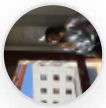

beacc
www.wuolah.com/student/beacc

Tema 1 - ABD.pdf

Apuntes Tema 1

3° Administración de Bases de Datos

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Tema 1 - Nivel Interno

Definiciones básicas

- **Campo**: almacena un valor.
- **Registro**: almacena un conjunto de campos.
- **Bloque**: almacena un conjunto de registros.
- **Fichero**: almacena un conjunto de bloques.
- **Clúster**: Conjunto de bloques agrupados en una misma vertical del disco.
- **Factor de bloqueo** (**Bfr**): es el número de registros que caben en un bloque.
- **FAT**: File Allocation Table
- Al conjunto de clústeres que están a la misma distancia del centro del disco se le llama **cilindro**.
- **LOB**: Large Object, tipo de dato complejo en un SGDB
- **Bloques homogéneos**: Almacenan registros con la misma estructura
- **Bloques heterogéneos**: Almacenan registros con distintas estructuras (clústeres)
- **Registros de tamaño fijo:** El tamaño de un registro de longitud fija es la suma del tamaño de todos sus campos. $\mathbf{R} = \mathbf{V_i}$ *Vi longitud del valor del campo i-ésimo*


CUNEF
Lidera tu futuro.
Define tu éxito.
POSTGRADO EN
DATA SCIENCE
PARA FINANZAS
SÚMATE AL ÉXITO
Excelencia, futuro, éxito.
www.cunef.edu

- **Registros de tamaño variable**: Es un registro en el que alguno de sus campos es un dato de tamaño variable. En un registro de longitud variable no se puede saber su tamaño exacto, se puede estimar.
  La estimación dependerá de : **R**=**a'**(**A**+**V**+**s**)

  a' *número medio de atributos*

  A *longitud media de los nombres de los atributos*

  V *longitud media de los valores de los atributos*

  s *número de separadores por atributo*

Sistemas de microsoft: Estructuras centralizadas

Sistemas de linux: Estructuras descentralizadas, **inodos**

Nivel físico

Bloque: Un bloque es la **unidad de información** transferida por un dispositivo de **almacenamiento masivo** o la unidad de información almacenada en el área de trabajo de la memoria (**buffer**) Un bloque tiene un **tamaño fijo** para toda la base de datos y además es un múltiplo del bloque físico del SO.

Los bloques almacenan registros, si se desperdicia espacio en un bloque el bloqueo de dicho bloque se denomina **bloqueo fijo,** sin embargo, cuando se llenan los bloques con registros hasta que no quede más espacio libre se trata de un **bloqueo partido.**

En un SGDB un bloque no pertenece a dos tablas diferentes a excepción de los clústeres, en los que se agrupan los elementos en común de las tablas.

MÉTODOS BÁSICOS DE BLOQUEO

Bloqueo fijo o entero:

- Bloqueo fijo con registros de longitud fija

$$Bfr = \frac{B - C}{R}$$

- Bloqueo fijo con registros de longitud variable: Cuando se manejan registros de longitud variable es necesario saber donde comienzan y terminan cada uno de ellos , para ello se utilizan las Marcas (M) que tienen como función separar los registros.

$$Bfr = \frac{B - C}{R + M}$$

Bloqueo partido o encadenado: En este tipo de bloqueo, los registros se escriben en bloques consecutivos de memoria pudiendo quedar un parte de un registro en el siguiente bloque, indicándose con un puntero al bloque siguiente.

En este tipo de bloqueo es **costoso buscar** registros partidos y actualizarlos pero como contrapartida **no se desperdicia** prácticamente ningún **espacio** en los bloques. Además, el bloqueo partido es la única solución cuando el tamaño de los registros excede el tamaño de los bloques.

-Bloqueo partido con registros de longitud variable

$$Bfr = \frac{B - P - C}{R + M} \qquad P \text{ es el apuntador de bloque}$$

Espacio desperdiciado (**W**): Las marcas de los registros, los apuntadores y los fragmentos de bloque no utilizados son un espacio "desperdiciado"

La proporción de espacio que se desperdicia por registros depende del tipo de registros y del bloqueo que tenga:

$$W = \frac{P + (Bfr \cdot M)}{Bfr} = \frac{P}{Bfr} + M$$

ESTRUCTURACIÓN DE LOS ARCHIVOS

- **Archivo Secuencial Fisico** (**ASF**), archivos ordenados secuencialmente como su nombre indica.
- **Archivo Secuencial Lógico** (**ASL**), archivo almacenados secuencialmente pero se intenta mejorar el acceso
- **Archivo Secuencial Indexado** (**ASI**), hay dos archivos, el que tiene el registro y el que tiene el índice
- **Archivo de Acceso Directo** (**AAD**), lo que se conoce por has; a partir de un valor de clave se obtiene una posición.

Archivo Secuencial Físico:

En este modelo de archivo, los datos se colocan según su orden de llegada sin tener en cuenta el significado de su contenido.

Hipótesis:

-Registros de estructura y longitud variable

-Registros formado por pares de la forma (Attr_id, valor)

-Dos separadores entre id y valor

¿Espacio ocupado por un registro? R

El tamaño medio de registro se calcula:

$$R = a' \cdot (A + V + 2)$$

a': Número medio de campos por cada registro

A: tamaño medio de los nombres de los atributos

V: Tamaño medio de los valores

2: Los dos separadores que se necesitan

¿Recuperación de un registro? T_F

En el peor de los casos puede ser necesario leer todo el fichero para localizar el registro deseado. Como todos los registros tienen la misma probabilidad de aparición en cualquier posición del archivo, como mínimo se leerá 1 archivo y como máximo n (todos).

$$T_F = \sum_{i=1}^{n} \frac{i}{n} \cdot T = \frac{n+1}{2} \cdot T \approx \frac{n}{2} \cdot T$$

¿Obtención del siguiente registro? T_N

Como el registro puede estar en cualquier parte, la posición se desconoce, por lo que el tiempo de obtener el siguiente registro sería igual al tiempo de obtener cualquier registro

$$T_N = T_F$$

¿Inserción de un registro ? T_I

El registro se insertará al final del archivo

$$T_I = T_W$$

¿Actualización de un registro? T_A

La actualización de un registro depende de si el tamaño del mismo varía o no.

-Si el tamaño del registro no varía o es menor, se sobreescribe sobre él mismo

$$T_A = T_F + T_W$$

-Si el tamaño del registro es mayor, hay que localizar el registro, marcarlo como borrado e insertar de nuevo, que deberá colocarse al final.

$$T_A = T_F + T_W + T_I$$

Los ficheros pueden abrirse en 3 modos: Modo apend, añadir, en el que añade las cosas al final de fichero, y el Modo sobreescrituta en el que se coloca al principio del fichero y por último el Modo de lectura.

¿Lectura de un fichero? T_x

-Independientemente del contenido del fichero

$$T_x = n \cdot T$$

donde T es el tiempo necesario para localizar un registro cualquiera

-Lectura ordenada según el valor de algún atributo del fichero

$$T_x = n \cdot T_F$$

Esta búsqueda supone un costo excesivamente elevado, que se evita ordenando previamente el dichero según en atributo de búsqueda.

¿Reorganización de un fichero? T_y

Si las actualizaciones y los borrados marcan los registros como obsoletos hay que eliminar estos registros físicamente para liberar espacio. ¿Cómo se hace? Se copia el fichero en otro y se omiten los registros marcados como obsoletos.

- **O** , es el número de registros añadidos
- **d** , es el número de registros marcados para borrar

$$T_y = (n + O) \cdot T + (n + O - d) \cdot T_w$$

Se escriben todos + los nuevos - los borrados, por lo que al final se reorganizan los que ya se habían leido. En este caos reorganizar un registro no tiene mucho sentido.

Conclusión: El archivo secuencial sirve para las copias de seguridad y poco más.

Archivo Secuencial Lógico:

En esta organización de archivos, los registros se **ordenan** según una secuencia específica que viene determinada por el contenido de un campo, la **clave física**. La clave física puede estar formada por un solo atributo o por varios.

Normalmente este tipo de organización no se aplica sobre archivos de longitud variable, se usa para registros de longitud fija.

Cuando el registro tiene un tamaño considerable, es muy costoso abrir hueco para mantener ordenado, por lo que se crea una **zona de desbordamiento** (no ordenada , como ASF). Cuando la zona de desbordamiento crece es necesario reconstruir el fichero.

Mientras los bloques están cargados en memoria caché, como es muy rápida, estos se ordenan, pero una vez pasados a discos se sigue de forma secuencial. A la hora de buscar un elemento, si se encuentra en la parte ordenada será mucho más rápido que si se encuentra en la parte desordenada.

Muchas veces los archivos secuenciales lógicos no nos interesan si la parte desordenada supera a la ordenada.

El problema de esta organización de ficheros aparece cuando hay un montón de valores nulos dentro de un registro, pues se convierte en un archivo no denso, ya que fuerzan a que sea de longitud fija.

¿Espacio ocupado por un registro? R

El tamaño del registro será constante y se calcula como:

$$R = \sum_i \mathrm{V}_i$$

¿Recuperación de un registro? T_F

-Si se busca por el valor que **no es clave**, el tiempo que se tarda es el tiempo que se tarda en buscar un elemento en un archivo secuencial físico: $\mathrm{T}_F = \frac{n}{2} \cdot T$

Si ademas la zona de desbordamiento no está vacía habrá que buscar en la zona ordenada (n) y en la no ordenada (O):

$$\mathrm{T}_F = \frac{n}{2} \cdot T + \frac{O}{2} \cdot T$$

-Si el valor por el que se busca **es clave física**, se realiza una búsqueda binaria

$$\mathrm{T}_F \cong \log_2(n) \cdot T$$

¿Obtención del siguiente registro? T_N

Si está en la zona ordenada

$$\mathrm{T}_N = \mathrm{T}$$

Si está en la zona desordenada, sería necesario leer los O archivos para saber con certeza cuál es el siguiente registro

$$\mathrm{T}_N = \mathrm{T} + O \cdot \mathrm{T}$$

¿Inserción de un registro ? T_I

Hay que mantener el orden. Depende si la zona ordenada está completa o no lo está. Recordemos que para insertar un registro debemos:

1: Localizar el punto donde se quiere insertar

2: Hacerle hueco (leer y escribir los registros restantes)

3: Escribir el registro en cuestión

$$T_I = T_F + \frac{n}{2} \cdot T + \frac{n}{2} \cdot T_w + [T_w]$$

-Si hay varios registros para insertar y hay fichero de desbordamiento, podemos insertar todos los nuevos registros en él

$$T_I = T_w$$

-pero después hay que insertarlos en el fichero maestro:

$$T_I = \frac{T_y}{O}$$

¿Actualización de un registro? T_U

La actualización de un registro depende de si se cambia o no se cambia el valor de la clave.

-Si no se cambia la clave, la actualización consistirá en la búsqueda y la escritura del nuevo registro

$$T_U = T_F + T_W$$

-Si además se produce un cambio en la clave, el proceso involucra el borrado e inserción del registros

$$T_U = T_F + T_W + T_I$$

¿Lectura de un fichero? T_x

La lectura de un archivo ASL dependerá de si tiene o no zona de desbordamiento. Si solo tiene la zona de datos el tiempo empleado en la lectura será igual que en un fichero ASF:

$$T_x = n \cdot T$$

donde T es el tiempo necesario para localizar un registro cualquiera

-En el caso de que además exista la zona de desbordamiento, deberá ordenarse primero y luego leer ambas zonas de forma simultánea:

$$T_X = T_c(O) + (n + O) \cdot T$$

¿Reorganización de un fichero? T_y

Esta operación tendrá sentido cuando exista un archivo de desbordamiento o bien cuando se haya producido un número elevado de operaciones de borrado.

¿cómo se reorganiza un ASL?

En primer lugar se ordena la zona de desbordamiento, más tarde se realiza la operación "**merge**" con la zona de datos ordenada mientras que en esta operación se omiten los archivos marcados como borrados, compactando así a la misma vez el espacio.

$$T_Y = T_c(O) + (n+O)\cdot T + (n+O-d)\cdot T_W$$

Archivo Secuencial Indexado:

En este tipo de archivo los datos están desordenados pero se tiene una estructura de datos adicional, el **índice**, que mantiene la ordenación de esos datos. Este método de acceso a los datos basado en un índice **acelera** el proceso de **búsqueda**.

El índice esta formado por un par clave - valor (ocupa muy poco espacio), donde la el valor indica la posición del fichero donde está la clave. Dependiendo de si la posición se trata de una posición de registro o de bloque, el **índice** sería **denso** o **no denso,** respectivamente.

- Un **índice denso** es aquel que tiene una entrada por cada valor.El número de registros del fichero de *índice* coincide con el número de registros del fichero *maestro*.
- Un **índice no denso** es aquel en el que no todos los valores tienen una entrada, hay una entrada por bloque.

Distinguiremos la siguiente estructura en un archivo ASI:

-El fichero de datos ASL o **fichero maestro**, con una posible área de desbordamiento (con los datos ordenados)

-El **fichero índice** , compuesto por registros de longitud fija. Los registros del índice constan todos de un **campo clave** por el que se mantienen ordenados (clave física única) y un **campo apuntador**, que contiene una dirección. A mayor nº de indices mayor tiempo se tarda en gestionar el fichero.

Habitualmente se trabaja con dos factores de bloqueo distintos, uno para el fichero maestro y otro para el fichero índice.

¿Espacio ocupado por un registro ASI Denso ? R

Los registros del fichero índice son iguales en número a los que posee el maestro, pero de menor tamaño, ya que solo almacenan pares clave-apuntador:

$$R = \sum_i V_i + \left(V_K + P \right)$$

¿Cúanto se tarda en recuperar un registro ASI Denso? T_F

Como el índice está ordenado el tiempo que tarda es el tiempo que tarda en leer el índice y el tiempo en buscarlo en la posición

$$T_F = \log_2(n) \cdot T + T$$

¿Cúanto se tarda en recuperar el siguiente registro ASI Denso? T_N

Si ya has buscado en el índice, el indicador se queda en el siguiente valor. Por lo tanto el tiempo sería el tiempo en leer la siguiente entrada en el índice y leerlo en el registro.

$$T_N = T + T = 2T$$

¿Cúanto se tarda en insertar un registro ASI Denso? T_I Hay que saberse la fórmula*

La operación de inserción de un nuevo registro supone, por un lado, las mismas operaciones que para la inserción en un fichero secuencial pero, por otro lado, es necesario actualizar también el índice.

$$T_I = T_{I1} + T_{I2} = 2 \cdot \left(T_F + \frac{n}{2} \cdot T + \frac{n}{2} \cdot T_W \left[+ T_W \right] \right)$$

El tiempo que se tarda en encontrar la posición que le corresponda en el índice, encuentra la posición que le corresponde, abre un hueco en la posición y una vez abierto el hueco escribe el registro.

- Si el fichero maestro tiene área de desbordamiento:

$$T_I = T_W + T_{I2}$$

$$T_{I2} = T_F + \frac{n}{2} \cdot T + \frac{n}{2} \cdot T_W \left[+ T_W \right] \Big)$$

¿Actualización de un registro ASI Denso? T_U

La actualización de un registro depende de si se cambia o no se cambia el valor de la clave.

-Si no se cambia la clave, la actualización consistirá en la búsqueda y la escritura del nuevo registro :

$$T_U = T_F + T_W$$

-Si además se produce un cambio en la clave, la actualización de datos deberá llevarse a cabo en ambos ficheros :

$$T_U = 2 \cdot (T_F + T_W) + T_{I1} + T_{I2}$$

¿Lectura de un fichero ASI Denso? T_x

La lectura de un archivo dependerá si se realiza por índice principal o por índice secundario y además habrá que tener en cuenta la zona de desbordamiento.

- Indice principal

$$T_x = n \cdot T$$

donde T es el tiempo necesario para localizar un registro cualquiera

- Indice secundario

$$T_x = n \cdot T + n \cdot T = 2n \cdot T$$

- Indice secundario + zona de desbordamiento

$$T_x = (n + O) \cdot T + n \cdot T + O \cdot T = 2 \cdot (n + O) \cdot T$$

¿Reorganización de un fichero ASI Denso? T_y

Esta operación se plantea cuando se ha producido un número elevado de borrados y por tanto hay una proporción alta de espacio desperdiciado.

¿cómo se reorganiza un ASI?

En primer lugar se ordena la zona de desbordamiento, mezclar ambos ficheros (maestro y desbordamiento) en un nuevo fichero, y por último crear un nuevo índice.

$$\mathrm{T}_Y = \mathrm{T}_c(O) + (n+O)\cdot T + (n+O-d)\cdot \mathrm{T}_W + (n+O-d)\cdot T_w$$

ASI No Denso : Los índices no densos surgieron para intentar disminuir el tamaño de los índices, produciendo una drástica reducción de entradas con respecto al índice denso.

¿Número de entradas de un ASI Multinivel ? R

$$\frac{n}{Bfr}$$

ASI Multinivel: Cuando el índice ocupa un gran tamaño, este se puede indexar , creando una estructura índice multinivel. El nivel superior se denomina índice raíz.

¿Número de entradas de un ASI Multinivel ? R

$$Y = \frac{B-C}{V+P}$$

Número de entradas del índice raíz: $i_1 = \frac{n}{Bfr}$

Número de entradas del restro de índices: $i_k = \frac{i_{k-1}}{y}$

Número de bloques del índice: $b_k = \frac{i_k}{y} = i_{k+1}$

Por tanto el tamaño total ocupado por un ASI Multinivel será el número total de bloques necesarios para todos sus niveles multiplicado por el tamaño de dichos bloques: $I = \left(b_1 + b_2 + \ \ldots \ b_k \right) \cdot B$

¿Espacio ocupado por un registro ASI Multinivel ? R

$$r = P + \sum_i V_i$$

Donde P es el campo de direccionamiento para el area de desbordamiento

¿Cúanto se tarda en recuperar un registro ASI Multinivel? T_F

Suponiendo que el índice esté cargado en memoria principal, para recuperar un registro habría que empezar leyendo el índice raíz y posteriormente los índices de los otros niveles hasta que nos dirigiesen al archivo principal.

$$T_F = T_M + \left(m - 1\right) \cdot T_F + T_F$$

Donde Tm es el tiempo que se tarda en revisar el índice maestro (raiz) y (m-1)Tf es el tiempo que se tarda en revisar los índices restantes.

¿Cúanto se tarda en recuperar el siguiente registro ASI Multinivel? T_N

$$T_N \simeq T$$

¿Cúanto se tarda en insertar un registro ASI Multinivel? T_I

Debemos considerar 3 posibles casos para la insercción:

- Existe espacio suficiente en el bloque que le corresponde

$$T_I \approx T_F + \frac{1}{2} \cdot Bfr \cdot \left(T_w + T \right) + T_w$$

- Se quiere insertar en el archivo principal pero no hay sitio en el bloque, por lo que la inserción provoca el desplazamiento del bloque a la zona de desbordamiento.

$$T_I \approx T_F + \frac{1}{2} \cdot Bfr \cdot \left(T_w + T \right) + 2T_w$$

- Se inserta en el fichero de desbordamiento.

$$T_I \approx T_F + 2T_w$$

¿Actualización de un registro ASI Multinivel? T_U

La actualización de un registro depende de si se cambia o no se cambia el valor de la clave.

-Si no se cambia la clave, la actualización consistirá en la búsqueda y la escritura del nuevo registro :

$$T_U = T_F + T_W$$

-Si además se produce un cambio en la clave, la actualización de datos deberá llevarse a cabo en ambos ficheros :

$$T_U = T_F + T_w + T_I$$

¿Lectura de un fichero ASI Multinivel? T_x

En este caso el índice puede ignorarse.

$$T_x \approx (n + O) \cdot T$$

¿Reorganización de un fichero ASI Multinivel? T_y

Se producirá la reorganización cuando las áreas de desbordamiento alcancen un tamaño excesivo o cuando el tiempo de recuperación se vuelva excesivo debido a las largas cadenas.

La reorganización de un ASI Multinivel supone, la lectura del archivo completo, la escritura de todos los registros menos los borrados y la creación de un nuevo índice:

$$T_y = T_x\left(n + O - d\right) \cdot T_w + k \cdot I$$

Conclusión: Los índices multinivel sólo son útiles si hay consultas que los utilicen.

Archivo Acceso Directo:

Los archivos de acceso directo son aquellos en los que para cada valor de cable se consigue una posición de **disco**, es decir, una **tabla hash,** como ya sabemos aplicando la función hash a una clave obtenemos la posición.

Función hah más simple: el módulo de un numero

La función de búsqueda de un valor teórica es de orden constante. Pero este orden no se cumple ya que es solo teórico y luego surgen problemas de colisiones.

Hay dos modos de tratar las colisiones:

Direccionamiento cerrado:

-Busqueda lineal , buscar el primer hueco

-Realeatorización o rehashing

Direccionamiento abierto:

-Listas enlazada para almacenar colisiones

-Bloques de desbordamiento

Hashing dinámico: el espacio de almacenamiento se cambia y se reestructura cambiando la función hash, se adaptan dinámicamente.

En la tabla hash el tiempo de escritura se multiplica por 2, porque si hay una colisión, en el lugar que le tocaba se escribe el nuevo valor y el otro se escribe en otro lugar de la tabla, lo que constituye 2 tiempos de escritura.

¿Espacio ocupado por un registro AAD? S_F

$$S_F = m \cdot r + c \cdot r$$

Suponiendo que los registros (r) son de longitud fija

¿Cúanto se tarda en recuperar un registro AAD? T_F

$$T_F = c + T + p \cdot T_{F\_cadena}$$

Siendo p la probabilidad de que se produzca una colisión.

¿Cúanto se tarda en insertar un registro AAD? T_I

$$T_I = C + 2 \cdot T_W$$

Una de las escrituras proviene de actualizar el el campo de dirección P en el registro del fichero principal y la otra de la escritura en el archivo de desbordamiento

¿Actualización de un registro AAD? T_U

$$T_U = T_F + T_W$$

La actualización de un registro por cualquier campo que no sea la clave, consiste en encontrarlo y en volver a escribirlo en el mismo sitio.

¿Lectura de un fichero AAD? T_x

$$T_x \simeq (m + c) \cdot T$$

¿Reorganización de un fichero AAD? T_y

La reorganización se realiza si ha aumentado mucho el número total de registros que se va a almacenar y no se tomaron medidas de previsión para su ampliación, o cuando se han realizado muchas eliminaciones.

$$T_Y = T_x + T_{carga}$$

$$T_{carga} = \sum_{i=1}^{n} T_i\left(i\right)$$