

# Tema I.- Diseño Físico

Ficheros

Índices

*Fundamentos de sistemas de bases de datos.* Elmasri, R; Navathe, S. Addison-Wesley  
[cap.13 en 5ª edic.]



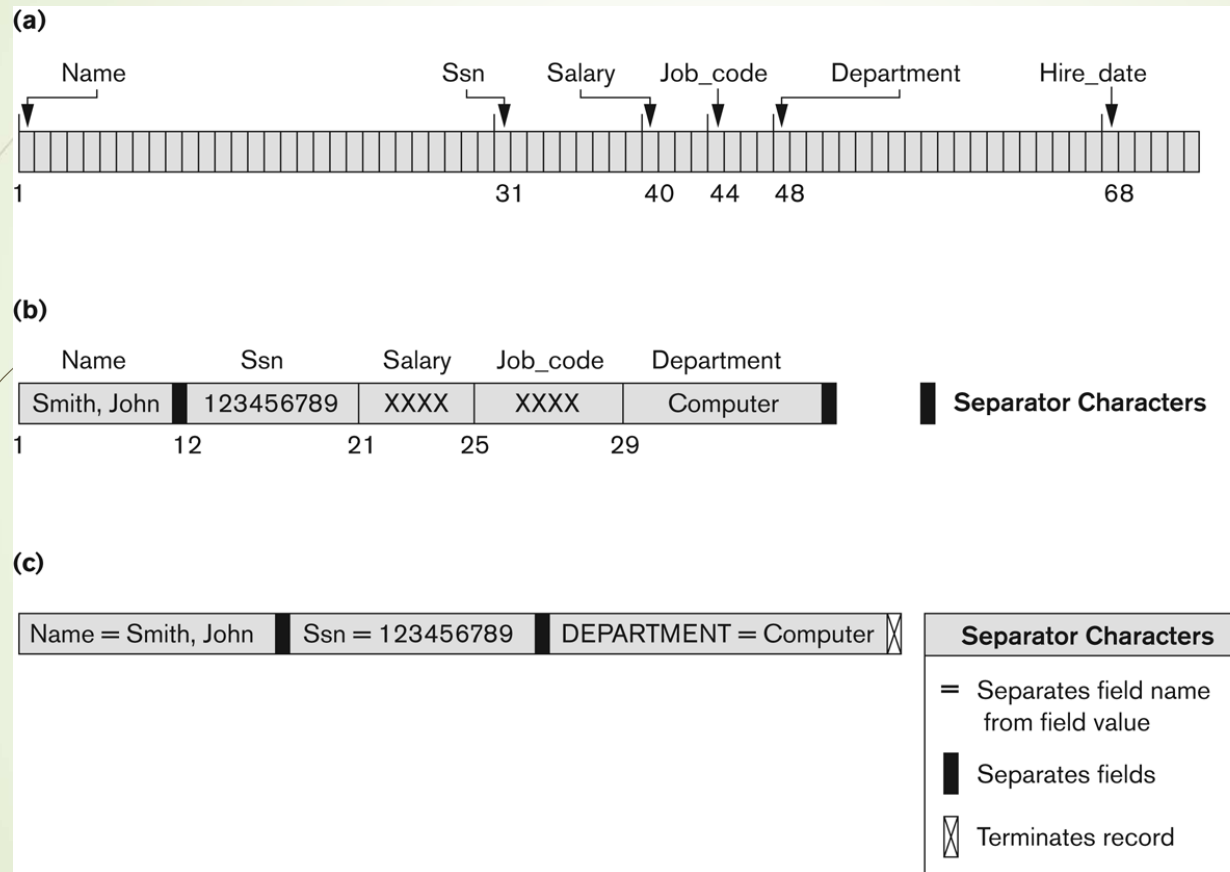
# Índice

- **Registros / Ficheros**
- **Operaciones sobre ficheros**
- **Ficheros de registros desordenados (heap)**
- **Ficheros de registros ordenados**
- **Técnicas de dispersión (fichero hash)**
  - Dispersión interna (memoria)
  - Dispersión externa (disco)
  - Técnicas de dispersión que permiten la expansión dinámica del fichero

# Registros /Ficheros

- **Registro** = {campos con valores dentro de un dominio determinado}
  - Ej: fecha, hora, edad, nombre
- **Fichero** = secuencia de registros de longitud **fija** (= tamaño) o **variable** (≠ tamaño)
- Un fichero puede tener registros de **longitud variable** porque:
  1. Los registros tienen **campos de longitud variable**
    - Separador al finalizar cada campo de longitud variable, o indicación previa de su longitud
  2. Los registros tienen **campos repetitivos**
    - Separador entre valores repetitivos + Separador al terminar el campo
  3. Los registros tienen **campos opcionales**
    - Opción 1: <nombre\_campo, valor>
    - Opción 2: <tipo\_campo, valor> (p.ej, <1, valor\_campo1> <2, valor\_campo2> ...)
  4. Contiene **diferentes tipos de registro** (fichero mixto)
    - Cada registro va precedido de un indicador de tipo de registro

# Registros /Ficheros



a) Registro de longitud fija

b) Registro con 2 campos de longitud variable y 3 de longitud fija

c) Registro con campos opcionales con 3 tipos de separadores

**Figure 13.5**

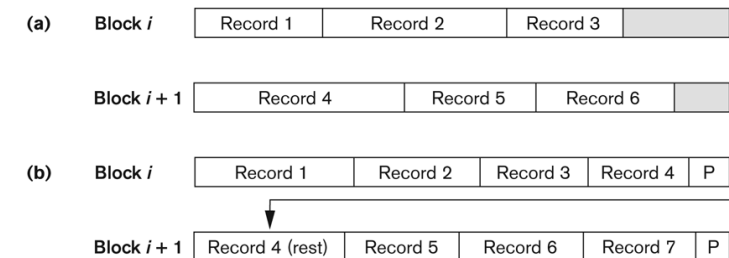
Three record storage formats. (a) A fixed-length record with six fields and size of 71 bytes. (b) A record with two variable-length fields and three fixed-length fields. (c) A variable-field record with three types of separator characters.

# Registros /Ficheros

- Los registros se asignan a bloques de disco (unidad mínima de transferencia)
- El **factor de bloqueo bfr** es el nº medio de registros que se pueden almacenar en 1 bloque.
- Los registros pueden almacenarse en formato **extendido** o **no extendido**
  - Organización No Extendida: los registros no pueden sobrepasar los límites de un bloque.
    - El factor de bloqueo se calcula como  $bfr = \lfloor B/R \rfloor$ , siendo B el tamaño del bloque y R el tamaño de los registros
    - El nº de bloques  $b$  necesario para un fichero de  $r$  registros es  $b = \lceil r/bfr \rceil$
  - Organización Extendida: los registros pueden abarcar más de un bloque.
    - El factor de bloqueo se calcula como  $bfr = (B-P)/R$  siendo B el tamaño del bloque, P el tamaño del puntero al siguiente bloque y R el tamaño de los registros
    - El nº de bloques  $b$  necesario para un fichero de  $r$  registros es  $b = \lceil r/bfr \rceil$

**Figure 13.6**

Types of record organization. (a) Unspanned. (b) Spanned.



# Ejercicio

1- Supongamos un fichero con 12,000 registros de longitud fija. Cada registro tiene los siguientes campos y tamaños. Se utiliza 1 byte adicional como marcador de eliminación.

NOMBRE	NSS	DIRECCION	TLF	FECHA NACIM	SEXO	HORARIO	GRUPO	AULA
6 bytes	20 bytes	50 bytes	9 bytes	8 bytes	1 byte	2 bytes	2 bytes	4 bytes

(a) Calcular el tamaño de un registro  $R$  en bytes.

$$\text{tamaño}_R = (6+20+50+9+8+1+2+2+4) + 1 = 103 \text{ bytes}$$

El bit adicional corresponde al marcador de eliminación

(b) Calcular el factor de bloqueo  $bfr$  y nº bloques  $b$  suponiendo una organización no extendida. El tamaño del bloque es 512 bytes

El factor de bloqueo  $bfr$  es el nº de registros por bloque. Si el tamaño del bloque es de  $B$  bytes, en caso de un fichero de registros de longitud fija de  $R$  bytes, podemos encajar

$$bfr = \lfloor B / \text{tamaño}_R \rfloor = \lfloor 512 / 103 \rfloor = 4 \text{ registros por bloque}$$

El número de bloques  $b$  necesario para un fichero de  $r$  registros, si la organización no es extendida (i.e., los registros no pueden sobrepasar los límites de un bloque), se calcula como:

$$b = \lceil r / bfr \rceil = \lceil 12000 / 4 \rceil = 3000 \text{ bloques}$$

(c) ¿Cuál sería el número de bloques del fichero  $b$  suponiendo una organización extendida, y un tamaño de puntero de 2 bytes?

En este caso los registros pueden sobrepasar los límites de un bloque, y se calcula como:

$$bfr = (B - P) / \text{tamaño}_R = (512-2) / 103 = 4,95 \text{ registros por bloque}$$

$$b = \lceil r / bfr \rceil = \lceil 12000 / 4,95 \rceil = 2425 \text{ bloques}$$

# Ejercicio

2- El 75% de los registros tienen un valor en el campo TLF, el 60% en HORARIO, y el 85% en AULA, y se usa un fichero de registros de longitud variable.

Cada registro tiene un *tipo de campo* (1 byte) por cada campo incluido en el registro, más el marcador de eliminación de registro (1 byte) y un marcador de fin de registro (1 byte). Se utiliza una organización extendida de registros, en la que cada bloque tiene un puntero de 4 bytes al siguiente bloque.

(a) Calcular el tamaño medio de un registro  $R$  en bytes, siendo el tamaño del bloque de 512 bytes.

Asumiendo que cada campo tiene 1 byte para indicar el tipo de campo, y que NOMBRE, NSS, DIRECCION, FECHA NACIM, SEXO y GRUPO tienen valores para todos los registros, se necesita el siguiente número de bytes para esos campos en cada registro, más 1 byte para el marcador de eliminación, y 1 byte para el marcador de fin de registro:

$$\text{tamaño}_{\text{CAMPOS FIJOS}} = (6+1) + (20+1) + (50+1) + (8+1) + (1+1) + (2+1) + 1 + 1 = 95 \text{ bytes}$$

Para los campos TLF, HORARIO y AULA, el número medio de bytes por registro es:

$$\text{tamaño}_{\text{CAMPOS VARIABLES}} = [(9 + 1) * 0.75] + [(2 + 1) * 0.60] + [(4 + 1) * 0.85] = 7.5 + 1.8 + 4.25 = 13.55 \text{ bytes}$$

Por tanto, el tamaño medio del registro es:

$$\text{tamaño}_R = \text{tamaño}_{\text{CAMPOS FIJOS}} + \text{tamaño}_{\text{CAMPOS VARIABLES}} = 95 + 13.55 = 108.55 \text{ bytes}$$

(b) Calcular el número de bloques necesarios para el fichero, siendo el tamaño del bloque de 512 bytes.

$$\text{bfr} = (512-4) / 108.55 = 4.68 \text{ registros por bloque}$$

$$b = \lceil 12000 / 4.68 \rceil = 2565 \text{ bloques}$$

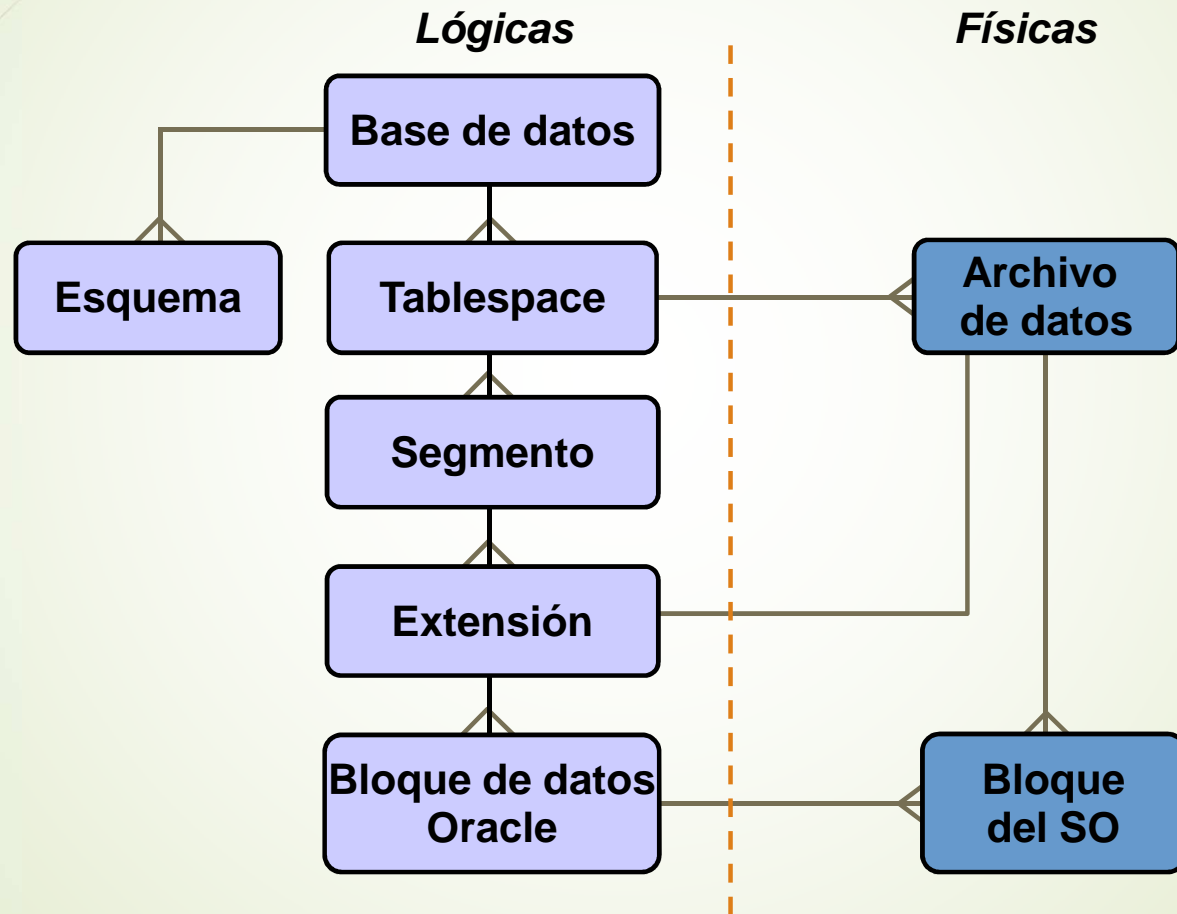


# Registros /Ficheros

- Hay varias técnicas para asignar los bloques de un fichero de disco:
  - Asignación **contigua**: bloques consecutivos
  - Asignación **enlazada**: cada bloque tiene un puntero al siguiente
  - Asignación de **extensiones** (bloques de discos consecutivos) que se enlazan con punteros
  - Asignación **indexada**: bloques de índice apuntan a los bloques del fichero
- Oracle???

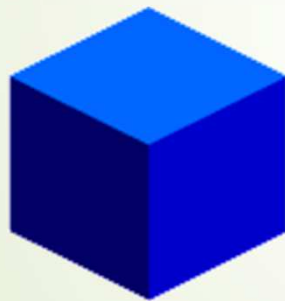


# Estructuras lógicas y físicas de Oracle



# Segmentos, Extensiones y Bloques en Oracle

- ▶ Los segmentos existen en un tablespace
- ▶ Cada segmento está formado por el conjunto de extensiones de una tabla
- ▶ Las extensiones son un conjunto de bloques de datos
- ▶ Los bloques de datos están asignados a bloques del SO



**Segmento**



**Extensiones**

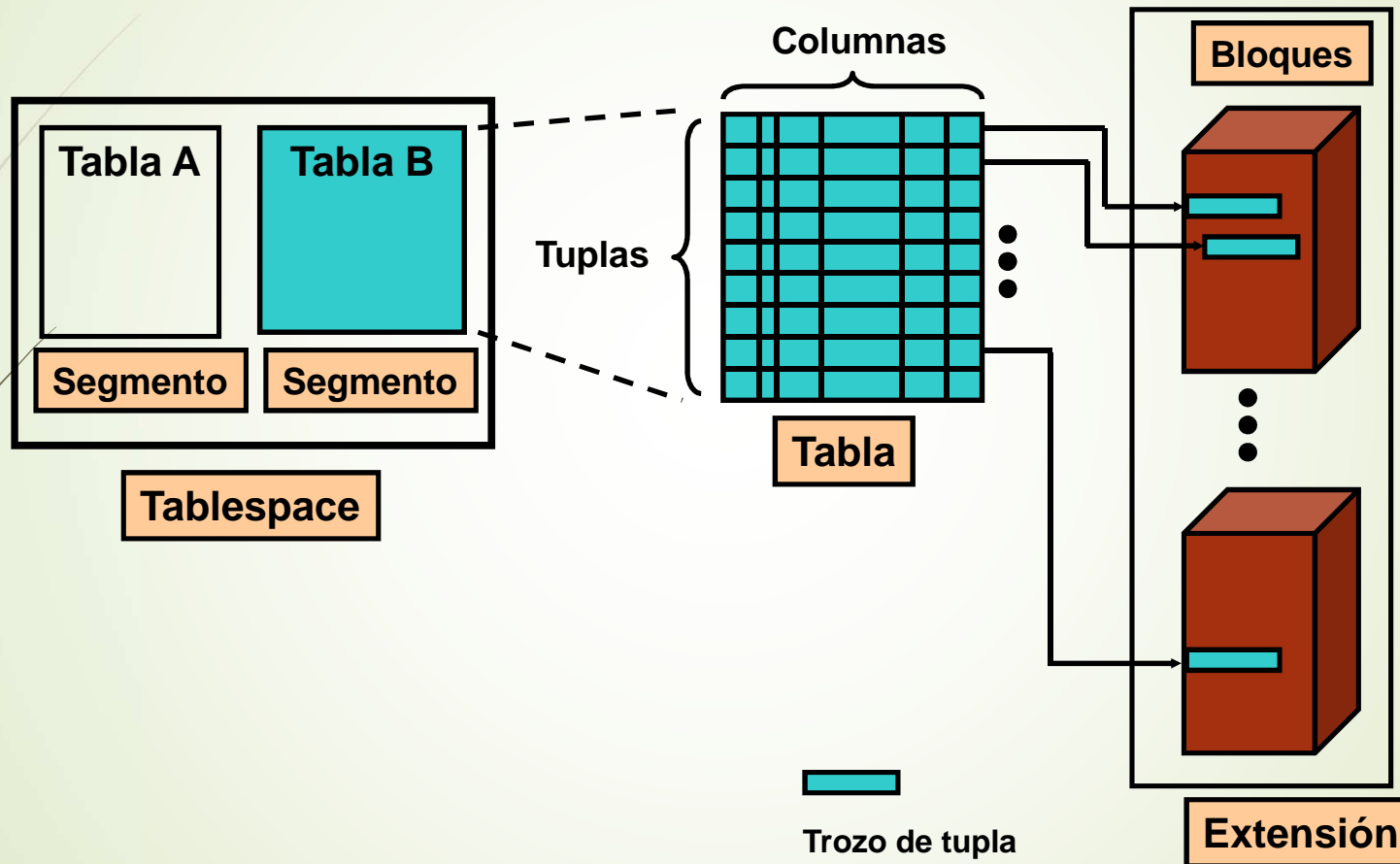


**Bloques  
de datos**

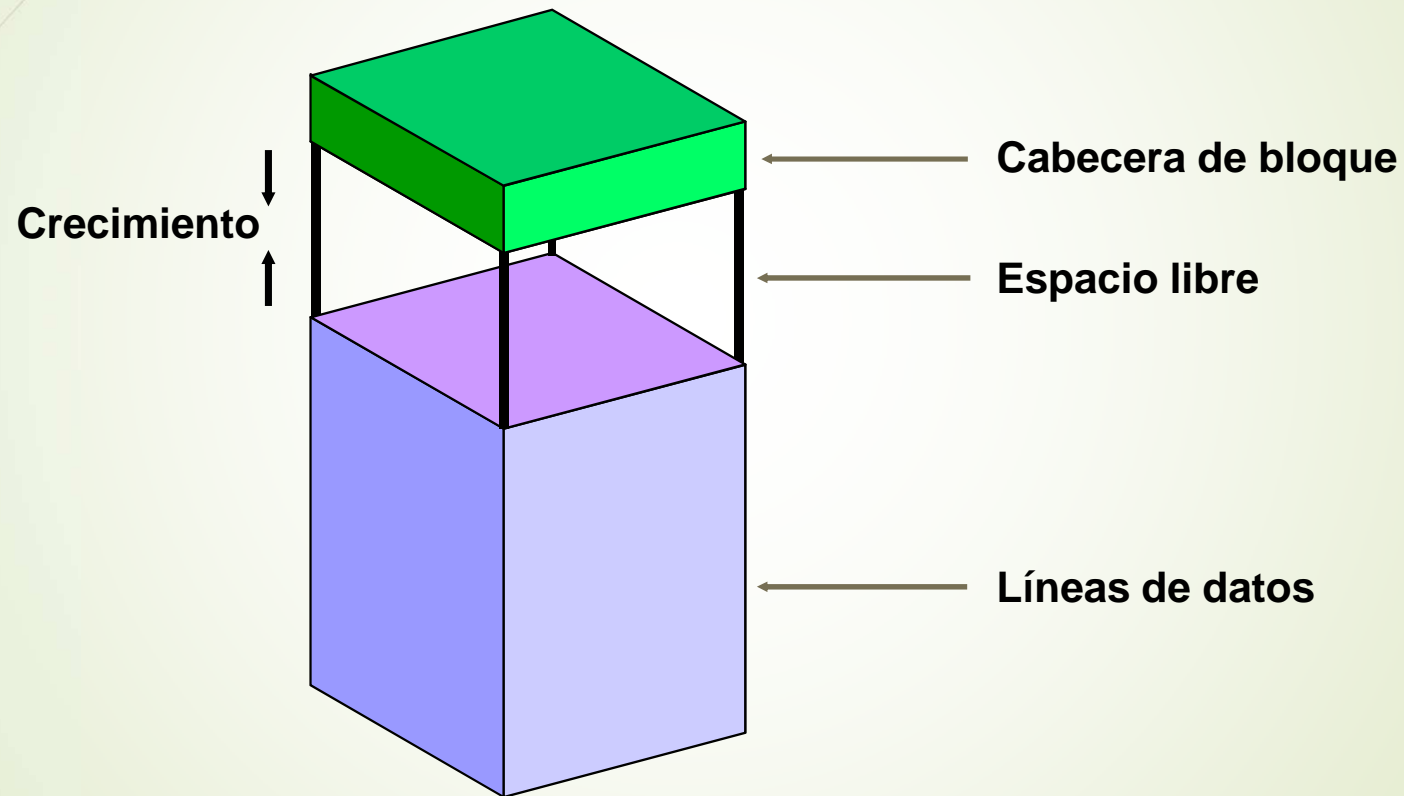


**Bloques  
de disco**

# Cómo se almacena una tabla en Oracle



# Contenido de un bloque de datos Oracle





# Operaciones con ficheros

- **Open:** Prepara el fichero para la lectura o escritura y apunta al principio del fichero
- **Reset:** Apunta al principio del fichero
- **Find:** Localiza el primer registro que satisface una condición de búsqueda y lo apunta
- **FindNext:** Localiza el siguiente registro que satisface una condición de búsqueda y lo apunta
- **FindAll:** Recupera todos los registros que satisfacen una condición de búsqueda
- **FindOrdered:** Recupera todos los registros del fichero en un orden determinado
- **Read:** Copia el registro actual a una variable de programa
- **Delete:** Marca el registro actual como borrado
- **Modify:** Cambiar el valor de algunos campos del registro actual
- **Insert:** Inserta un nuevo registro en el fichero y lo apunta
- **Close:** Finaliza el acceso al fichero
- **Reorganize:** Reorganiza el fichero (por un orden específico, eliminando físicamente los registros borrados, etc.)

# Ficheros desordenados (heap)

- Los registros se guardan en el mismo orden que se insertan (al final del fichero)
- Inserción
  1. El último bloque del fichero se copia en el búfer (memoria)
  2. Se añade el registro nuevo
  3. Se reescribe el bloque en disco
- Búsqueda
  - Búsqueda lineal, bloque a bloque
  - Por término medio, se requiere buscar  $b/2$  bloques.
- Eliminación
  1. Se busca el bloque mediante búsqueda lineal
  2. Se copia el bloque en el búfer
  3. Se elimina el registro en el búfer
  4. Se rescribe el bloque en disco
    - Otra opción es tener 1 byte o bit como marcador de borrado
- Oracle???



# Ficheros ordenados (secuenciales)

- Los registros están ordenados en función de un campo llamado **campo de ordenación**
  - Si el campo es clave se denomina **clave de ordenación**
- Inserción
  - Es costosa: los registros deben estar físicamente ordenados
  - Habitualmente se utiliza un fichero temporal (**fichero de desbordamiento** (overflow) o de transacciones) desordenado
    - Los nuevos registros se insertan al final
    - Periódicamente se ordenan y se mezclan con el fichero principal
- Búsqueda
  - Búsqueda binaria
  - Por término medio, se requiere buscar  $\log_2$  bloques
- La lectura de los registros en orden del campo de ordenación es eficiente

# Ficheros ordenados (secuenciales)

**Figure 16.7**

Some blocks of an ordered (sequential) file of EMPLOYEE records with Name as the ordering key field.

	Name	Ssn	Birth_date	Job	Salary	Sex
Block 1	Aaron, Ed					
	Abbott, Diane					
	⋮					
	Acosta, Marc					
Block 2	Adams, John					
	Adams, Robin					
	⋮					
	Akers, Jan					
Block 3	Alexander, Ed					
	Alfred, Bob					
	⋮					
	Allen, Sam					
Block 4	Allen, Troy					
	Anders, Keith					
	⋮					
	Anderson, Rob					
Block 5	Anderson, Zach					
	Angeli, Joe					
	⋮					
	Archer, Sue					
Block 6	Arnold, Mack					
	Arnold, Steven					
	⋮					
	Atkins, Timothy					
⋮						
Block $n-1$	Wong, James					
	Wood, Donald					
	⋮					
	Woods, Manny					
Block $n$	Wright, Pam					
	Wyatt, Charles					
	⋮					
	Zimmer, Byron					



# Tiempos medios de acceso

**Table 16.2** Average Access Times for a File of  $b$  Blocks under Basic File Organizations

Type of Organization	Access/Search Method	Average Blocks to Access a Specific Record
Heap (unordered)	Sequential scan (linear search)	$b/2$
Ordered	Sequential scan	$b/2$
Ordered	Binary search	$\log_2 b$