SESION 2: ÍNDICES

Ejercicio (Importante: índices multinivel secuencial, índice B+, índice Hash, coste de buscar con un índice)

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. El archivo se encuentra ordenado por número de carnet
- 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. Calcular el número de bloques de un índice multinivel secuencial sobre el campo carnet de 3 niveles. Para las siguientes consultas determinar el número de registros estimados y el coste de lectura.

   Clasificamos el índice → primario + campo clave → primer nivel puede ser:

   - Denso → nri = V(carnet)
   - Disperso → nri = Br (número de bloques de la tabla)

   Si es denso → nri = V(carnet) = 100.000 reg índice

   $L_{ri} = L_{carnet} + L_{preg} = 8 + 7 = 15$ bytes.

   $B_{util} = B - L_{control} = 521 - 12 = 500$ bytes * 0.65 = 325 bytes

   $Fri = \lfloor 325 / 15 \rfloor = 21$ reg/bloq

   $Bri = \lceil 100.000 / 21 \rceil = 4762$ bloq

   Para segundo nivel y superior → se hace índice disperso sobre el nivel anterior

   $Lri2 = L_{carnet} + L_{pbloque} = 8+6 = 14$ bytes, $Fri = \lfloor 325 / 14 \rfloor = 23$ reg/bloq

   $Bri2 = \lceil 4762 / 23 \rceil = 208$ bloq , nivel 2

   $Bri3 = \lceil 208 / 23 \rceil = 10$ bloq , nivel 3

   Número de bloques = 10 + 208 + 4762 = 4980 bloq

   Si es disperso → nri = br (una entrada por cada bloque de datos) = 20.000 reg índice

   Lri= Lcarnet + Lpreg = 8 + 7 = 15 bytes.

$B_{util}=B-L_{control}$ = 521-12 = 500 bytes * 0.65 = 325 bytes

Fri=$\lfloor 325 / 15 \rfloor$ = 21 reg/bloq

Bri=$\lceil 20.000 / 21 \rceil$ = 953 bloq

Para segundo nivel y superior → se hace índice disperso sobre el nivel anterior

$Lri2=L_{carnet} + L_{pbloque}$ = 8+6= 14 bytes, Fri=$\lfloor 325 / 14 \rfloor$ = 23 reg/bloq

Bri2=$\lceil 953 / 23 \rceil$ = 42 bloq , nivel 2

Bri3=$\lceil 42 / 23 \rceil$ = 2 bloq , nivel 3

Número de bloques = 2 + 42 + 953 = 997 bloq

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

Nrc= Nv*nr/V(carnet)=1*100.000/100.000 = 1 reg

Coste = Cindice + Cdatos = Buscar nivel superior + 1 bloque por cada niveles inferiores al superior + bloques de la tabla a recuperar

Para el denso: Coste = $\lceil \log_2 (bri3) \rceil + 1 + 1 + \lceil nrc/fr \rceil = \lceil \log_2 (10) \rceil + 1 + 1 + \lceil 1/5 \rceil = 4 + 1 + 1 + 1 = 7$ bloq

Para el disperso: Coste = $\lceil \log_2 (bri3) \rceil + 1 + 1 + \lceil nrc/fr \rceil = \lceil \log_2 (2) \rceil + 1 + 1 + \lceil 1/5 \rceil = 1 + 1 + 1 + 1 = 4$ bloq

Búsqueda secuencial = Coste = 20.000 blq (peor)

Búsqueda binaria = $\lceil \log_2 (20.000) \rceil + \lceil 1/5 \rceil - 1 = 15$ blq

b. $\sigma_{código\_carrera=2}(r)$

Nrc=Nv*nr/V(código_carnet)=1*100.000/10 = 10.000 blq

Solo búsqueda secuencial → Coste = br = 20.000 blq

2.Calcular el número de bloques de un índice multinivel secuencial sobre el campo código_carrera. Para las siguientes consultas determinar el número de registros estimados y el coste de lectura.

Clasificamos el índice → secundario + campo no clave → cajones de punteros

Primer nivel solo denso → nri = V(código_carrera)=10

Como es denso → nri = V(código_carnet) = 10 reg índice

Lri= Lcodigo_carrera + Lbloques = 2 + 6 = 8 bytes.

Butil=B-Lcontrol = 521-12 = 500 bytes * 0.65 = 325 bytes

Fri=$\lfloor 325 / 8 \rfloor$ = 40 reg/bloq

Bri=$\lceil 10 / 40 \rceil$ = 1 bloq

Como solo hay 1 bloque en el nivel inferior → solo tiene un nivel

Cajones de punteros → Número de cajones Nc = V(código_carnet) = 10

Cada cajon tiene un número de registros nc=nr/V(código_carnet)=100.000/10)= 10.000 reg

Los cajones de punteros llevan los Pregistro → Lrc=LPreg= 7 bytes → Frc=⌊325 / 7⌋ = 46 reg/bloq → brc = ⌈ 10.000 / 46⌉ = 218 blq

Número de bloques del índice = bri + Nc*brc = 1 + 10*218 = 2181 bloq

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

Solo búsqueda secuencial → Coste = br = 20.000 bloq

b. $\sigma_{código\_carrera=2}(r)$

Nrc=Nv*nr/V(código_Carnet)=1*100.000/10 = 10.000 reg (ordenados → Consecutivos)

Coste = 1 + 1 cajon de punteror + nrc = 1 + 218 + 10.000 = 10.219 bloq

Búsqueda secuencial→ Coste = br = 20.000 bloq

3. Calcular el número de bloques de un índice B+ sobre el campo carnet. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

Clasificamos el índice → primario + campo clave.

Los B+ siempre densos → nri = V(carnet)=100.000 reg

Nodo raíz/intermedio

n*Lploque + (n-1)*Lcarnet <= Butil → n*6+(n-1)*8<=325 → 14*n<=333 → n= 23 punteros a bloque

Nodo hoja

nh*(Lcarnet + Lpreg)+ Lpbloque <= Butil → nh*(8+7)+6<=325 → 15*nh<=319 → n= 21 valores del campo

Nodos hoja = ⌈ 100.000 / 21⌉ = 4762 bloq

Nodos intermedio 1 = ⌈ 4762 / 23⌉ = 208 bloq

Nodos intermedio 2 = ⌈ 208 / 23⌉ = 9 bloq

Nodo raíz = = ⌈ 9 / 23⌉ = 1 bloq

Número de bloques = 1+9+208+4762 = 4980 bloq , Son 4 niveles

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

Nrc = 1 reg, como vimos antes.

Para buscar los registros:

Número Búsqueda secuencial→ Coste = br = 20.000 bloq

Búsqueda en el B+ (1 sola) = niveles + nrc/fr = 4 + 1 = 5 bloq

b. $\sigma_{código\_carrera=2}(r)$

Nrc = 10.000 reg, como vimos antes.

Para buscar los registros:

Búsqueda secuencial→ Coste = br = 20.000 bloq

4. Calcular el número de bloques de un índice B+ sobre el campo código_carrera. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

Clasificamos el índice → secundario + campo no clave → Cajones de punteros

Los B+ siempre densos → nri = V(código_carrera)= 10 reg

Nodo raíz/intermedio

n*Lploque + (n-1)*Lcodigo_carrear <= Butil → n*6+(n-1)*2<=325 → 8*n<=327 → n= 40 punteros a bloque

Nodo hoja

nh*(Lcodigo_carrera + Lbloque)+ Lpbloque <= Butil → nh*(2+6)+6<=325 → 8*nh<=319 → n= 39 valores del campo

Nodos hoja/Raíz = $\lceil 10 / 39 \rceil$ = 1 bloq

Número de bloques = 1 bloq, es 1 nivel

Hay cajones de punteros. Lo mismo que en el apartado 2.

Cajones de punteros → Número de cajones Nc = V(código_carnet) = 10

Cada cajon tiene un número de registros nc=nr/V(código_carnet)=100.000/10)= 10.000 reg

Los cajones de punteros llevan los Pregistro → Lrc=LPreg= 7 bytes → Frc=$\lfloor 325 / 7 \rfloor$ = 46 reg/bloq → brc = $\lceil 10.000 / 46 \rceil$ = 218 blq

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

Nrc = 1 reg, como vimos antes.

Para buscar los registros:

Búsqueda secuencial→ Coste = br = 20.000 bloq

b. $\sigma_{código\_carrera=2}(r)$

Nrc = 10.000 reg, como vimos antes.

Para buscar los registros:

Búsqueda secuencial→ Coste = br = 20.000 bloq

Búsqueda en el B+ (1 sola) = Número niveles + 1 cajón de punteros + los registros a los que apunta el cajón (nrc) = 1 + 1*218+10.000 = 10.219 blq

5. Calcular el número de bloques de un índice Hash creado sobre el campo carnet cuya **función de asociación devuelve un número binario de 8 bits**.

Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

Clasificamos el índice → primario + campo clave

Los hash son siempre densos → nri = V(campo) = nr = 100.000 reg

Se examina la función de asociación → número binario de 8 bits → Número de cajones → N = 2^8 =256 cajones.

Cada cajón lleva de media nc = nri / N = 100.000 / 256 = 390,625 reg → 391 reg

Lric = Lcarnet + Lpreg = 8 + 7 = 15 reg, frc= =⌊325 / 15⌋ = 21 reg / bloq , bc = ⌈391 / 21⌉ = 19 bloq

El índice ocupa : Bri = N* Bc = 256* 19 = 4864 blq

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

Nrc = 1 reg, como vimos antes.

Para buscar los registros:

Búsqueda secuencial→ Coste = br = 20.000 bloq

Búsqueda binaria = ⌈ $\log_2$ (20.000)⌉ + ⌈ 1/5⌉ -1 = 15 blq

Búsqueda por Índice Hash → 1 valor en 1 cajón + ⌈ nrc/fr⌉ = 19 + 1 = 20 blq

b. $\sigma_{código\_carrera=2}(r)$

Nrc = 10.000 reg, como vimos antes.

Para buscar los registros:

Búsqueda secuencial→ Coste = br = 20.000 bloq

6. Calcular el número de bloques de un índice Hash creado sobre el campo código_carrera cuya función de asociación es **código_carrera mod 5**. Para cada una de las siguientes consultas determinar el número de registros estimados y el coste de lectura.

Clasificamos el índice → secundario + campo no clave → cajones de punteros

Los hash son siempre densos → nri = V(campo) = nr = 10 reg

Se examina la función de asociación → código_carrera mod 5 → Número de cajones → N = 5 (del 0 al 4)

Cada cajón lleva de media nc = nri / N = 10 / 5 = 2 reg

Lric = Lcarnet + Lbloque = 2 + 6 = 8 reg, frc= =⌊325 / 8⌋ = 40 reg / bloq , bc = ⌈2 / 40⌉ = 1 bloq

Hay cajones de punteros. Lo mismo que en el apartado 2.

Cajones de punteros → Número de cajones Nc = V(código_carnet) = 10

Cada cajon tiene un número de registros nc=nr/V(código_carnet)=100.000/10)= 10.000 reg

Los cajones de punteros llevan los Pregistro → Lrc=LPreg= 7 bytes → Frc=⌊325 / 7⌋ = 46 reg/bloq → brc = ⌈ 10.000 / 46⌉ = 218 blq

El índice ocupa : Bri = N* Bc + V(código_Carrera)*Brc = 5 * 1 + 10 * 218 = 2185 blq

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

Nrc = 1 reg, como vimos antes.

Para buscar los registros:

Búsqueda secuencial→ Coste = br = 20.000 bloq

Búsqueda binaria = ⌈ $\log_2$ (20.000)⌉ + ⌈ 1/5⌉ -1 = 15 blq

b. $\sigma_{código\_carrera=2}(r)$

Nrc = 10.000 reg, como vimos antes.

Para buscar los registros:

Búsqueda secuencial→ Coste = br = 20.000 bloq

Búsqueda índice hash → 1 valor a buscar en 1 cajón + leer el cajón de punteros correspondiente + los registros que apuntan nrc = 1 + 218 + 10.000 = 10.219 bloq

7. Cuestiones opcionales para resolver por los alumnos. Considerando los tipos de índices 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 recuperar nrc = 90.001*100.000/100.00 = 90.001 reg

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- Binaria → Coste = ⌈ $\log_2$ (20.000)⌉ + ⌈ 90.001/5⌉ -1 = 15+18.001 -1 = 18.015 bloq
- B+ en carnet, como es primario, se busca valor 10.000 luego se leen 90.001 registros consecutivos → Coste = 4 + ⌈90.001/5⌉ = 18.004 bloq
- Hash en carnet, como es primario, se busca valor 10.000 y luego se leen 90.001 registros consecutivos → Coste = 19 (1 Cajón) + ⌈ 90.001/5⌉ = 18.019 bloq
-

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

Número de registros a recuperar nrc = 9999*100.000/100.00 = 9.999 reg

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- Secuencial limitada → Solo hay que leer 9.999 reg empezando por el principio del archivo ordenado → Coste = $\lceil 9.999/5 \rceil$=2000 bloq
- Binaria → Coste = $\lceil \log_2 (20.000) \rceil + \lceil 9.999/5 \rceil -1$ = 15+2000 -1 = 2014 bloq
- B+ en carnet, como es primario, se busca valor 1 y luego se leen 9.999 registros consecutivos → Coste = 4 + $\lceil 9.999/5 \rceil$ = 2004 bloq
- Hash en carnet, como es primario , se busca valor 1 y luego se leen 9.999 registros consecutivos → Coste = 19 (1 Cajón) + $\lceil 9999 /5 \rceil$ = 2019 bloq

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

Número de registros a recuperar nrc = 99999*100.000/100.00 = 99.999 reg. Queremos todos menos el 10.000

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- Binaria → No están todos los registros seguidos, luego leer desde el 1 hasta al 9.999, y luego localizar el 10001 para leer hasta el 100.000. En este caso al ser un campo con todos los valores diferentes y al querer todos menos 1, mejor secuencial.
- B+ en carnet, ocurre lo mismo que antes.
- Hash en carnet, ocurre lo mismo que antes, hay que leerse todos los bloques del archivo, luego saldrá más coste que la secuencial.

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

Número de registros a recuperar nrc = (14999-10001+1)*100.000/100.000 = 4.999 reg. Desde el 10.001 al 14.999, incluidos.

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- Binaria → Coste = $\lceil \log_2 (20.000) \rceil + \lceil 4.999/5 \rceil -1$ = 15+1000 -1 = 1014 bloq. Busco el primer valor que cumple y luego leo los registros consecutivamente del archivo de datos que para eso está ordenado.

- B+ en carnet → Coste = 4 + $\lceil 4.999/5 \rceil$ = 4 + 1000 = 1004 bloq. Localizo el primer valor que cumple la condición con el árbol y luego tiro por el archivo de datos.
- Hash en carnet → Coste = 19 (1 Cajón) + $\lceil 4.999/5 \rceil$ = Localizo el primer valor que cumple condición y luego sigo por el archivo de datos.

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

Número de registros a recuperar nrc = (9-7+1)*100.000/10 = 30.000 reg. Se pide el 7,8,9. El archivo no está ordenado, luego sacar todos los registros a partir del índice.

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- B+ en carnet → Coste = 1 (nodo raíz/hoja para leer el puntero a cajones de punteros a registro del 7,8,9) + 3*218 + 3*10.000= 30.655 bloq. Peor que la secuencial.
- Hash en carnet → Coste = 3*1 (caen en 3 cajones diferentes) + 3*218 + 3*10.000 = 30.657 bloq. Peor que la secuencial.

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

Número de registros a recuperar nrc = (4-0+1)*100.000/10 = 50.000 reg. Se pide el 0,1,2,3,4. El archivo no está ordenado, luego sacar todos los registros a partir del índice.

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- B+ en carnet → Coste = 1 (nodo raíz/hoja para leer el puntero a cajones de punteros a registro del 0,1,2,3,4) + 5*218 + 5*10.000= 51.091 bloq. Peor que la secuencial.
- Hash en carnet → Coste = 5*1 (caen en 5 cajones diferentes) + 5*218 + 5*10.000 = 51.095 bloq. Peor que la secuencial.

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

Número de registros a recuperar nrc = (10-1)*100.000/10 = 90.000 reg. Se pide todos menos el 8. El archivo no está ordenado, luego sacar todos los registros a partir del índice.

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- B+ en carnet → Coste = 1 (nodo raíz/hoja para leer el puntero a cajones de punteros a registro del 0,1,2,3,4,5,6,7,9) + 9*218 + 9*10.000= 91.936 bloq. Peor que la secuencial.

- Hash en carnet → Coste = 5*1 (caen en 5 cajones diferentes) + 9*218 + 9*10.000 = 91.940 bloq. Peor que la secuencial.

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

Número de registros a recuperar nrc = (9-8+1)*100.000/10 = 20.000 reg. Se pide el 8,9. El archivo no está ordenado, luego sacar todos los registros a partir del índice.

Cómo se puede buscar:

- Secuencial → Coste = 20.000 bloq
- B+ en carnet → Coste = 1 (nodo raíz/hoja para leer el puntero a cajones de punteros a registro del 8,9) + 2*218 + 2*10.000= 20.437 bloq. Peor que la secuencial.
- Hash en carnet → Coste = 2*1 (caen en 2 cajones diferentes) + 2*218 + 2*10.000 = 20.438 bloq. Peor que la secuencial.

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

Número de registros a recuperar nrc = 100.000/(100.000*10) = 0.1 reg. Será o ningún registro o 1 como máximo debido al campo carnet.

Como es un AND significa que la tabla se puede buscar por uno de ellos, el más barato, que como se ha visto antes es por carnet. Luego el coste sería como el visto en los apartados anteriores con condición de carnet = 2345. Si se busca por código de carrera al devolver más registros entonces su coste es mayor y además el archivo no está ordenado por ee campo. En este caso para Carnet el B+ da menor coste.

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

Número de registros a recuperar nrc = 100.000/100.000 + 10.000/10 = 10.001 reg. Es un OR sobre dos campos luego equivale a realizar dos búsquedas una por cada campo si no se usa la lectura secuencial.

Coste será:

- ➔ Coste de carnet=53456 que si se tienes un B+ es más barato para este problema con Coste =4+1 = 5 bloques. Obtendría 1 reg o 0 registros.
- ➔ Coste de código de carrera=9, El hash o el B+ en código de carrera porque su coste es: Coste = 1 + 218 + 10.000 = 10.219 bloq en este caso.
- ➔ La secuencial tiene un coste de 20.000 bloques, luego es más barato buscar por los dos índices: B+ por carnet y B+ o Hash por código de carerra teniendo un coste total de 5 + 1 + 218 + 10000 = 10.224 bloq