

Tema 6: Organización Secuencial

-
- 6.1 Introducción
 - 6.2 Estructura de la organización secuencial
 - 6.3 Ocupación del fichero secuencial
 - 6.4 Operaciones con los ficheros secuenciales
 - 6.5 Resumen
-

Contenidos extraídos del libro:

Ficheros. Organizaciones clásicas para el almacenamiento de la información

Autores: *Irene Luque Ruiz, Juan Antonio Romero del Castillo y Miguel Ángel Gómez-Nieto*

Editorial: Servicio de Publicaciones de la Universidad de Córdoba, 1998.

ISBN 84-7801-468-3

6.1 Introducción

El fichero secuencial:

- Registros de estructura perfectamente definida
- Registros de longitud fija o variable
- Concepto de **clave** del registro

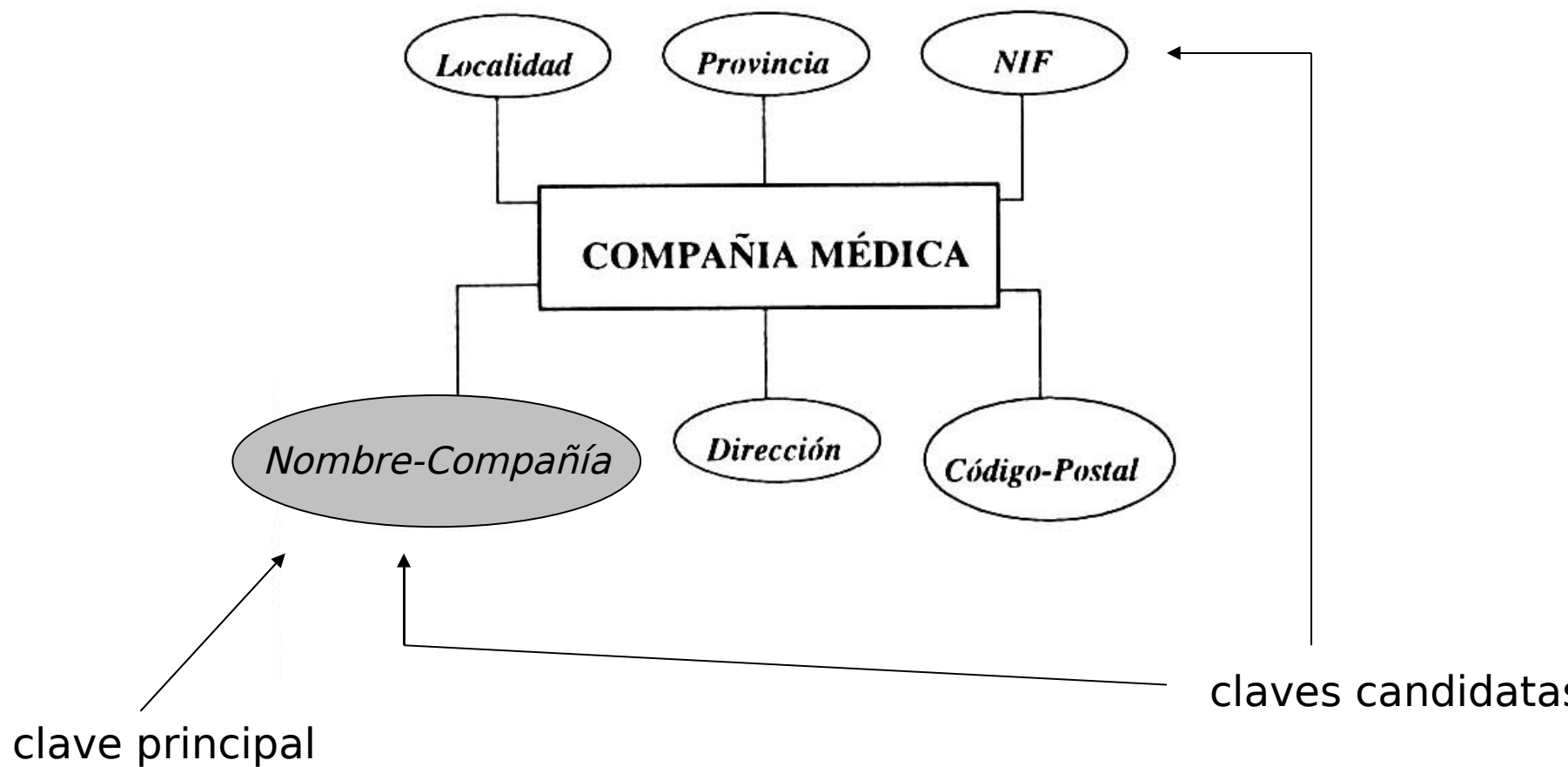
6.1 Introducción

Ejemplo: Gestión de compañías médicas

En el sistema del Laboratorio de Análisis Clínicos, se desea mantener información de las Compañías Médicas a las cuales puede estar afiliado cada uno de los pacientes que acude al laboratorio. Además, se desea que el sistema pueda mantener información referente a las Compañías Médicas, la facturación que se les realiza a las mismas, número de pacientes de una determinada compañía que visita el Laboratorio en un tiempo determinado, etc.

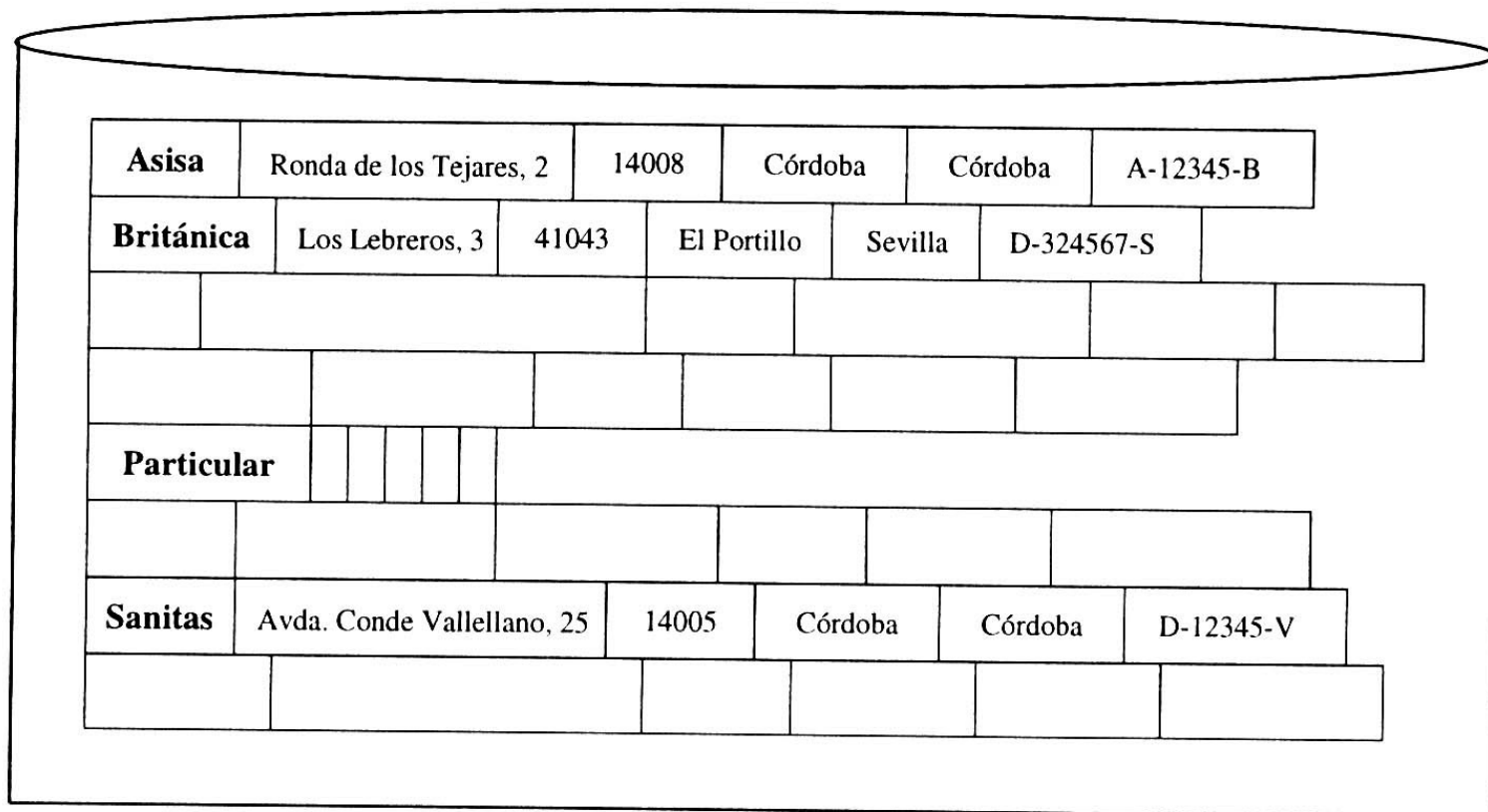
6.1 Introducción

Ejemplo: Gestión de compañías médicas



6.2 Estructura de la Organización Secuencial

Fichero secuencial con registros de longitud variable y clave interna



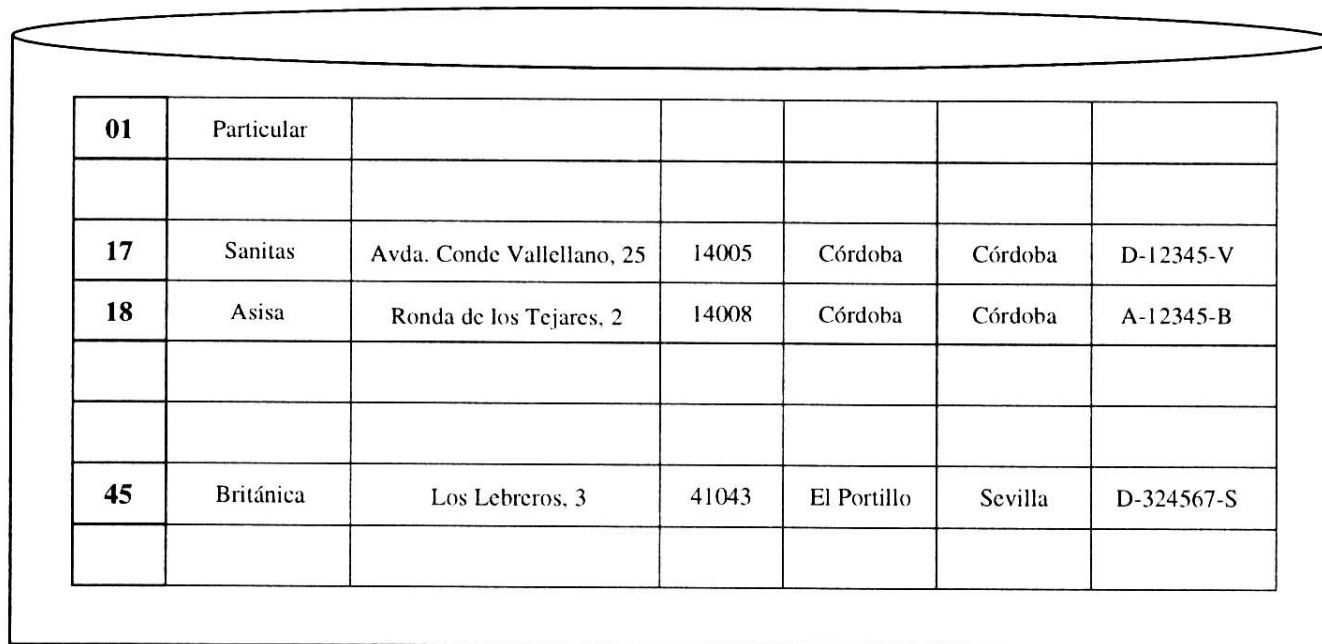
6.2 Estructura de la Organización Secuencial

La clave interna *Nombre-Compañía* supone:

- El dato será necesario en todos los accesos del usuario. De forma exacta o con ayudas de búsqueda, etc.
- Será el dato que se guarde en otros ficheros relacionados con éste.

6.2 Estructura de la Organización Secuencial

Fichero secuencial con registros de longitud fija y clave externa



01	Particular					
17	Sanitas	Avda. Conde Vallellano, 25	14005	Córdoba	Córdoba	D-12345-V
18	Asisa	Ronda de los Tejares, 2	14008	Córdoba	Córdoba	A-12345-B
45	Británica	Los Lebreros, 3	41043	El Portillo	Sevilla	D-324567-S

También llamados: **ficheros relativos**

6.3 Ocupación

Fichero Secuencial (igual que apilo estructurado)

longitud variable, tamaño medio de registro

$$\bar{R} = \sum_{j=1}^k (L_{V_j} + L_S)$$

longitud variable, ocupación del fichero

$$\Gamma = \sum_{i=1}^r \sum_{j=1}^k (L_{V_{ij}} + L_{S_j})$$

longitud fija, tamaño de registro

$$R = \sum_{j=1}^k L_{V_j}$$

longitud fija, ocupación del fichero

$$\Gamma = R \times r$$

6.4 Operaciones

Inserción en la ZONA MAESTRA: T_I

- 1) Localizar posición de inserción del nuevo registro.
- 2) Desplazamiento del resto de registros una posición hacia al final.

$$T_I = T_L + \frac{1}{r} b (T_{tB} + T_{RE})$$

T_L : tiempo de lectura

T_{tB} : tiempo de transferencia de un bloque

T_{RE} : tiempo de reescritura de un bloque

! Muy Costoso !

6.4 Operaciones

Inserción en ZONA DE DERRAMA: T_I

- Para aminorar los costes de la operación anterior, se dispone un archivo auxiliar apilo estructurado (ZONA DE DERRAMA)
- Ahora las inserciones son **diferidas**
- El tiempo de inserción ahora es igual que el del apilo estructurado:

$$T_I = t_l + t_r + T_{tB} + T_{RE}$$

! Más eficiente ! pero a costa de un costoso mantenimiento de una zona de DERRAMA que deteriora la organización como veremos a continuación

6.4 Operaciones

Inserción en ficheros secuenciales RELATIVOS: T_I

$$T_I = t_l + t_r + T_{tB} + T_{RE}$$



En los ficheros relativos la inserción es SIEMPRE tan eficiente como en la apilo, pero la clave está restringida a la posición relativa del registro en el fichero.

$$\textit{Posición del primer byte} = R \times (C_{V_r} - 1)$$

Inconveniente ficheros relativos:
desperdicio de espacio si las claves están dispersas

6.4 Operaciones

Lectura: T_L

- Siempre en base a un criterio de búsqueda
- Si el criterio no es en base a la clave la lectura tarda igual que en la organización apilo:

$$\bar{T}_L = \frac{1}{2} b T_{tB} = \frac{1}{2} r T_{tR}$$

6.4 Operaciones

Lectura: T_L

Si la búsqueda es en base a la clave se aplica la búsqueda binaria o dicotómica:

1. Se lee como primer bloque el bloque central del fichero
2. Se analizan las claves de sus registros con la clave del registro buscado
3. Si son mayores se consideran los $b/2$ bloques siguientes (si son menores se consideran los $b/2$ bloques anteriores) y se repite el proceso.

$$\bar{T}_L = T_{L_0} + \log_2 b (t_r + T_{tB} + T_P)$$

T_{L_0} : tiempo lectura del primer bloque, ya que este siempre cuesta más

T_P : tiempo de comparación de las claves de cada bloque leído

6.4 Operaciones

Lectura: T_L

- Si existe zona de DERRAMA y la búsqueda no ha tenido éxito en la zona maestra, es necesario una búsqueda **adicional** del registro en la zona de derrama.
- Al ser la zona de derrama APILO, su tiempo de lectura será igual que la de un fichero apilo:

$$\bar{T}_L = \frac{1}{2} b_D T_{tB} = \frac{1}{2} r_D T_{tR}$$

b_D : número de bloques de la zona de derrama

r_D : número de registros de la zona de derrama

6.4 Operaciones

Lectura en ficheros RELATIVOS: T_L

- Si el criterio de búsqueda es la clave relativa es muy rápido:

$$T_L = t_l + t_r + T_{tB} + T_P$$

- Si es otro el criterio de búsqueda la lectura es como en un apilo:

$$\bar{T}_L = \frac{1}{2} b T_{tB} = \frac{1}{2} r T_{tR}$$

6.4 Operaciones

Lectura Consecutiva: T_{LC}

Si se realiza por cualquier atributo diferente a la clave:

$$\bar{T}_{LC} = T_L = \frac{1}{2} b T_{tB} = \frac{1}{2} r T_{tR}$$

Si es en base a la clave:

- El registro estará en el mismo bloque que el registro anteriormente leído con lo que $T_{LC} = 0$
- El registro estará en el siguiente bloque con una probabilidad de $1/F_B$ y $T_{LC} = T_{tB}$

$$\bar{T}_{LC} = \frac{T_{tB}}{F_B} = T_{tR}$$

! Muy eficiente !

6.4 Operaciones

Lectura Consecutiva: T_{LC}

Si hay que buscar en la zona de DERRAMA la eficiencia de la operación anterior se pierde en la medida que aumenta dicha zona de derrama

Lectura Consecutiva en ficheros RELATIVOS:

No hay zona de derrama, luego siempre será igual de eficiente que la secuencial sin derrama

$$\bar{T}_{LC} = \frac{T_{tB}}{F_B} = T_{tR}$$

! Muy eficiente !

6.4 Operaciones

Actualización: T_A

- La actualización de un registro suele considerarse igual que el borrado de un registro.
- Marcar para borrado simplifica el operativo y permite la recuperación de la información.
- Aunque genera espacio desperdiciado hasta la siguiente reorganización del fichero.

Proceso cuando cambia la clave:

1. Lectura del registro que se desea modificar.
2. Reescritura de ese registro marcado para borrado.
3. Inserción del nuevo registro.

$$T_A = \bar{T}_L + T_{RE} + T_I + T_P$$

6.4 Operaciones

Proceso cuando NO cambia la clave: T_A

1. Lectura del registro que se desea modificar
2. Modificación del registro
3. Reescritura del registro

$$T_A = \bar{T}_L + T_{RE} + T_P$$

6.4 Operaciones

Lectura Exhaustiva (total): T_{LT}

Supone leer sistemáticamente todos los datos sin importar el orden

$$T_{LT} = (b + b_D) T_{tB} = (r + r_D) T_{tR}$$

6.4 Operaciones

Lectura Ordenada: T_{LO}

- Leer todos los datos en orden
- En este caso ordenados por la clave del fichero secuencial
- Basamos esta operación en la Lectura Consecutiva

$$T_{LO} = T_L + (r - 1)T_{LC}$$

Si NO EXISTE zona de DERRAMA

$$T_{LO} = T_{LT} = bT_{tB}$$

**! Muy eficiente !
La mejor lectura ordenada
de todas las organizaciones de ficheros**

6.4 Operaciones

Si EXISTE zona de DERRAMA: T_{LO}

Primero reorganizamos y luego leemos de forma exhaustiva:

$$T_{LO} = T_{RO} + T_{LT}(r + r_D)$$

T_{RO} : tiempo de reorganización del fichero

O bien se puede hacer: se ordena la zona de derrama y se leen ambos archivos en paralelo:

$$T_{LO} = T_C(r_D) + T_{LT}(r + r_D) + T_P$$

T_C : tiempo de ordenación

6.4 Operaciones

Si es por un atributo distinto a la clave: T_{LO}

Primero ordenamos todos los registros por el nuevo atributo y luego leemos de forma exhaustiva el fichero resultante:

$$T_{LO} = T_C(r + r_D) + T_{LT}(r + r_D)$$

6.4 Operaciones

Reorganización: T_{RO}

- Las operaciones de actualización y borrado generan registros marcados para borrado
- Volatilidad de los datos: cuando los datos almacenados dejan de ser útiles debido a una operación (borrado, actualización, etc.)
- La reorganización debe restituir el fichero a su estado óptimo:
 - ❑ Liberación de espacio no utilizado de los registros marcados para borrado
 - ❑ Eliminación de la zona de derrama e integración de sus registros en la zona secuencial

6.4 Operaciones

Reorganización:

El proceso incluye: T_{RO}

1. Eliminar registros no significantes
2. La zona de derrama debe quedar vacía
3. Todos los registros quedan ordenados: ordenar zona de derrama y leer en paralelo junto con la secuencial y crear un único fichero con todos los registros ordenados
4. Borrar ficheros originales y renombrar el nuevo

$$T_{RO} = T_C(r_D) + (r + r_D)T_{tR} + (r + r_D - r_B)T_{tR}$$

r : número de registros en la zona secuencial

r_D : número de registros en la zona de derrama

r_B : número de registros marcados para borrado

6.5 Resumen

- Solucionan algunos inconvenientes de los ficheros apilo
- Accesos más efectivos debido a la clave
- Los ficheros relativos permiten un acceso muy eficiente a costa de su rigidez lógica
- La lectura ordenada es la más eficiente de todas las organizaciones cuando no hay zona de derrama
- Esta organización se deteriora con el tiempo y requiere constantes reorganizaciones

Fin