

Tema 5: Introducción y Conceptos Básicos

- **Punto de Partida**
- **Estructuras Lógicas vs. Estructuras Físicas**
 - **Consecutividad y Concepto de Cubo**
- **Diseño de Ficheros**
 - **Diseño Lógico**
 - **Diseño Físico del Registro Lógico**
 - **Diseño Físico**
- **Objetivos del Diseño Físico**
- **Selección y Localización Física**

Tema 5.1: Punto de partida



enfoco lógico o externo



enfoco físico o interno





El usuario ve *archivos*,
que son colecciones de *registros*,
que son agregaciones de *datos*



Aplicación
SGBD
SGF

La máquina accede a *ficheros*,
que son secuencias de *bloques*,
que son conjuntos de *bytes*

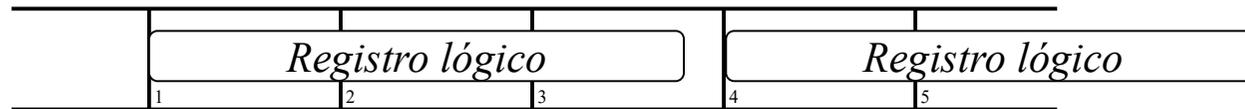


Correspondencia físico-lógica a nivel de registro:

consideración de tamaño: R_l y R_f no suelen coincidir en tamaño

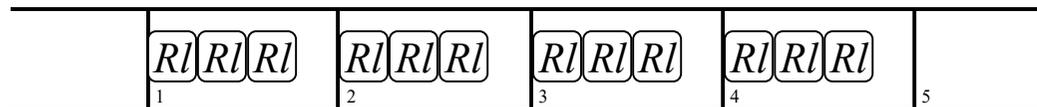
- **Registro Expandido**

Cuando el registro lógico es mayor que el físico (ocupa varios)



- **Bloque**

Cuando el registro físico es mayor que el lógico (cabén varios)



Factor de Bloqueo: número de registros lógicos que caben en uno físico

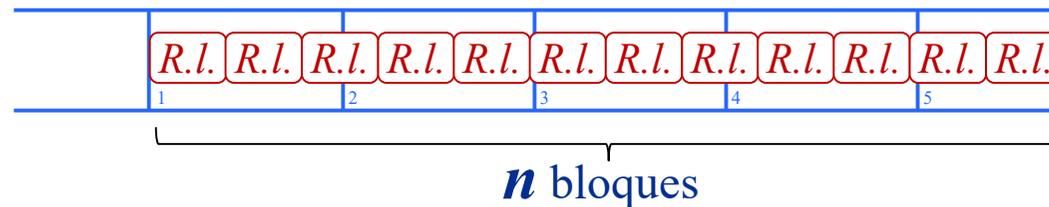
$$f_b = \frac{T_{bq}}{T_{reg}}$$

Correspondencia físico-lógica entre registros:

A este nivel, la organización de registros puede ser de dos tipos:

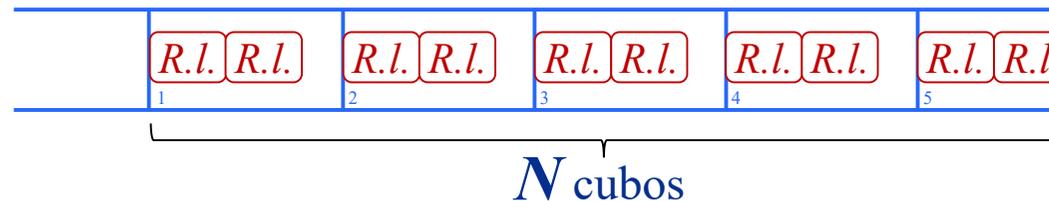
- Organización Consecutiva

Todo registro lógico comienza siempre a continuación del anterior



- Organización No Consecutiva

Todo registro físico comienza con el inicio de un registro lógico



* *Sólo los ficheros seriales y secuenciales pueden tener regs. consecutivos.*

Cubo (DB block): cjto. de bytes con una condición de acceso común (comparten dirección, información de control, organización, espacio libre, ...)

- suele procesarse como una unidad: es como un *'bloque virtual'*
- dentro del cubo, la organización de los registros es consecutiva
- entre cubos, la organización es no consecutiva

Espacio de cubo (E_c): cantidad de información asignada al cubo (habitualmente se mide en bloques, pero puede darse en bytes, KB,...)

- es un múltiplo del tamaño de bloque
- si el espacio es menor de un bloque, se denomina *'celda'*
- la memoria intermedia suele paginarse al espacio de cubo

Tamaño de cubo (T_c):

cuántos registros lógicos contiene el cubo

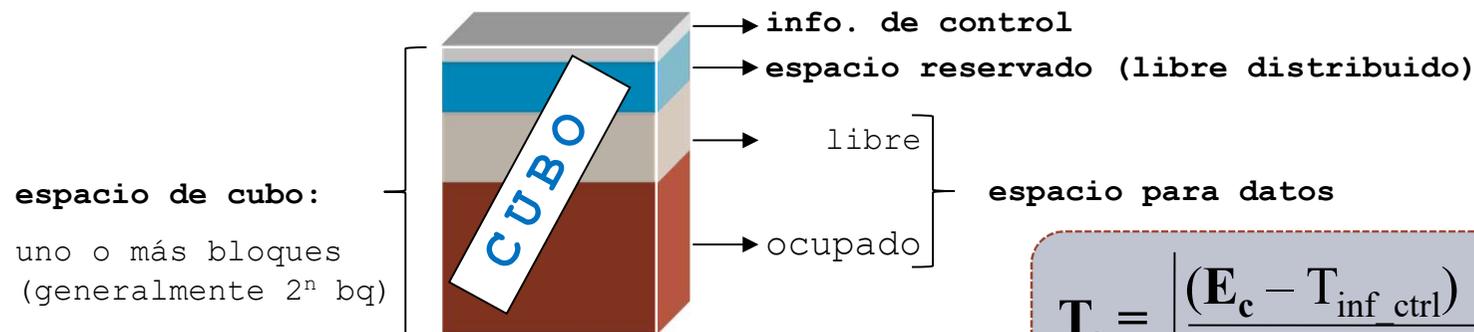
(como el factor de bloqueo, pero registros enteros en cubos)

$$T_c = \left\lfloor \frac{E_c}{T_{reg}} \right\rfloor$$

Tema 5.2: Concepto de Cubo

Cubos y Extensiones

- El cubo precisa emplear parte de su espacio (bytes) en información de control:
 - cabecera, directorio de cubo, puntero encadenamiento, byte de desbordamiento, primera posición libre, ...
- También puede dejarse reservado un espacio para determinadas operaciones (*espacio libre distribuido*). Por ejemplo, en previsión de que una modificación de registro (update) haga crecer su tamaño (aka, *pctfree*)



$$T_c = \left\lfloor \frac{(E_c - T_{\text{inf ctrl}}) \cdot (1 - \text{ELD})}{T_{\text{reg}}} \right\rfloor$$

- *Otros parámetros:*
 - Ocupación (tamaño) máxima de cubo: límite preestablecido por encima del que no se insertan registros
 - Ocupación mínima de cubo (*pctused*): umbral por debajo del que se forzarán inserciones en el cubo.

Diseño: *concepción original de un objeto u obra destinado a la producción en serie*

Diseño Físico:

“Determinación de la organización física de un fichero”

Organización física: disposición de los registros en el soporte, relativa a su implementación, orden, direccionamiento, apuntamientos, etc.

Diseño Lógico:

“Descripción de la estructura lógica de los registros de un fichero”

Estructura lógica (de un registro): descripción y disposición de los elementos de datos de un registro, que en conjunto definen un individuo.

Elementos de datos: campos, vectores, y grupos repetitivos.

Diseño Físico del registro Lógico:

“Implementación de un registro lógico en secuencias de bytes”

Diseño Físico-Lógico (registro):

descripción de las cadenas de *bytes* utilizadas para almacenar registros, y de los convenios necesarios para su interpretación.



Diseño del Registro del Fichero

- **Elemento de datos** (**campo**): unidad mínima e indivisible de datos que interviene en un proceso

Notación: *campo* tipo(tamaño)
(si es opcional, se encerrará entre corchetes [...])

- **Agregado de datos**: colección de elementos (campos y/o agregados)

- **Vector**: agregado compuesto por un número fijo de elementos cuya interpretación es complementaria (entre todos los elementos definen un concepto).

Notación: (*elemento*₁; *elemento*₂; ...; *elemento*_n)

- **Grupo Repetitivo**: agregado compuesto por un número fijo o variable de elementos cuya interpretación es común (el mismo concepto para definir cada elemento)

Notación: (*elemento*)^{*} para 0..N elementos
o bien (*elemento*)⁺ para 1..N elementos

Ejemplo: diseño lógico

películasDVD

Título	C(42)		
Año	N(4)		
Productora	C(20)		
Género	C(9)		
Director	(nombre C(20),	apellido C(20))	
Actores	(nombre C(20),	apellido C(20)) ⁺⁵	
[Autor B.S.	(nombre C(20),	apellido C(20))	
Formato	C(3) ^{*3}]		

- **Es la implementación del registro en secuencias de bytes**
- Objetivo: eficiencia (reducir espacio → menor número accesos)
- Debe dar cabida a todos los registros (incluso el de mayor tamaño)

Ejemplo: hallar diseño físico-lógico pésimo y su densidad ideal

- Título (el más largo tiene 42 letras, y de media 30)
- El año de su realización siempre son 4 dígitos
- La productora son 20 car. como máximo, y de media 9 car.
- El género, 9 caracteres como máximo y de media se usan 8.1
- Los nombres de personas usan, en media, 25 caracteres (máximo 40)
- Se han llegado a registrar hasta 5 actores/actrices, y de media 2.8
- El 90% de las películas tienen banda sonora, y de media se tienen 0.5 formatos de cada banda sonora existente ('Formato' siempre gasta 3 car.)

- *El diseño físico-lógico pésimo reserva el mayor tamaño para cada campo*
- *Si la ocurrencia es menor, se usará un carácter de relleno (padding)*

películasDVD

Título	b(42)		
Año	b(4)		
Productora	b(20)		
Género	b(9)		
Director	(nombre	b(20),	apellido b(20))
Actor/iz 1	(nombre	b(20),	apellido b(20))
Actor/iz 2	(nombre	b(20),	apellido b(20))
Actor/iz 3	(nombre	b(20),	apellido b(20))
Actor/iz 4	(nombre	b(20),	apellido b(20))
Actor/iz 5	(nombre	b(20),	apellido b(20))
Autor B.S.	(nombre	b(20),	apellido b(20))
Formato1	b(3)		
Formato2	b(3)		
Formato3	b(3)		

Objetivo: Espacio (de un registro)

Volumen y Ocupación de un registro

- **volumen**: número de caracteres necesarios para almacenarlo
- **ocupación útil**: caracteres útiles del registro

$$\text{volumen} \geq \text{ocupación útil}$$

- **densidad ideal** (d_i) de un registro: relación entre cantidad de información útil y cantidad de información almacenada

$$d_i = \frac{\text{ocupación útil}}{\text{volumen real}} = \frac{\text{tamaño medio de un registro}}{\text{tamaño real de un registro}}$$

Disminuye al aumentar el espacio desperdiciado y/o la información de control

Campos de Control (*marcas*): mejoran el manejo en soporte

- Las marcas de elemento de datos ahorran espacio evitando el *padding*.

Elemento de Datos

Existencia: en campos opcionales, indica si aparece o no

Longitud: en campos variables, indica el número de caracteres

Reiteración: en grupos repetitivos, indica el número de ocurrencias

Fin de Campo: en campos indefinidos, indican que acaba

Registro

Fin (inicio) de Registro: separa registros consecutivos en el soporte

Tipo: indica el tipo de registro a continuación (f. heterogéneos)

Mapa: indica los registros que se aplican (f. heterogéneos)

Ejemplo:

películasDVD

longitud_titulo	b(1)	
Título		b(longitud_titulo)
Año		b(4)
long_productora	b(1)	
Productora		b(long_productora)
Género		b(9)
Director		(long_nombre b(1) , nombre b(long_nombre), long_apellido b(1) , apellido b(long_apellido))
num_actores	b(1)	
Actores		(long_nombre b(1) , nombre b(long_nombre), long_apellido b(1) , apellido b(long_apellido)) num_actores
Existe_BS	b(1)	
(Autor BS		(long_nombre b(1) , nombre b(long_nombre), long_apellido b(1) , apellido b(long_apellido))
num_formato	b(1)	
Formato		b(3) num_formato) existe_BS

Marcas de longitud
¿1 ó 2 bytes?

- convenio aplicado: todas las cadenas de bytes codificadas en ASCII

$$d_i = 169.95 / (1+30+4+1+9+9+2+25+1+2.8*(2+25)+1+0.9*(2+25+1+0.5*3)) \approx \mathbf{91.8\%}$$

Codificación de campos: consiste en sustituir una información por otra equivalente pero de menor tamaño

Algunos Tipos:

1. **Codificación numérica**: no almacenar dígitos (en caracteres alfanuméricos), representar números (en base 256). Ejemplo: `tlf C(9) → byte(4)`

2. **Enumerados**: equivalen a una codificación numérica natural (se puede establecer una codificación de ese tipo)

Ejemplo: `color ∈ {blanco, amarillo, naranja, rojo, verde, azul, negro}`
`color C(8) → byte(1)`

- **Pseudoenumerados**: incluyen un valor 'otros', cuya ocurrencia implica la aparición explícita del valor a continuación en el registro.

Se puede representar como sigue: `color byte(1)`
`[color_raro byte(8)]color='otros'`

3. **Fecha Juliana**: cada fecha es un número natural (desde 1/1/-4713 12:00 pm).

4. **Agrupación de varios campos**: por ejemplo, booleanos en un mapa

Ejemplo: (ver 1-35)

películasDVD

```

longitud_titulo  b(1)
Título           b(longitud_titulo)
Año              b(2)
long_productora  b(1)
Productora       b(long_productora)
Género           b(1)
Director         (long_nombre b(1), nombre b(long_nombre),
                 long_apellido b(1), apellido b(long_apellido))
num_actores      b(1)
Actores          (long_nombre b(1), nombre b(long_nombre),
                 long_apellido b(1), apellido b(long_apellido))
long_nombre      b(1),
( Autor BS      (nombre b(long_nombre),
                 long_apellido b(1), apellido b(long_apellido))
num_formato      b(1)
Formato          b(1) num_formato ) long_nombre > 0
  
```

Codificación Numérica
¿cuántos bytes?

Codificación Enumerados
¿cuántos bytes?

ocupación útil = $(30+2+9+1+25+2.8*25+0.9*(25+0.5*1)) = 159.95$ B

volumen real = $(1+30+2+1+9+1+2+25+1+2.8*(2+25)+1+0.9*(1+25+1+0.5*1)) = 173.25$ B

$d_i = 159.95 / 173.25 \approx 92.3$ %

Ejemplo: *(ver 1-35)*

películasDVD

```

longitud_titulo  b(1)
Título          b(longitud_titulo)
Año             b(2)
long_productora  b(1)
Productora      b(long_productora)
Género          b(1)
Director        (long_nombre b(1), nombre b(long_nombre),
                 long_apellido b(1), apellido b(long_apellido))
num_actores     b(1)
Actores         (long_nombre b(1), nombre b(long_nombre),
                 long_apellido b(1), apellido b(long_apellido))num_actores
long_nombre     b(1),
( Autor BS     (nombre b(long_nombre),
                 long_apellido b(1), apellido b(long_apellido))
Formato        b(1))long_nombre>0

```

**Codificación
Mapa
¿cuántos bits?**

$$\text{ocupación útil} = (30+2+9+1+25+2.8*25+0.9*(25+1)) = 159.55 \text{ B}$$

$$\text{volumen real} = (1+30+2+1+9+1+2+25+1+2.8*(2+25)+1+0.9*(1+25+1)) = 172.8 \text{ B}$$

$$d_i = 159.95 / 172.8 \approx 92.56 \%$$

A partir de los *registros*, implementados en secuencias de bytes...



¿cómo **organizar** esas secuencias de bytes para ...?

- A) que ocupen menos espacio de almacenamiento
- B) que el coste de acceso sea mínimo

Diseño Físico



Objetivos: Espacio (de un fichero)

- **densidad real (d_r)** de un fichero: relación entre cantidad de información útil y cantidad de información almacenada

$$d_r = \frac{\text{ocupación fichero}}{\text{volumen fichero}} = \frac{\text{ocupación/reg} * n^{\circ}\text{registros}}{n \text{ bloques} * T_{bq}}$$

Siempre está referida a un dispositivo (soporte) y a una organización (O_n)

Siempre se cumple que $d_i \geq d_r$

- **densidad de ocupación (d_o) de un fichero no consecutivo:** relación entre cantidad de registros almacenados y cantidad de ellos que caben, mide el porcentaje de espacio potencialmente útil no utilizado (referido siempre a una O_n)

$$d_{oc} = \frac{n^{\circ}\text{registros}}{N \text{ cubos} * T_c}$$

Objetivos: Coste



- ¿Cuál de esas organizaciones aprovecha mejor el espacio?
- ¿En cuál cuesta menos sacar un listado completo? (procesar todo)
- ¿Y si el listado debe seguir un criterio de ordenación?
- ¿Y en cuál cuesta menos encontrar un libro concreto?
- ¿Y en cuál cuesta menos insertar? ¿o borrar? ¿o modificar?
- ¿Y si tengo varios procesos distintos? ...

Objetivos: Coste Global (cjto. Procesos)

Frecuencia Relativa de un Proceso:

Todo sistema de archivos estará sometido a un cjto. de procesos $P \equiv \{P_1..P_n\}$

Cada proceso tendrá una frecuencia f_i asociada, referida a...

- a una unidad de tiempo (por ejemplo, segundos u horas)
- frecuencia relativa: al conjunto P de todos los procesos, tal que $\sum_{i=1..n} f_i = 1$

Coste Global (referido a la organización O_k y al conjunto de procesos P):

- Una organización física del Sistema de Archivos (O_k) define todas las organizaciones base de los archivos que incluye, vínculos entre los mismos, y las organizaciones auxiliares con las que cuenta.
- Cada proceso P_i tendrá en O_k un coste C_i asociado (y expresado en nº accesos o en tiempo), que dependerá directamente de la organización.
- El coste global de la organización se define como la media ponderada de accesos lógicos (por carga o unidad de tiempo).

$$C(O_k, P) = \sum_{i=1..n} C_i \cdot f_i$$

Elección del soporte



- Determinados soportes mejoran su comportamiento en acceso **serial**
 - **Lectura aleatoria:** se localiza un bloque para operarlo.
El tiempo de localización depende del tiempo que el disco tarda en dar una vuelta (latencia) y que puede estar entorno a los 5ms.
 - **Secuencialidad de disco:** acceder al bloque siguiente (consecutivo / entrelazado) es muchísimo más rápido (hasta 500 veces)
- Los SSD carecen de esta ventaja, y son menos duraderos (<10000 ciclos). Pero suelen ser más rápidos (depende de la tecnología y la operación).

Uso de Memorias Intermedias (buffers)

- Se pueden almacenar bloques en memoria privilegiada (en *páginas*)



- Acceso paginado:
al pedir un bloque, devuelve la dir. página buffer donde está; si no está en buffer, vacía una página, lo lee y lo almacena allí.
- Otras mejoras: adelantar lecturas / demorar escrituras
- **ventaja: velocidad (ahorra accesos al dispositivo)**
- **inconveniente: coste elevado → recurso muy escaso**

- *hit ratio* ($hr \mid \phi_{buf}$): porcentaje de accesos ahorrados por la M_{int}

PIO

coste real (efectivo) = $(1 - hr) * \text{coste global}$

LIO

Clave: campo (o conjunto de campos) con una función específica en la interacción de los usuarios (procesos) con el fichero.

Tipos de clave:

- **Clave de Identificación** (identificador, clave identificativa o unívoca):

identifica unívocamente cada ocurrencia en el archivo
(toma valores diferentes para cada registro)

- **Clave No Identificativa**:

Toma valores en un dominio. Los valores se repiten en distintos registros.

- cardinalidad del dominio: $\#valores(CO)$
- Coincidencia k : número de registros (en media) que adoptan ese valor
 $k = r / \#valores(CO)$, , donde r es el número de registros del fichero

Clave de Búsqueda:

Información utilizada (\pm frecuentemente) para seleccionar registros.

También entran en esta categoría los atributos proyectados (objetivo).

Clave Privilegiada:

La que cuenta con un mecanismo de localización (física) eficiente.

Las claves de búsqueda no privilegiadas se denominan *claves alternativas*

Existen varios sub-tipos:

- Clave de **Direccionamiento**: determina la ubicación del registro
- Clave de **Ordenación**: criterio de ordenación, ya sea físico (org) o lógico (proceso)
- Clave de **Indización (indexación)**: privilegiada a través de una estructura auxiliar
- Clave de Agrupamiento (**clusterización**): reúne registros con esa clave común.

- **Actualización vs. Recuperación**

- Tipos de Actualización:

- Insert: incorporar datos (*selectivo vs. bulk*)
- Delete: eliminar datos (*selectivo vs. vaciado*)
- Update: modificar datos (*selectivo vs. a la totalidad*)

- Características de Procesos de Recuperación:

- Proc. Selectivo vs Incondicional (acceso a la totalidad o *full scan*)
- Proc. sobre un archivo o varios (combinación de archivos)
- Consulta de datos agregados (analítica) vs individuales
- Resultado Ordenado (o sin ordenar)

- Actualización (selectiva) - Subprocesos

conlleva coste (acc)

tipo	ámbito	subproceso	Insert	Delete	Update
comprobacn. restricciones	registro	validación			
	archivo	unicidad (PK/UK)	✓		✓
	esquema	integridad (FK)		✓	✓
propagación	esquema	integridad (FK) (otra operación)		~	~
localización	archivo		~	✓	✓
escritura	archivo		✓	✓	✓

- Recuperación Selectiva vs Incondicional
- Selección **Identificativa** (*exact match*) vs **No Identificativa**
- Subtipos de Selección **simple** (un atributo):
 - Selección de un registro por clave identificativa (*exact match*)
 - Selección de k registros (coincidentes) por clave no identificativa
 - Selección de k registros en un rango (sobre cualquier clave de búsqueda)
- Subtipos de Selección **multiclave**: (varias claves **no identificativas**)
 - Selección de k registros en un rango (*window query*)
 - Proyección de pocos atributos del resultado de una WQ

- Acceso a la totalidad (full scan) vs *filtrado de cubos*
 - Combinación de varios orígenes de datos (*join*)
 - recorridos anidados
 - orígenes ordenados
 - composición dispersa
 - Filtrado por combinación (*consultas anidadas*)
 - Agrupación / Agregación
 - Ordenación (mezcla natural)
- Otros (independientes de org.): combinación resultados, manejo cursores, etc.

¿ *Proceso en Mem. Principal* vs. *Virtualización en Disco* !!