SESION 1: ARCHIVOS

Ejercicio (Importante: Tamaño de archivos, ordenación, organización de archivos, estimación de tuplas de una consulta, coste de buscar en una consulta sencilla)

Se dispone de un archivo r de datos que contiene información sobre estudiantes donde se almacena:

- Número de carnet: 8 bytes, todos los valores del campo son diferentes. El campo es un serial que comienza por valor 1.
- Nombre del alumno: 40 bytes
- Código de carrera: 2 bytes con valores incluidos (IN) (0,1,2,3,4,5,6,7,8,9)
- Edad: 8 bytes
- Índice académico: 2 bytes

El número de registros a almacenar es de 100.000 y cada bloque tiene un tamaño de 512 bytes. El grado de ocupación de cada bloque es del 65%. La longitud de puntero a bloque es de 6 bytes y la longitud de puntero a registro de 7 bytes. Cada bloque tiene 12 bytes de control. Se pide para cada una de las situaciones que se muestran a continuación lo siguiente:

1. Para un archivo que se organiza en forma de montículo (sin ordenación) calcular el tamaño de bloques. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

   $L_{registro}$= 8+40+2+8+2=60 bytes

   Tamaño de bloque útil B_{util} = 512-12 = 500 bytes. Al 65% = 500*0.65 = 325 bytes

   N registros/ Bloque fr = $\lfloor 325 / 60 \rfloor = \lfloor 5.41 \rfloor$ = 5 registros / bloque al 65% de capacidad

   Ó

   N registros/ Bloque fr = $\lfloor 500 / 60 \rfloor = \lfloor 8.33 \rfloor$ = 8 registros / bloque al 100% → al 65% serán 8 * 0.65 = 5.2 → 5 reg (redondeando al entero más pequeño)

   N bloques br = $\lceil 100.000 / 5 \rceil$ = 20.000 bloques

   a. $\sigma_{carnet=2345}(r)$

      V(carnet)=todos diferentes = 100.000

      Registros a recuperar nrc=nr/V(carnet)=100.000/100.000= 1 reg

      Archivo sin ordenación → lectura secuencial → Coste =leer todos los bloques = br= 20.000 bloq

   b. $\sigma_{código\_carrera=2}(r)$

      V(codigo_carrera)= 10 valores diferentes

      Nrc=nr/V(codigo_carrera)=100.000/10 = 10.000 reg de media

      Archivo in ordenación → lectura secuencial → Coste = br = 20.000 blq

2. Para un archivo que se organiza en forma secuencial (ordenado) por el campo carnet calcular el tamaño de bloques. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

   En este caso el fichero está ordenado por el campo carnet, luego la información a guardar no cambia → ocupa lo mismo que el archivo en montículo → br= 20.000 bloques (mismos cálculos que aparado 1)

   a. $\sigma_{carnet=2345}(r)$

      Número de registros a recuperar no cambia, es lo mismo que antes, luego 1. Da igual que esté ordenado o no

      Coste de buscar → dos posibilidades:

      - Secuencial → leer todos los bloques → Coste = br = 20.000 blq
      - Búsqueda binaria → No se leen todos los bloques → de media se leen Coste = $\lceil \log_2(br) \rceil + \lceil nrc/fr \rceil - 1 = \lceil \log_2(20.000) \rceil + \lceil 1/5 \rceil - 1 = \lceil 14.28 \rceil + \lceil 1/5 \rceil - 1 = 15 + 1 - 1 = 15$ bloq.

   b. $\sigma_{código\_carrera=2}(r)$

      El número de registros a recuperar no cambia si la información en el fichero es la misma, da igual cómo se organice → nrc = 10.000 reg

      En este caso el fichero está ordenado por campo carnet → no está ordenado por código_carrera → lectura secuencial → Coste = br= 20.000 blq

3. Para un archivo que se organiza en forma secuencial (ordenado) por el campo código de carrera, se ha añadido un puntero de 12 bytes a cada registro que sirve para formar la lista enlazada que marque la ordenación lógica del fichero según ese campo. Calcular el tamaño de bloques del archivo. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

   Ahora existe un campo interno más para cada registro de 12 bytes, luego la longitud de registro cambia.

   $L_{registro}$= 8+40+2+8+2+12=72 bytes

   Tamaño de bloque útil B_{util} = 325 bytes al 65%

   N registros/ Bloque fr = $\lfloor 325 / 72 \rfloor = \lfloor 4.51 \rfloor$ = 4 registros / bloque al 65% de capacidad

   El archivo ahora ocupa br = $\lceil 100.000 / 4 \rceil$ = 25.000 bloques

   a. $\sigma_{carnet=2345}(r)$

      Número de registros a recuperar nrc = 1

      El archivo no está ordenado por campo carnet → lectura secuencial → coste = br = 25.000 bloques

   b. $\sigma_{código\_carrera=2}(r)$

      Número de registros a recuperar nrc = 10.000 reg que están consecutivos

      Para buscar esos registros hay dos posibilidades en este caso:

- Lectura secuencial → Coste = br = 25.000 blq
- Búsqueda binaria → Coste = $\lceil \log_2(br) \rceil + \lceil nrc/fr \rceil - 1 = \lceil \log_2(25.000) \rceil + \lceil 10.000/4 \rceil - 1 = \lceil 14.609 \rceil + \lceil 10.000/4 \rceil - 1 = 15 + 2500 - 1 = 2514$ bloq.

4. Para un archivo que se organiza en forma de árbol B+ por el campo carnet calcular el tamaño de bloques. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

   Un árbol B+ está compuesto de una serie de nodos donde se clasifican como raíz, intermedios y hoja. Solo hay 1 nodo raíz, y los intermedios y raíz son iguales. En este caso al ser una organización de archivo, los nodos hoja contienen los registros a organizar. Cualquier nodo ocupa 1 bloque. El árbol se organiza por un campo, en este caso el campo carnet.

   <u>Nodo raíz/intermedio</u>

   n * Lpbloque + (n-1)*Lcarnet <= B (espacio útil)

   n*6 + (n-1)*8 <=325 → 14* n <=333 → n <= 23.78 → n = 23 (número de punteros a bloque, 22 valores del campo en el nodo) al 65%

   <u>Nodo hoja</u>

   nh*Lregistro + Pbloque <= 325

   nh*60 + 6 <= 325 → 60*nh <= 319 → nh <= 5.3 → nh = 5 registros al 65%

   Ocupa el archivo → se organiza por niveles, desde los nodos hoja hasta la raíz

   Número nodos hoja = $\lceil nr / nh \rceil = \lceil 100.000/5 \rceil = 20.000$ blq

   Número nodos intermedio 1 = $\lceil 20.000/23 \rceil = \lceil 869.565 \rceil = 870$ blq

   Número nodos intermedio 2 = $\lceil 870/23 \rceil = \lceil 37.82 \rceil = 38$ blq

   Número nodos intermedio 3 = $\lceil 37/23 \rceil = \lceil 1.60 \rceil = 2$ blq

   Raíz = $\lceil 2/23 \rceil = 1$

   Hay 5 niveles (hoja, 3 intermedios, raíz)

   Tamaño del archivo br=suma de todos los bloques = 20.000 + 870 + 38 +2 +1 = 20.911 blq

   a. $\sigma_{carnet=2345}(r)$

      Número de registros a recuperar nrc = 1 (no cambia)

      Buscar el registro, el archivo está organizado por el árbol B+, se debe empezar a buscar siempre por el nodo raíz, luego 1 bloque por nivel hasta llegar al nodo hoja, donde se verán cuántos hay que leer.

      Coste = 1 (raíz) + 1 (intermedio 1) + 1 (intermedio 2) + 1 (intermedio 3) + 1 (1 nodo hoja) = 5 bloques

   b. $\sigma_{código\_carrera=2}(r)$

      Número de registros a recuperar = 10.000 (no cambia)

Hay que leer todos los nodos hoja, luego se localiza el primer nodo hoja usando el árbol = 1 (raíz) + 3 (nodos intermedios) + 20.000 hojas = 20.004 blq

5. Para un archivo que se organiza por medio de asociación (hash) con función de asociación **carnet mod 100**, calcular el tamaño de bloques. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

   Se denomina particionamiento por función de asociación hash o organización de archivo asociativa. El número de particiones N lo da la función de asociación carnet mod 100. Se examina la función obteniendo los posibles valores. En este caso de 0 a 99 → 100 particiones o cajones

   De media cada partición tendrá de medio nr / N = 100.000/100 = 1000 reg

   Cada partición es un archivos secuencial, luego como Lregistro = 60 bytes, factor de bloque es fr = 5 reg/bloq al 65%, cada partición ocupa brpartición = $\lceil 1000/5 \rceil$ = 200 bloq. El archivo ocupa en total 100*2000 = 20.000 blq

   a. $\sigma_{carnet=2345}(r)$

      Número de registros a recuperar es 1

      Para buscar se debe evaluar la función de asociación: 2345 mod 100 = 45, luego hay que leerse solo la partición 45. Coste = leer secuencialmente la partición 45 = 200 bloq.

   b. $\sigma_{código\_carrera=2}(r)$

      Número de registros a recuperar es 10.000

      No se puede usar la función de asociación → no se sabe donde esta cod_carrera = 2, luego hay que leer secuencialmente todas las particiones → 100 * 200 = 20.000 blq

6. Cuestiones opcionales para resolver por los alumnos. Considerando los tipos de archivos del ejercicio anterior, determinar el número de registros a recuperar y el coste de buscar para cada una de las siguientes consultas.

   a. $\sigma_{carnet>=10.000}(r)$

      Número de registros a recupera. V(carnet)=100.000 , comenzando por el 1, luego va de 1 -100.000. La condición es >=10.000, luego serán 100.000-10.000+1 = 90.001 valores diferentes queremos, Nv=90.001.

      nrc=Nv*nr/V(carnet)=90.001*100.000/100.000 = 90.001 reg

      Para leer carnet por cada apartado:

      - Secuencial : Coste = br = 20.000 blq
      - Ordenado : Coste = $\lceil \log_2(20.000) \rceil + \lceil 90.001/5 \rceil - 1 = 15 + 18001 - 1 = 18.014$ blq
      - B+ → Buscar el valor 10.000 + buscar 90.001 registros consecutivos en los nodos hoja = 1 +1 + 1 +1 + $\lceil 90.001/5 \rceil$ = 4 + 18001 = 18.005 blq

- Hash → Hay particiones → buscar en todas las particiones → Coste = 100*200 = 20.000 reg

b. $\sigma_{carnet<10.000}(r)$

Hay que recuperar desde el 1 al 9.999 carnet. Luego son nrc = Nv * nr/V(carnet) = 9.999*100.000/100.000 = 9.999 reg

- Secuencial : Coste = br = 20.000 blq
- Ordenado: No hace falta búsqueda binaria. Hay que leer desde el bloque 1 y leer 9.999 reg → Coste = $\lceil 9.999/5 \rceil$ = 2.000 blq
- B+ → Buscar el valor primer nodo hoja + buscar 9.999 registros consecutivos en los nodos hoja = 1 +1 + 1 +1 + $\lceil 9.999/5 \rceil$ = 4 + 2000 = 2.004 blq
- Hash → Hay particiones → buscar en todas las particiones, hay 9999 valores diferentes que evaluar en hash → Coste = 100*200 = 20.000 reg

c. $\sigma_{carnet<>10.000}(r)$

Hay que recuperar todos menos el 10.000: 99.999*100.00/100.00=99.999, casi todo el fichero.

- Secuencial : Coste = br = 20.000 blq
- Ordenado: No hace falta búsqueda binaria. Hay que leer desde el bloque 1 hasta el 20.000 → Coste = 20.000 blq. Si se hiciese 99.999 búsquedas serían: 99.999*$\lceil \log_2 (20.000) \rceil$=1.399.986 bloq
- B+ → Buscar el valor primer nodo hoja + buscar todos los registros menos el 10.000 en los nodos hoja = 1 +1 + 1 +1 + 20.000 = 4 + 20.000 = 20.004 blq. Si se hacen 99.999 búsquedas serían: 99.999* (4 + 1) = 499.995 bloq
- Hash → Hay particiones → buscar en todas las particiones, hay 99.999 valores diferentes que evaluar en hash → Coste = 100*200 = 20.000 reg

d. $\sigma_{carnet>10.000 \wedge carnet < 15.000}(r)$

Se mira la condición, es un rango > 1000 y < 15000, luego son 4.999 valores diferentes. Número de registros a recuperar = 4999*100.000/100.000 = 4.999 reg

- Secuencial : Coste = br = 20.000 blq
- Ordenado: Búsqueda binaria. Se busca el valor . Hay que leer desde el valor 9.999 hasta el 14.999 → Coste = $\lceil \log_2 (20.000) \rceil$ + $\lceil 4.999/5 \rceil$ -1 = 15 + 1000 -1 = 1014 blq
- B+ → Buscar el primer valor en el árbol y luego leer consecutivamente los nodos hoja con los 4999 valores necesitados = 1 +1 + 1 +1 + $\lceil 4.999/5 \rceil$ = 4 + 1.000 = 1.004 blq.

- Hash → Hay particiones → buscar en todas las particiones en este caso, hay 4999 valores diferentes que evaluar en hash → Coste = 100*200 = 20.000 reg

e. $\sigma_{\text{código\_carrera} > 6}(r)$

Se mira la condición, es un rango > 6, luego nos valen el 7,8,9, 3 valores diferentes se quieren. Número de registros a recuperar = 3*100.000/10 = 30.000 reg

- Secuencial : Coste = br = 20.000 blq
- Ordenado: Búsqueda binaria. Se busca el valor 7. Hay que leer desde el valor 7 hasta el 9. → Coste = $\lceil \log_2(25.000) \rceil + \lceil 30.000/4 \rceil - 1 = 15 + 7500 - 1 = 7514$ blq
- B+ por carnet: Coste = 4 + 20.000 bloq —> Se tienen que leer todos los hojas, Se localiza la primera con 4 accesos
- Hash por carnet: no se puede usar la función hash que está definida por carnet: Coste = leer todas las particiones = 100*200=20.000 bloq

f. $\sigma_{\text{código\_carrera} < 5}(r)$

Se mira la condición, es un rango < 5, luego nos valen el 0,1,2,3,4 valores diferentes que se quieren. Número de registros a recuperar = 5*100.000/10 = 50.000 reg

- Secuencial : Coste = br = 20.000 blq
- Ordenado: Se lee desde el bloque 1 hasta localizar el primer valor de 5. → Coste = $\lceil 50.000/4 \rceil = 12.500$ blq
- B+ por carnet: Coste = 4 + 20.000 bloq —> Se tienen que leer todos los hojas, Se localiza la primera con 4 accesos
- Hash por carnet: no se puede usar la función hash que está definida por carnet: Coste = leer todas las particiones = 100*200=20.000 bloq

g. $\sigma_{\text{código\_carrera} <> 8}(r)$

Se requieren todos los registros menos los que tienen valor 8. Número de registros a recuperar = 9*100.000/10 = 90.000 reg

- Secuencial : Coste = br = 20.000 blq
- Ordenado: Se lee desde el bloque 1 hasta localizar el valor de 7. → Coste = $\lceil 70.000/4 \rceil = 17.500$ blq

  Luego hay que buscar el código_carrera=7 y leer 20.000 reg seguidos, luego coste = $\lceil \log_2(25.000) \rceil + \lceil 20.000/4 \rceil - 1 = 15+5000-1 = 5014$ blq.

  Total P 17.500+5014 = 22.514 bloq

- B+ por carnet: Coste = 4 + 20.000 bloq —> Se tienen que leer todos los hojas, Se localiza la primera con 4 accesos
- Hash por carnet: no se puede usar la función hash que está definida por carnet: Coste = leer todas las particiones = 100*200=20.000 bloq

h. $\sigma_{\text{código\_carrera} > 7 \wedge \text{código\_carrera} <= 9}(r)$

Se requieren todos los registros > 7 y <=9, esos son el 8 y el 9, 2 de 10 valores. Número de registros a recuperar = 2*100.000/10 = 20.000 reg

- Secuencial : Coste = br = 20.000 blq
- Ordenado: Se lee desde el valor 8 hasta el 9 hasta localizar el primer valor de 8. → Coste = $\lceil \log_2 (25.000) \rceil + \lceil 20.000/4 \rceil - 1$ = 15 + 5000 -1 = 5014 blq
- B+ por carnet = 4 + 20000 = 20.004 bloq
- Hash por carnet = 100*200 = 20.000 bloq

i. $\sigma_{\text{carnet} = 53456 \wedge \text{código\_carrera} = 9}(r)$

Se requieren los registros que tienen el valor de carnet 53456 y código de carrera = 9. Solo se puede hacer una búsqueda con un AND. Número de registros a recuperar nrc = nr /(V(carnet)*V(codigo_carrera) = 100.000 /(100.000*10)= 0.1 reg. Viendo la realidad solo puede haber 1 como máximo.

- Secuencial : Coste = br = 20.000 blq
- Ordenado:
  - Sobre carnet → Número de registros a recuperar = 1.

    Coste = $\lceil \log_2 (20.000) \rceil + \lceil 1/5 \rceil - 1$ = 15 +1 -1 = 15 blq
  - Sobre codigo_carrera → Número de registros a recuperar = 100.000/10 =10.000 reg

    Coste = $\lceil \log_2 (25.000) \rceil + \lceil 10.000/4 \rceil - 1$ = 15 +2.500 -1 = 2484 blq
- Sobre B+ en carnet: Coste = 1 + 1 + 1 + 1 + 1 = 5 bloq
- Sobre Hash carnet: Coste = leer una partición = 200 blq

j. $\sigma_{\text{carnet} = 53456 \vee \text{código\_carrera} = 9}(r)$

Se fija uno en la condición, es un OR, luego valen los registros que cumplen al menos una de las condiciones. Un OR equivale a una suma implícitamente, luego el número de registros a recuperar es: 100.000/100.000 + 100.000/10 = 1 + 10.000 = 10.001 reg

Coste:

- Caso 1: Secuencial, leer una sola vez todo → Coste = br=20.000 blq
- Caso 2: Ordenado por carnet → Se podría usar la ordenación en carnet, pero no hay nada más que secuencial en código_carrera, luego → Lectura Secuencial → Coste = br =20.000 blq
- Caso 3: Ordenador por código_carrera → Una de las condiciones no tiene nada más que buscar secuencialmente → Coste = br =25.000 blq
- Caso 4: B+ sobre carnet → solo vale para buscar por carnet → hay que leer todas las hojas para encontrar los otros registros de código_Carrera → coste)= 4 hasta buscar a la primera hoja + 20.000 hojas = 20.004 blq
- Caso 5: Hash sobre carnet → No vale sobre código_Carrera → leer todas las particiones → Coste = 100*200= 20.000 blq