Organizaciones clásicas de ficheros

Organización Encadenada

Introducción

- Para almacenar grades cantidades de información es necesario repartirla entre varios ficheros.
- Tendremos que relacionar la información entre los diferentes ficheros para:
  - Impedir redundancia.
  - Asegurar la consistencia.
- **Entidades:** Modelan conceptos de la vida real (cliente, cuenta, proveedor, producto, ...)
  - Cada entidad se almacenará en un fichero.
- **Relaciones:** Modelan la relación entre entidades (Cliente con cuenta. Proveedor con producto).
  - Dependiendo del tipo de relación también se almacenarán en ficheros.
- **Punteros:** Identifica de forma unívoca una instancia de una entidad.
  - Utilizan el campo clave (interno o externo).
  - Sirven para expresar las relaciones (cliente 30791 con cuenta 201).
- Tipos de punteros:
  - Dirección física de la máquina (cilindro, cabeza, sector).
  - Dirección relativa al principio del fichero.
  - Valor simbólico que identifica al registro (clave interna o externa).

Estructura encadenada

- Fichero encadenado: Al menos un campo de los registros es un puntero a otros registros.
- Los punteros podrán ser:
  - Internos: referencia a otro registro dentro del mismo fichero, estableciendo un orden. Podemos tener más de uno para varios ordenes.
  - Externos: referencia a otro registro en otro fichero.
- Cadena: lista ordenada enlazada (usando punteros) de registros. Esta lista tendrá cabeza y cola.
- Ventajas respecto a org. secuencial:
  - El orden no está sujeto a una secuencia ordenada de registros.
  - Podemos tener varios órdenes (una cadena por orden).

Ejemplo

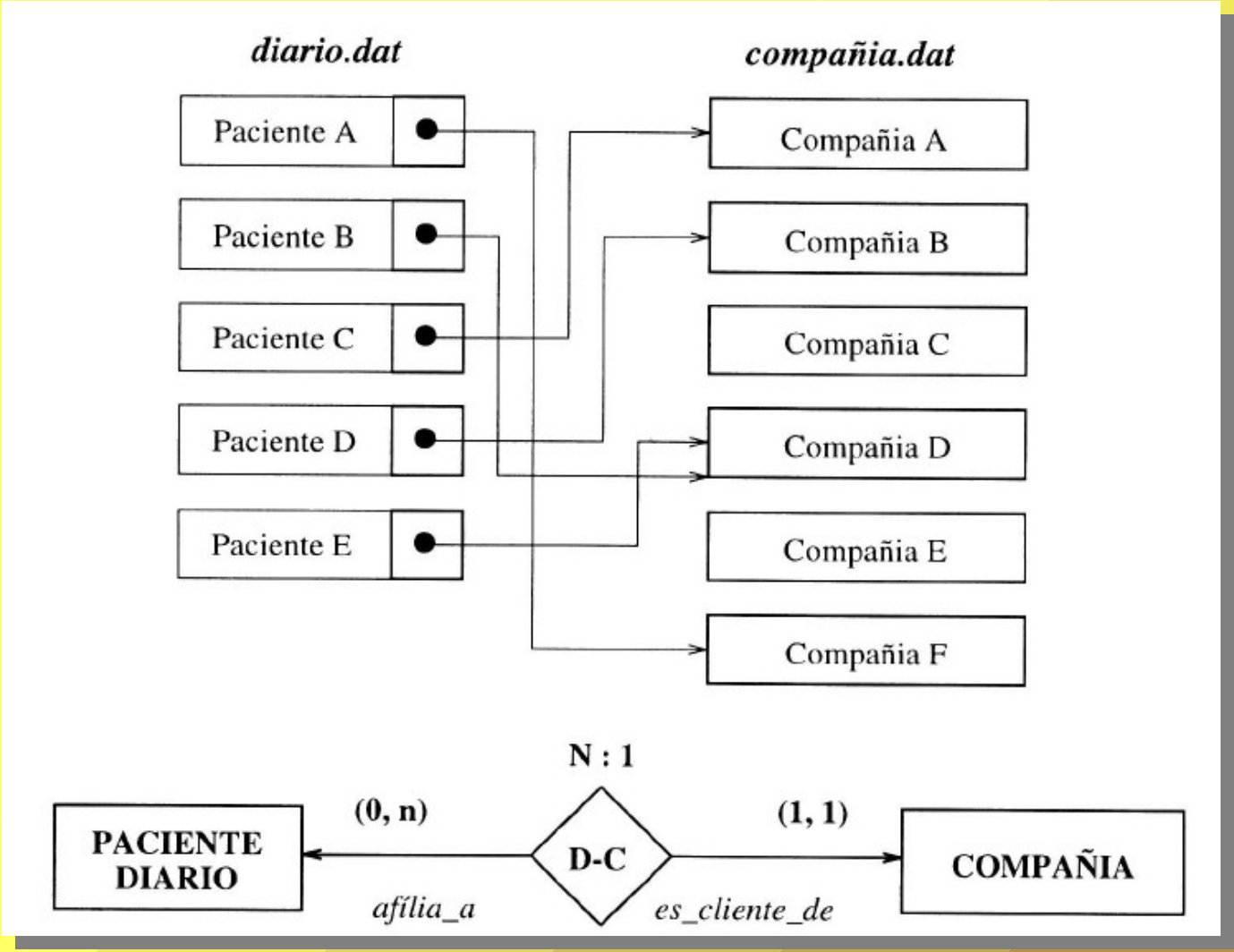
Los pacientes diarios:

*En el sistema del Laboratorio de Análisis Clínicos, se desea mantener información sobre los **pacientes** que acceden diariamente al **Laboratorio**. La primera vez que un paciente acude al Laboratorio se le abre una ficha con sus datos personales (aquellos que se consideran que no son dependientes con los análisis que se van a realizar) y cada vez que acude se le da de alta en el archivo denominado **diario.dat** en el que se desea almacenar: la identificación del paciente, el doctor que prescribe el conjunto de análisis y la compañía médica a la cual se le facturará el coste del análisis.*

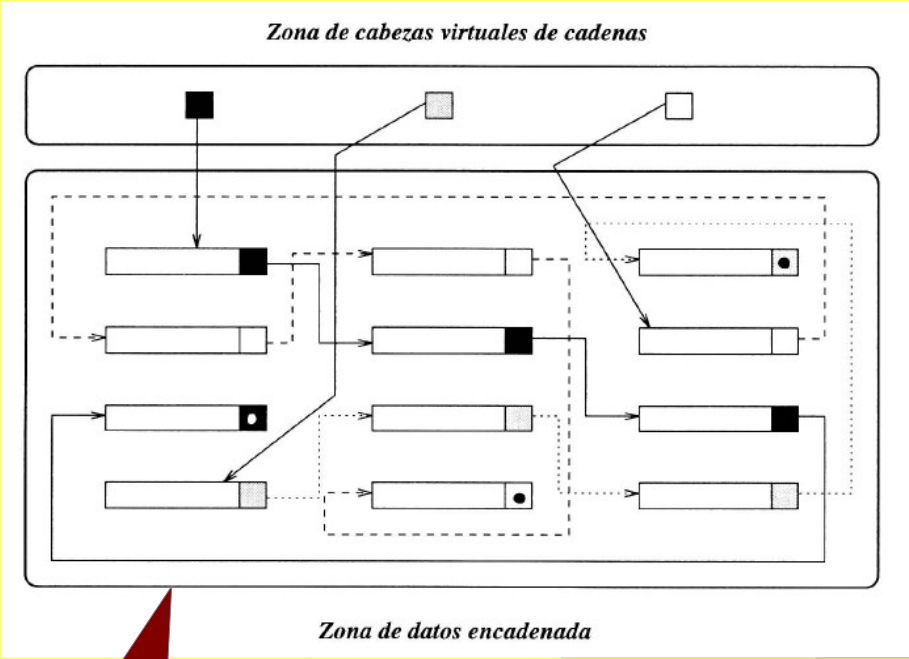
Ejemplo

- Solución 1:
  - En diario.dat almacenar los datos del paciente y de la compañía.
    - Inconvenientes: Redundancia y posible incoherencia.
- Solución 2:
  - Dos ficheros relacionados: diario.dat para pacientes y compañía.dat para las compañías.
    - Ventajas: elimina redundancia e incoherencia.

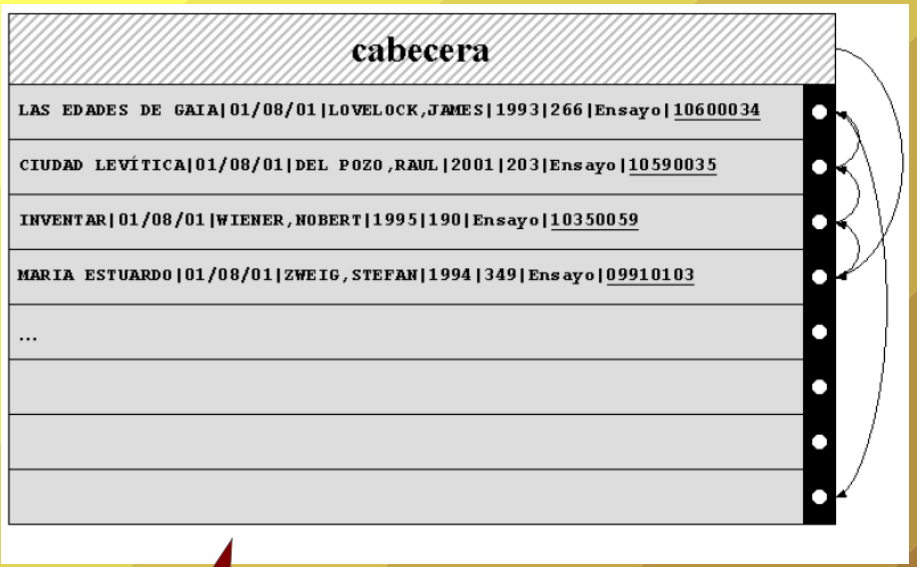
Ejemplo


diario.dat
compañia.dat
Paciente A
Paciente B
Paciente C
Paciente D
Paciente E
Compañia A
Compañia B
Compañia C
Compañia D
Compañia E
Compañia F
N : 1
(0, n)
(1, 1)
PACIENTE DIARIO
D-C
COMPAÑIA
afília_a
es_cliente_de

Estructura



Estructura lógica



Estructura física

Operaciones

- Inserción O(N):
  - Insertar en cualquier posición libre o añadir al final (puede existir una lista de bloques libres).
  - Reasignar punteros para mantener los ordenes en las cadenas disponibles O(N).
- Localización O(N):
  - Rápida para secuencias en orden.
  - Similar a apilo si es fuera de orden.
- Lectura consecutiva en orden O(1) o fuera de orden O(N).
- Lectura exhaustiva O(N).
- Lectura ordenada O(N) o $O(N^2)$ si fuera de orden.

Operaciones

- Actualización O(1) o O(N):
  - Manteniendo orden O(1).
  - Rompiendo orden O(N) (puede ser mejor borrar e insertar nuevo registro).
- Borrado O(1):
  - Marcar y reasignar punteros.
  - Si hay varias cadenas, tener varias marcas de borrado según cadena.

Operaciones

- Reorganización O(N):
  - Periódicamente para eliminar registros que están borrados en todas las cadenas.
  - Alternativas:
    - Una sola cadena: copiar en orden en un fich. auxiliar secuencial los no marcados como borrados. El puntero apunta al siguiente registro.
    - Varias cadenas $O(N^2)$:
      - Tratar la cadena principal como antes.
      - Para cada cadena secundaria: generar un fichero secuencial auxiliar según ese orden con un puntero con la posición del reg. en el fichero principal.
      - Actualizar el puntero secundario en el fichero principal siguiendo el orden en el fichero auxiliar.

Resumen

- Ventajas:
  - Permite definir varios órdenes internos.
  - Permite representar estructuras de información complejas entre distintos tipos de entidades.
  - Reduce el coste de mantener información heterogénea relacionada.
- Inconvenientes:
  - La gestión de las cadenas es delicada propensa a errores. Un sólo fallo en un registro puede deteriorar el fichero completo.
  - Elevado coste de reorganización cuando hay más de un orden.

Referencias

- Luque Ruiz I. y otros, “Ficheros. Organizaciones clásicas para el almacenamiento del a información”, U. de Córdoba, 1998.
- A. Aliaga, apuntes de la asignatura “Organización y Gestión de Archivos”. U. de Almería. http://www.ual.es/personal/analiaga