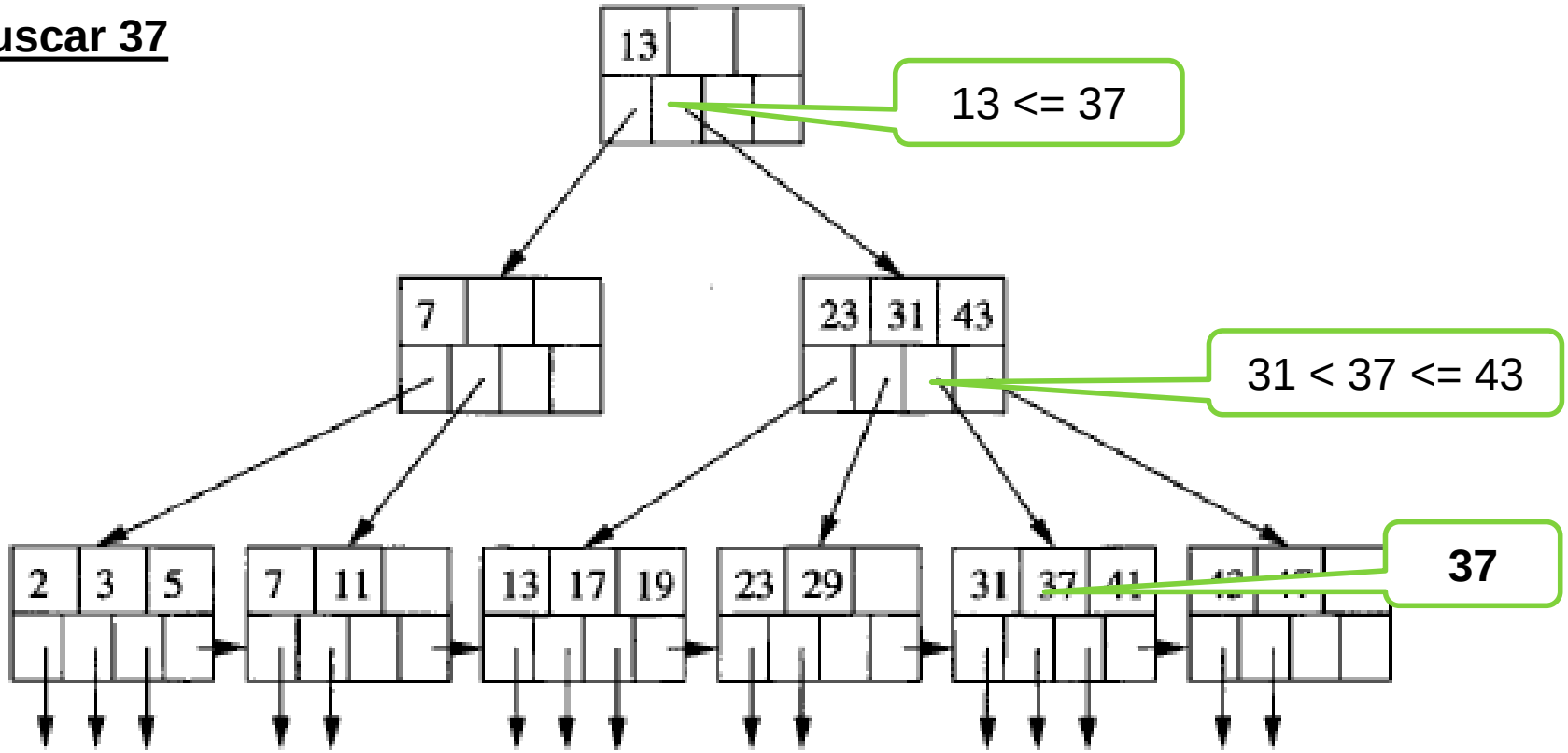
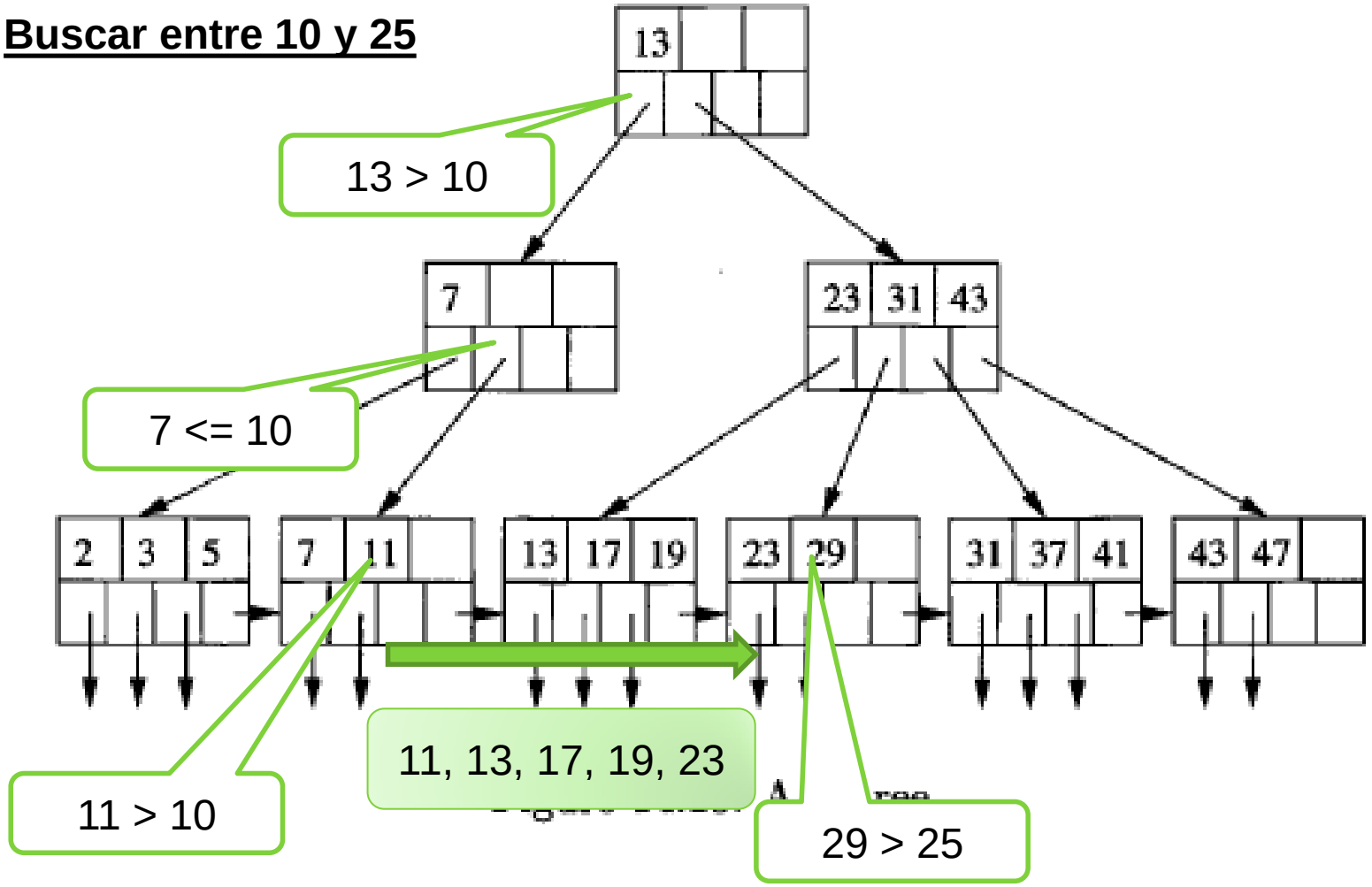


Figure 14.13: A B-tree

ÍNDICES MULTINIVEL: ÁRBOLES B+ - BÚSQUEDA

Buscar entre 10 y 25



ÍNDICES MULTINIVEL: ÁRBOLES B+ - INSERCIÓN

1. Encontrar espacio en el nodo hoja apropiado.
2. Si no hay espacio en el nodo hoja, insertar un nuevo bloque o nodo y dividir las llaves existentes entre los dos bloques resultantes, para cumplir **reglas de apuntadores de nivel intermedio y de nodos hoja**.
3. Actualizar, recursivamente, los nodos de nivel superior; si hay espacio, insertar el índice apropiado; si no, dividir igual que la estrategia en 2).

ÍNDICES MULTINIVEL: ÁRBOLES B+ - INSERCIÓN

Insertar 40

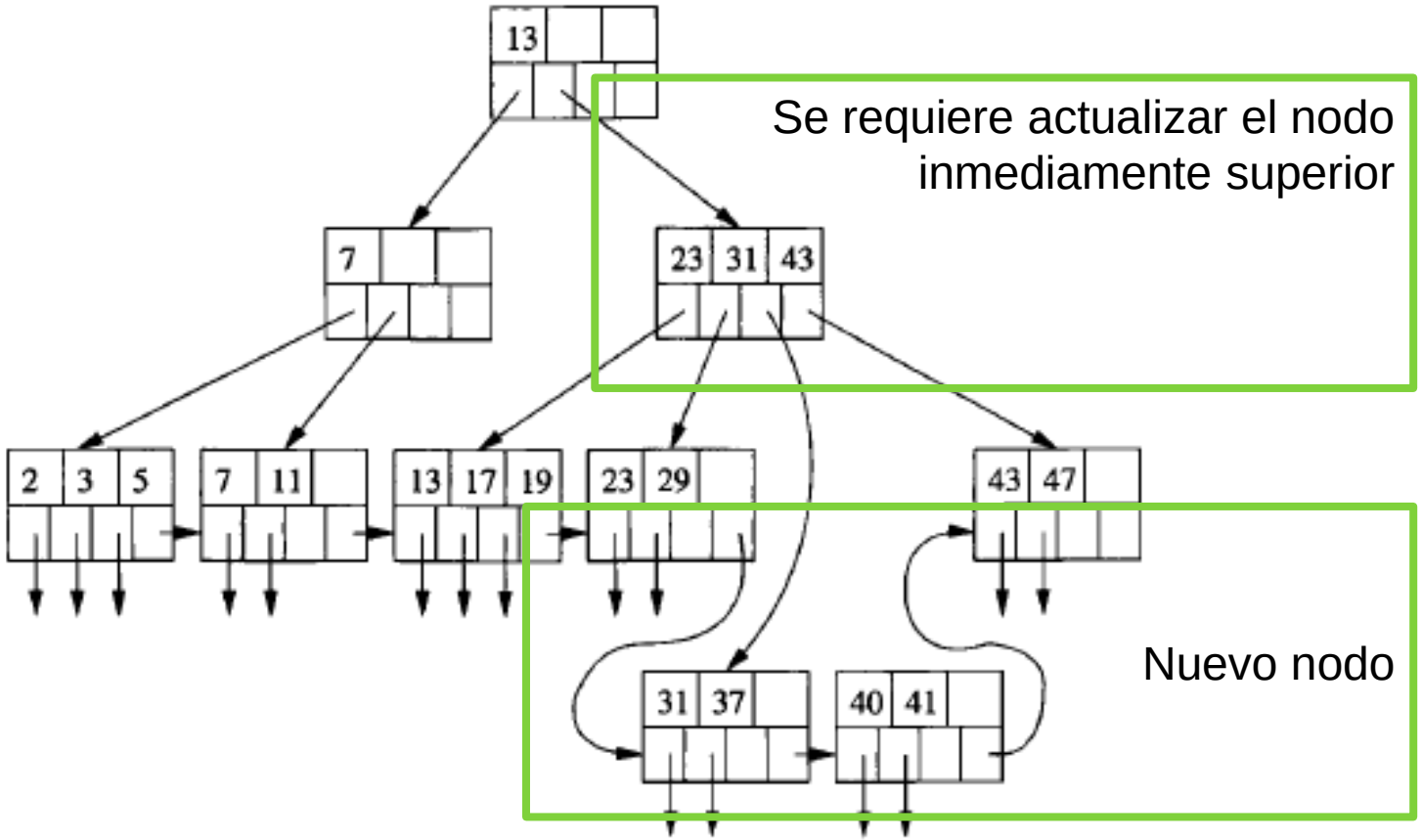


Figure 14.15: Beginning the insertion of key 40

ÍNDICES MULTINIVEL: ÁRBOLES B+ - INSERCIÓN

Insertar 40

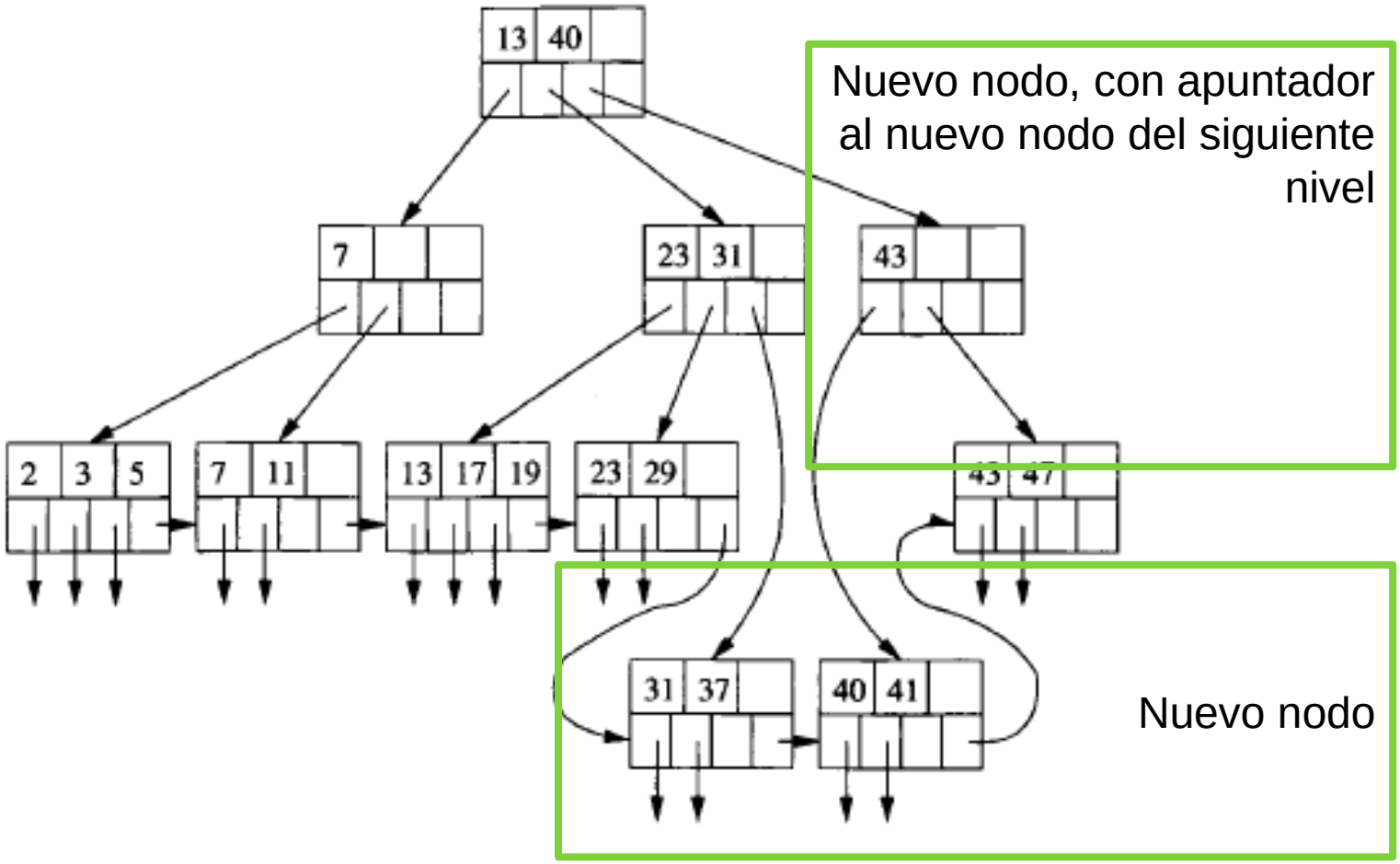


Figure 14.16: Completing the insertion of key 40

ÍNDICES MULTINIVEL: ÁRBOLES B+ - ELIMINACIÓN

1. Encontrar el índice a eliminar.
2. Si se viola la regla de **apuntadores de nodo hoja**:
    1. Si alguno de los hermanos del nodo del que se eliminó el registro de índice tiene más de la regla (piso n+1/2), pasar el primer (o último) registro de ese hermano al nodo actual. Podría requerir actualización de nodos del nivel inmediatamente anterior.
    2. Si no, mezclar el nodo del que se eliminó el registro de índice con el hermano. Podría requerir actualización de nodos del nivel inmediatamente anterior.

ÍNDICES MULTINIVEL: ÁRBOLES B+ - ELIMINACIÓN

Eliminar 7

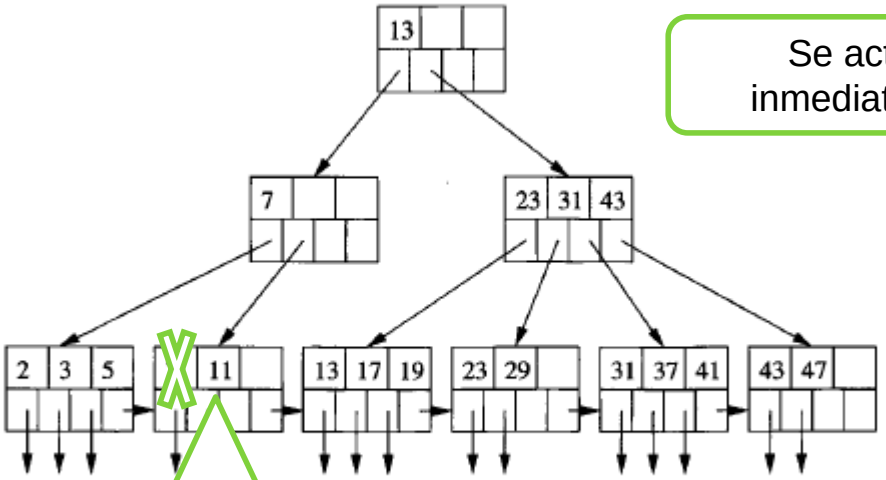


Figure 14.13: A B-tree

La hoja contendrá menos del mínimo requerido (piso n+1/2).

Se actualiza el nodo inmediatamente anterior.

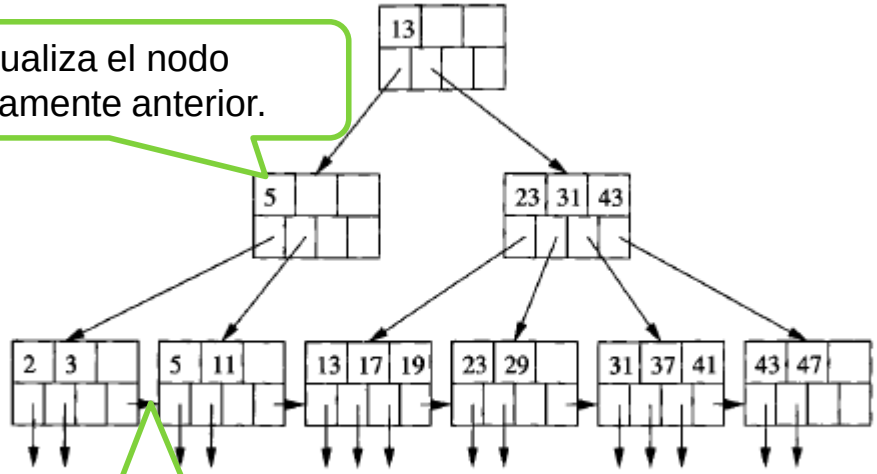


Figure 14.17: Deletion of key 7

Se pasa el último registro del hermano izquierdo. Los dos hermanos obedecerán a la regla.

ÍNDICES: HASHING

- Índice integrado, primario (clustered) o con un archivo de índice, secundario (unclustered).
- Dos tipos: 1) Estático → Si los datos cambian poco. 2) Dinámico (extendible o lineal).
- Agrupación de índices en “buckets”.

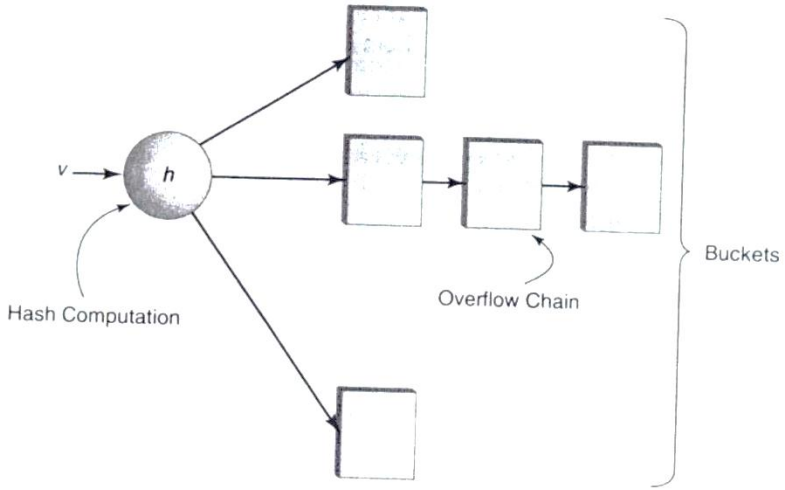


FIGURE 9.28 Schematic depiction of a hash index.

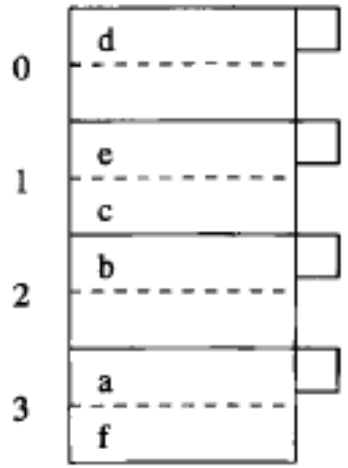


Figure 14.20: A hash table

ÍNDICES: HASHING - INSERCIÓN Y ELIMINACIÓN

- Inserción: Se aplica la función y se añade al “bucket” correspondiente.
- Eliminación: Se aplica la función, se encuentra el registro y se elimina.

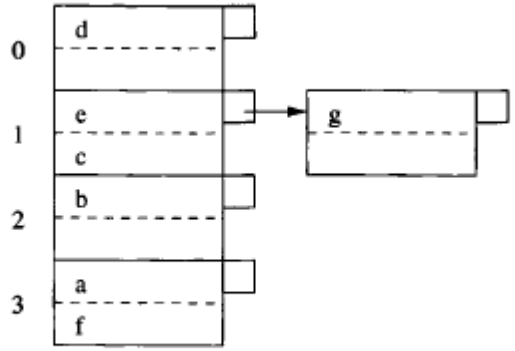


Figure 14.21: Adding an additional block to a hash-table bucket

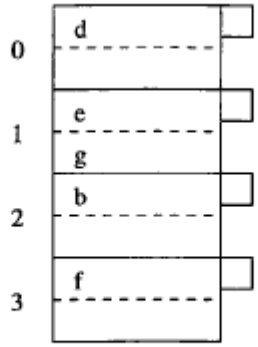


Figure 14.22: Result of deletions from a hash table

ÍNDICES: ¿CUÁLES ÍNDICES Y CUÁL MECANISMO UTILIZAR?

- Evaluar las necesidades de la aplicación: ¿Cuáles consultas son las más típicas?
  - Si hay búsquedas de igualdad, quizá sea más eficiente el mecanismo de hashing.
  - Si hay búsquedas de rangos o llaves parciales, quizá el mecanismo de árbol sea más eficiente.
    - ISAM si los índices son más bien estáticos; si no, árboles B+.
- Evaluar el costo:
  - En hashing, el costo depende del número de registros.
  - En árboles, el costo depende del número de nodos hoja.

ÍNDICES: ¿CUÁNDO UTILIZARLOS?

- Selectividad: **Porcentaje de filas que arroja una consulta (Si está alrededor de un 25% -Oracle- ó 20% teórico).**

- En el WHERE
  - *columna1 = constante → índice sobre columna1*
  - *Rangos: columna1 between val1 and val2*

ÍNDICES: ¿CUÁNDO UTILIZARLOS?

- *Funciones agregadas (AVG, SUM, MAX), sin where o group by.*
- *Con ORDER BY sobre columnas que tengan un índice, y un constraint de no nulidad.*
- Cuando existe un índice compuesto no se utilizará si hay una condición del where sobre una columna que no sea cabeza o PORCIÓN de un índice

ÍNDICES: NO SE USAN SI…

- Las condiciones sobre las columnas son:
  - *IS NULL, IS NOT NULL*
  - *NOT IN*
  - *!= expr*
  - *LIKE ‘%valor’*
  - *NOT EXISTS*
- Selectividad baja.

ÍNDICES: ¿SE UTILIZAN?

- Select avg(edad) from estudiantes;
- Select avg(edad) from estudiantes where codigo > 2000;
- Select codigo, nombre from estudiantes order by nombre;
- Select * from Estudiantes where año_graduado is not null;
- Select * from prod_existencias where cod_producto = 123;

ÍNDICES: ¿SE UTILIZAN?

- Simples
  - Selectividad: ± 25%
  - Columnas frecuentemente usadas en el WHERE
  - Columnas usadas frecuentemente en JOINS (FK)
- Compuestos
  - Columnas frecuentemente usadas con el conector AND, con selectividad menor a la obtenida en combinación.

DISEÑO FÍSICO: ¿EN DÓNDE VAMOS?

- Jerarquías de memoria.
- Ordenamiento externo.
- Optimización de consultas.
- **Índices.**

Reducir la cantidad de operaciones de I/O a memoria secundaria.

TRABAJO EN GRUPO