

ORGANIZACIÓN DE LOS ARCHIVOS

Los archivos se organizan lógicamente como secuencias de registros. Estos registros se corresponden con los bloques del disco. Los archivos se proporcionan como un instrumento fundamental de los sistemas operativos, por lo que se supondrá la existencia de un sistema de archivos subyacente. Hay que tomar en consideración diversas maneras de representar los modelos lógicos de datos en términos de archivos. Aunque los bloques son de un tamaño fijo determinado por las propiedades físicas del disco y por el sistema operativo, los tamaños de los registros varían. **En las bases de datos relacionales las tuplas de las diferentes relaciones suelen ser de tamaños distintos.**

Los archivos con registros de longitud fija son más sencillos de implementar que los archivos con registros de longitud variable.

Bibliografía:

Fundamentos de Bases de Datos 4ta edición, Abraham Silberschatz Bell Laboratories Henry F. Korth Bell Laboratories S. Sudarshan Instituto Indio de Tecnología, Bombay

ORGANIZACIÓN DE LOS REGISTROS EN ARCHIVOS

- **Organización de archivos en montículo.** En esta organización se puede colocar cualquier registro en cualquier parte del archivo en que haya espacio suficiente. No hay ninguna ordenación de los registros. Generalmente sólo hay un archivo por cada relación.
- **Organización de archivos secuenciales.** En esta organización los registros se guardan en orden secuencial, basado en el valor de la clave de búsqueda de cada registro.
- **Organización asociativa (hash) de archivos.** En esta organización se calcula una función de asociación (hash) de algún atributo de cada registro. El resultado de la función de asociación especifica el bloque del archivo en que se deberá colocar el registro.

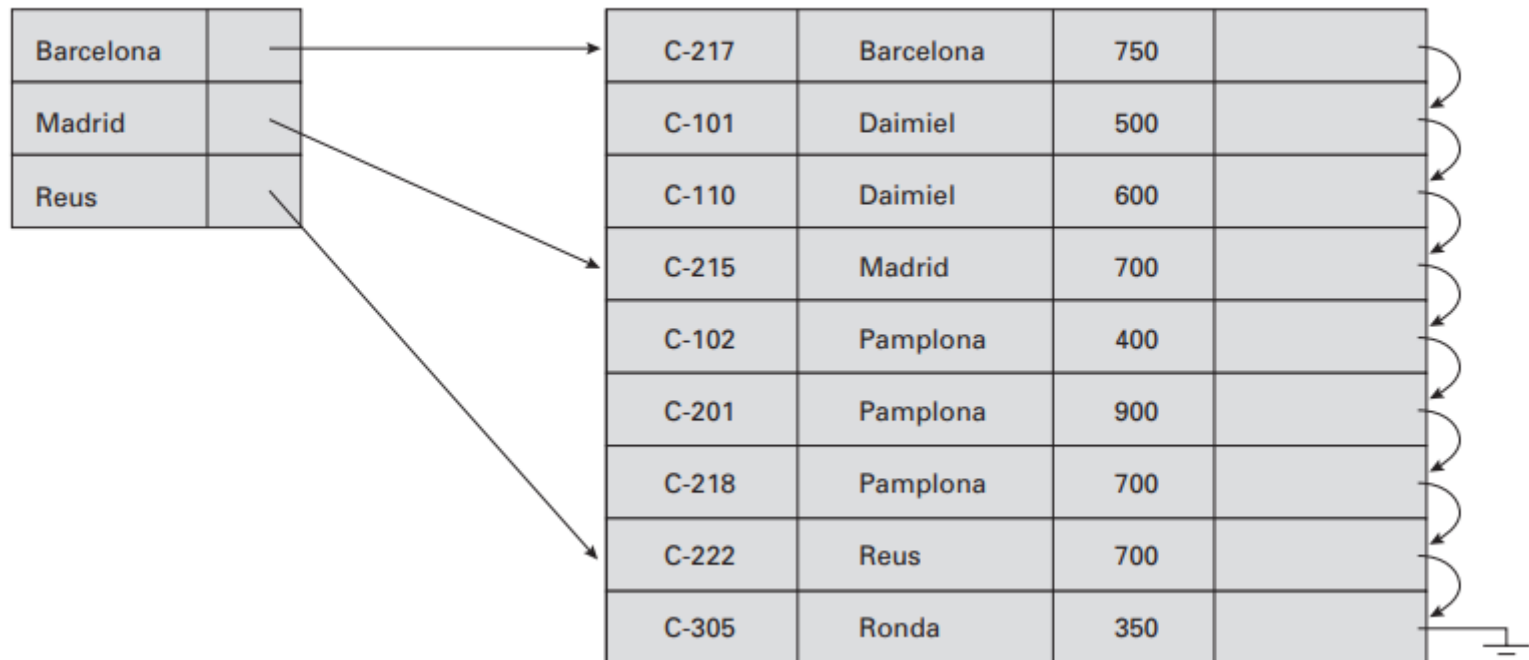
INDEXACIÓN Y ASOCIACIÓN

- **Índices ordenados.** Estos índices están basados en una disposición ordenada de los valores.
- **Índices asociativos (hash indices).** Estos índices están basados en una distribución uniforme de los valores a través de una serie de cajones (buckets). El valor asignado a cada cajón está determinado por una función, llamada función de asociación (hash function).

ÍNDICES ORDENADOS

Índice primario están ordenados secuencialmente según alguna clave de búsqueda

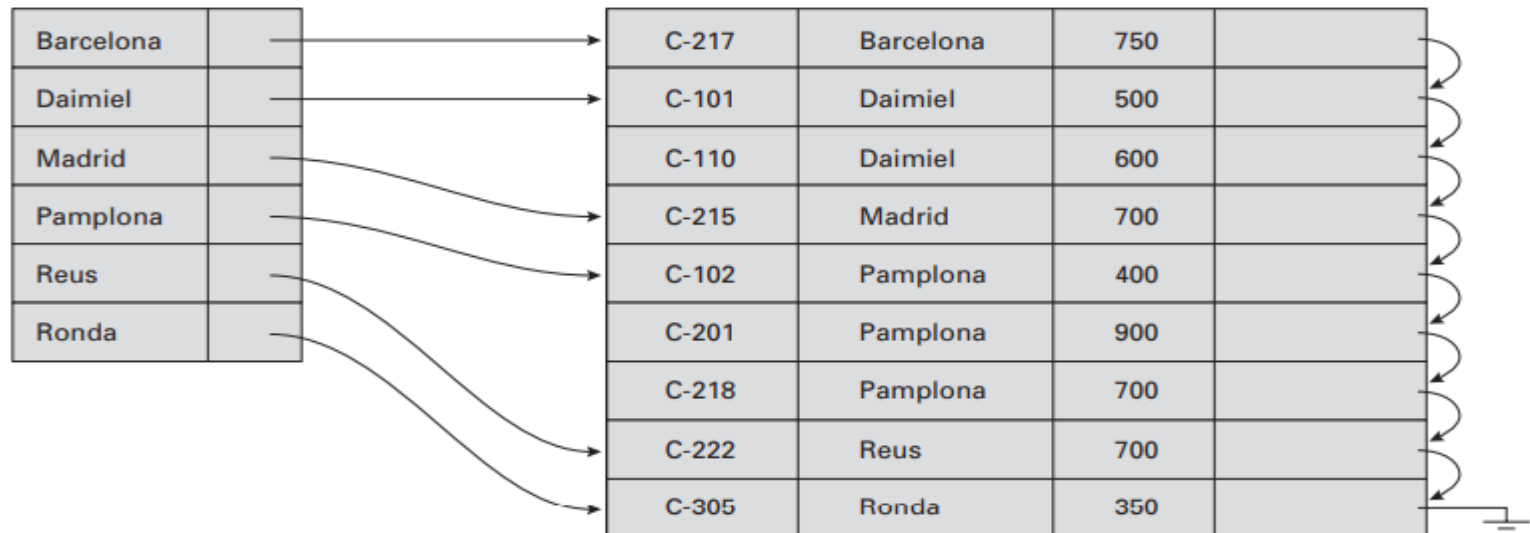
Estos archivos con índice primario según una clave de búsqueda se llaman archivos secuenciales indexados. Representan uno de los esquemas de índices más antiguos usados por los sistemas de bases de datos. Se emplean en aquellas aplicaciones que demandan un procesamiento secuencial del archivo completo así como un acceso directo a sus registros.



Índices densos y dispersos

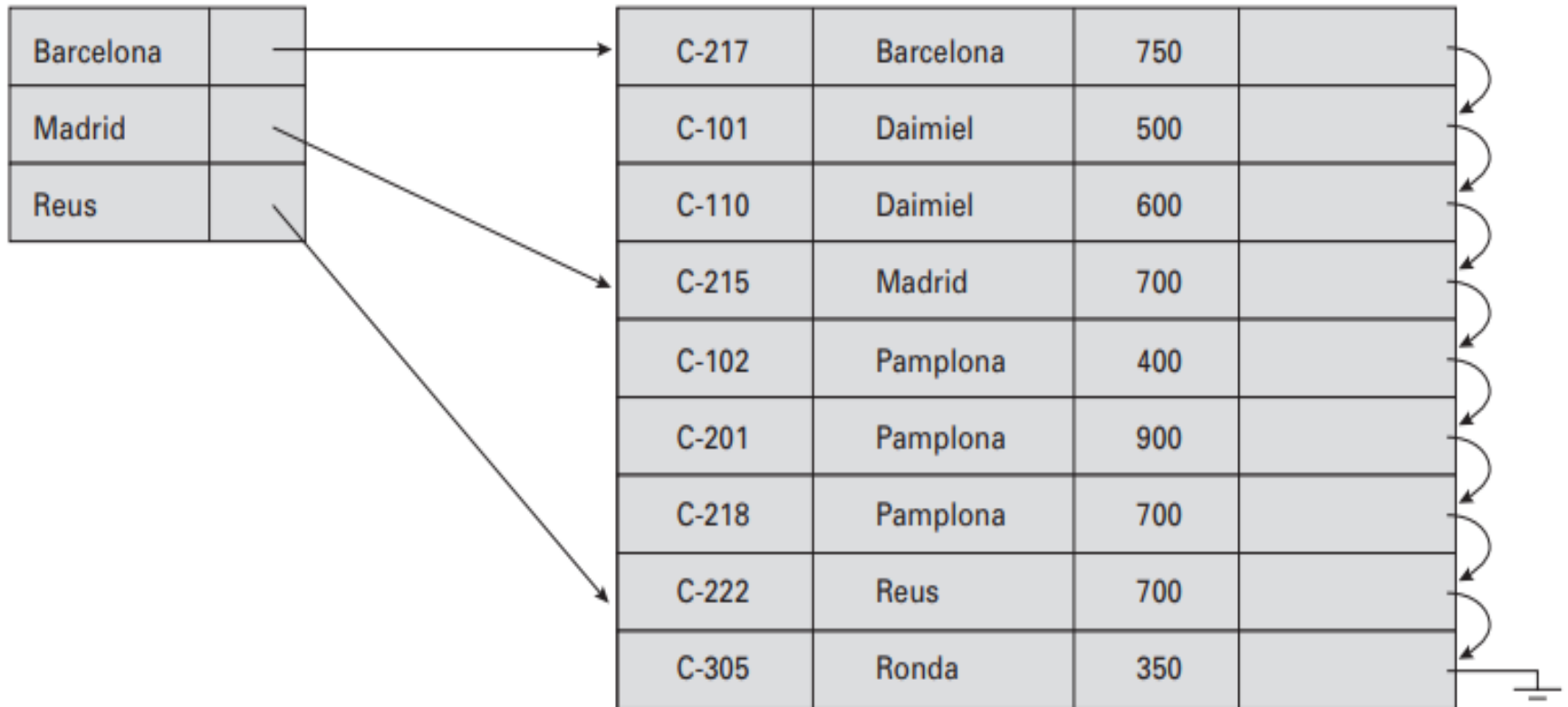
Un registro índice o entrada del índice consiste en un valor de la clave de búsqueda y punteros a uno o más registros con ese valor de la clave de búsqueda. El puntero a un registro consiste en el identificador de un bloque de disco y un desplazamiento en el bloque de disco para identificar el registro dentro del bloque. Hay dos clases de índices ordenados que se pueden emplear:

- Índice denso. Aparece un registro índice por cada valor de la clave de búsqueda en el archivo. El registro índice contiene el valor de la clave y un puntero al primer registro con ese valor de la clave de búsqueda. El resto de registros con el mismo valor de la clave de búsqueda se almacenan consecutivamente después del primer registro, dado que, ya que el índice es primario, los registros se ordenan sobre la misma clave de búsqueda. Las implementaciones de índices densos pueden almacenar una lista de punteros a todos los registros con el mismo valor de la clave de búsqueda; esto no es esencial para los índices primarios.



Índices densos y dispersos

- Índice disperso. Sólo se crea un registro índice para algunos de los valores. Al igual que en los índices densos, cada registro índice contiene un valor de la clave de búsqueda y un puntero al primer registro con ese valor de la clave. Para localizar un registro se busca la entrada del índice con el valor más grande que sea menor o igual que el valor que se está buscando. Se empieza por el registro apuntado por esa entrada del índice y se continúa con los punteros del archivo hasta encontrar el registro deseado.



Índices multinivel

Incluso si se usan índices dispersos, el propio índice podría ser demasiado grande para un procesamiento eficiente. En la práctica no es excesivo tener un archivo con 100.000 registros, con 10 registros almacenados en cada bloque. Si tenemos un registro índice por cada bloque, el índice tendrá 10.000 registros. Como los registros índices son más pequeños que los registros de datos, podemos suponer que caben 100 registros índices en un bloque. Por tanto, el índice ocuparía 100 bloques. Estos índices de mayor tamaño se almacenan como archivos secuenciales en disco. Si un índice es lo bastante pequeño como para guardarlo en la memoria principal, el tiempo de búsqueda para encontrar una entrada será breve. Sin embargo, si el índice es tan grande que se debe guardar en disco, buscar una entrada implicará leer varios bloques de disco. Para localizar una entrada en el archivo índice se puede realizar una búsqueda binaria, pero aun así ésta conlleva un gran coste. Si el índice ocupa b bloques, la búsqueda binaria tendrá que leer a lo sumo $\log_2(b)$ bloques. Para el índice de 100 bloques, la búsqueda binaria necesitará leer siete bloques. En un disco en el que la lectura de un bloque tarda 30 milisegundos, la búsqueda empleará 210 milisegundos, lo que es mucho. Obsérvese que si se están usando bloques de desbordamiento, la búsqueda binaria no sería posible. En ese caso, lo normal es una búsqueda secuencial, y eso requiere leer b bloques, lo que podría consumir incluso más tiempo. Así, el proceso de buscar en un índice grande puede ser muy costoso. Para resolver este problema se trata el índice como si fuese un archivo secuencial y se construye un índice disperso sobre el índice primario, como se muestra. Para localizar un registro se usa en primer lugar una búsqueda binaria sobre el índice más externo. El puntero apunta a un bloque en el índice más interno. El puntero de este registro apunta al bloque del archivo que contiene el registro buscado.

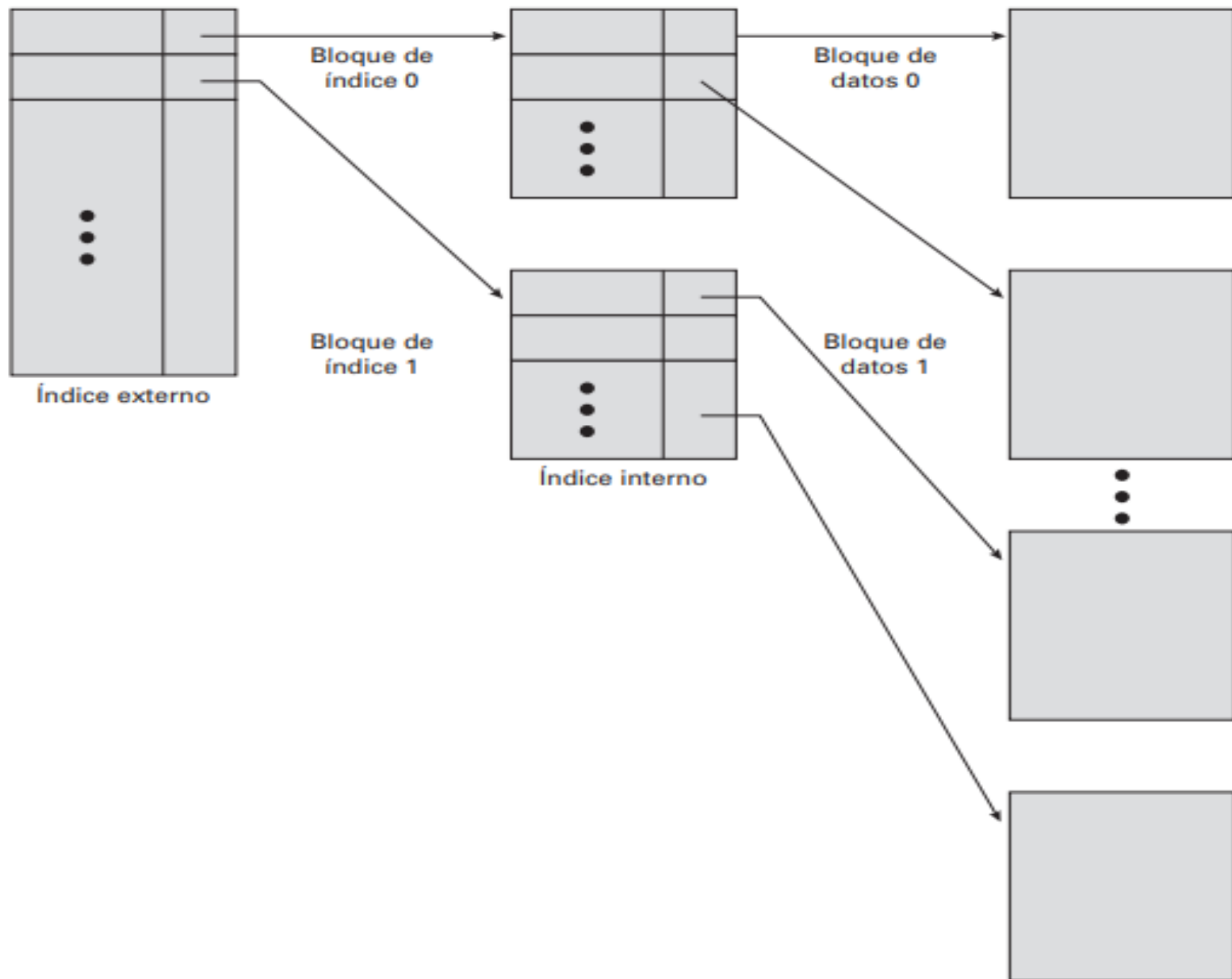


FIGURA 12.4. Índice disperso de dos niveles.

Índices secundarios

Los índices secundarios deben ser densos, con una entrada en el índice por cada valor de la clave de búsqueda, y un puntero a cada registro del archivo. Un índice primario puede ser disperso, almacenando sólo algunos de los valores de la clave de búsqueda, ya que siempre es posible encontrar registros con valores de la clave de búsqueda intermedios mediante un acceso secuencial a parte del archivo. Si un índice secundario almacena sólo algunos de los valores de la clave de búsqueda, los registros con los valores de la clave de búsqueda intermedios pueden estar en cualquier lugar del archivo y, en general, no se pueden encontrar sin explorar el archivo completo. Un índice secundario sobre una clave candidata es como un índice denso primario, excepto en que los registros apuntados por los sucesivos valores del índice no están almacenados secuencialmente. Por lo general, los índices secundarios están estructurados de manera diferente a como lo están los índices primarios. Si la clave de búsqueda de un índice primario no es una clave candidata, es suficiente si el valor de cada entrada en el índice apunta al primer registro con ese valor en la clave de búsqueda, ya que los otros registros podrían ser alcanzados por una búsqueda secuencial del archivo. Por tanto, un índice secundario debe contener punteros a todos los registros. Se puede usar un nivel adicional de indirección para implementar los índices secundarios sobre claves de búsqueda que no sean claves candidatas. Los punteros en estos índices secundarios no apuntan directamente al archivo. En vez de eso, cada puntero apunta a un cajón que contiene punteros al archivo. Se muestra la estructura del archivo cuenta, con un índice secundario que emplea un nivel de indirección adicional, y teniendo como clave de búsqueda el saldo

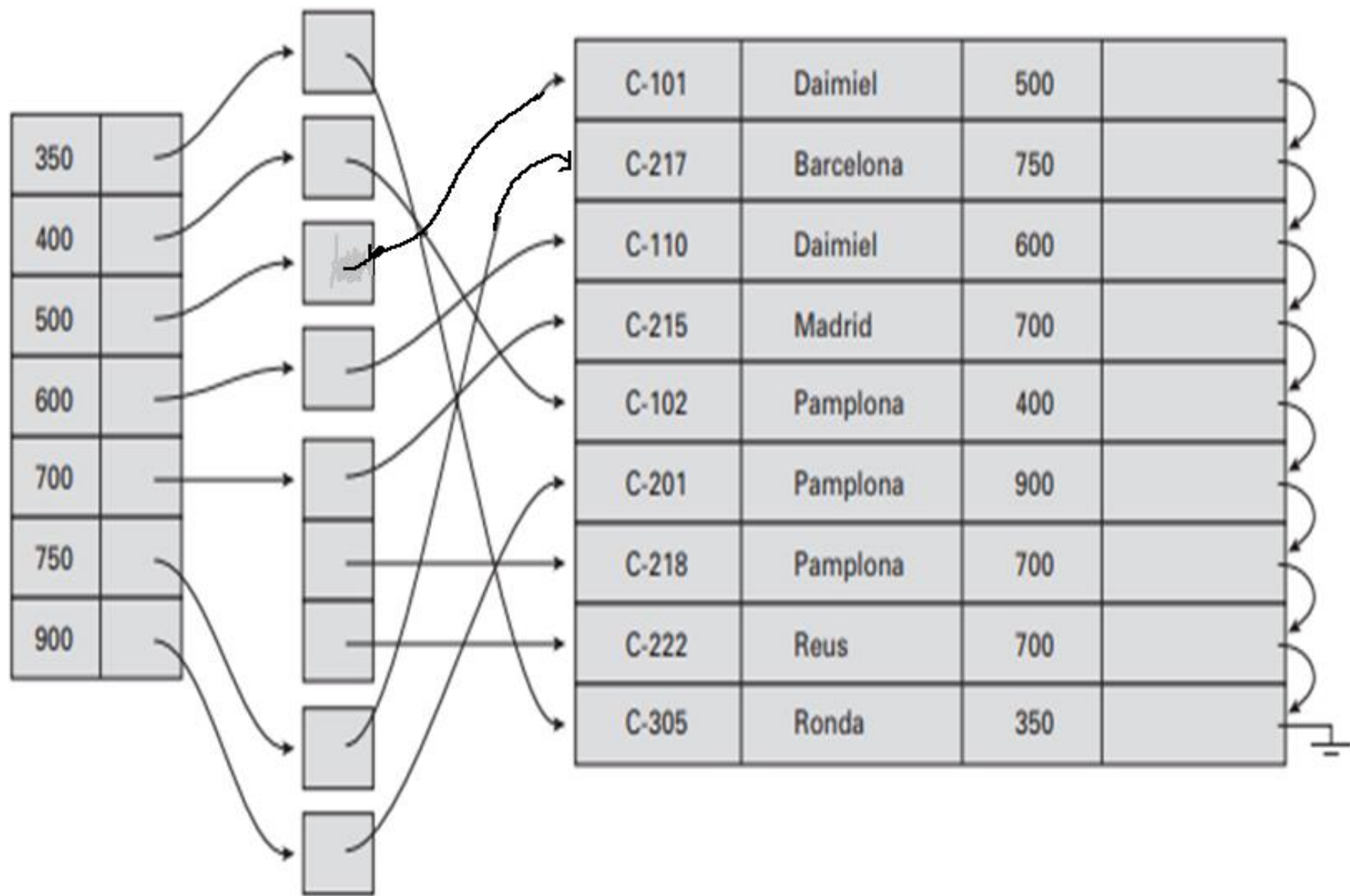


FIGURA 12.5. Índice secundario del archivo *cuenta*, con la clave no candidata *saldo*.