

Relación 3

1. Se dispone de una relación $R(a, b)$ donde a es la clave de valores únicos por la que se mantiene ordenado el archivo y b es un atributo con valores duplicados, además se tiene $B = 4096B$, $C = 10B$, $P = 10B$, $N(R) = 1000$, $V(R, b) = 200$, $\text{size}(a) = 10B$, $\text{size}(b) = 40B$. Se montan dos índices I_A e I_B , uno por cada atributo. Calcula el tamaño en bloques de cada índice.

En primer lugar, calcularemos el tamaño en bloques del índice construido sobre el atributo a .

El tamaño de cada entrada del índice sería:

$$L(I_A) = 10B + 10B = 20B$$

Para ver cuántos registros entran en un bloque del índice, calculamos el **factor de bloqueo**:

$$Bfr(I_A) = \text{parte entera}\left(\frac{B-C}{L(I_A)}\right) = \text{parte entera}\left(\frac{4096B - 10B}{20B}\right) = 204$$

El número de bloques necesario para almacenar las 1000 entradas de la relación en el índice se calcula:

$$B(I_A) = \text{redondeo hacia arriba}\left(\frac{N(R)}{Bfr(I_A)}\right) = \text{redondeo hacia arriba}\left(\frac{1000}{204}\right) = 5$$

El tamaño del índice I_A es de 5 bloques.

Para calcular el tamaño del índice construido sobre el atributo b , necesitamos hacer algunos cálculos adicionales.

Dado que un índice no puede tener valores repetidos, puesto que la búsqueda binaria no funcionaría, hemos de construir otro tipo de índice. En concreto, podríamos usar una estructura de índice multinivel en el que, el primer nivel (raíz) no tuviera valores repetidos, y el segundo nivel tuviera valores repetidos.

Siendo así, cada valor distinto del atributo b sólo podría aparecer una vez en el primer nivel. Para calcular cuántos bloques hacen falta para el primer nivel, tenemos que tener en cuenta el número de entradas:

$$B(I_B^1) = \text{redondeo hacia arriba}\left(\frac{N(I_B)}{Bfr(I_B)}\right) = \text{redondeo hacia arriba}\left(\frac{V(R, b)}{Bfr(I_B)}\right)$$

$$Bfr(I_B) = \text{parte entera}\left(\frac{B-C}{L(I_B)}\right) = \text{parte entera}\left(\frac{4096B - 10B}{L(I_B)}\right)$$

$$L(I_B) = 40B + 10B = 50B$$

Por lo que:

$$Bfr(I_B) = \text{parte entera}\left(\frac{4086B}{50B}\right) = 81$$

$$B(I_B^1) = \text{redondeo hacia arriba} \left(\frac{V(R, b)}{Bfr(I_B)} \right) = \text{redondeo hacia arriba} \left(\frac{200}{81} \right) = 3 \text{ bloques}$$

Véase que el número de entradas del índice de primer nivel sería el número de valores distintos para el atributo b en la relación R, y que la estructura de todos los índices es exactamente igual, sean de un sólo nivel o multinivel, por lo que el cálculo del factor de bloqueo para el índice I_B es igual que el calculado para el índice I_A .

Con respecto al segundo nivel del índice, cada bloque contendrá entradas con el mismo valor de clave que se recorrerán secuencialmente. En ese caso, habría que calcular cuántos bloques se estiman necesarios para contener las 1000 entradas distintas de la tabla agrupadas en tantos grupos como valores distintos haya del atributo b.

El número de bloques estimado para contener los registros con valores iguales de b se calcula:

$$\begin{aligned} \text{redondeo hacia arriba} \left(\frac{N(I_B^2)}{Bfr(I_B)} \right) &= \text{redondeo hacia arriba} \left(\frac{\frac{L(R)}{V(R, b)}}{Bfr(I_B)} \right) = \\ &= \text{redondeo hacia arriba} \left(\frac{\frac{1000}{200}}{81} \right) = \text{redondeo hacia arriba} \left(\frac{1000}{200 \cdot 81} \right) = 1 \end{aligned}$$

De este modo, para cada uno de los 200 valores de b, se estiman 5 registros que caben en un bloque. De ese modo, cada valor distinto tendría 1 bloque en el segundo nivel del índice, lo que hacen:

$$B(I_B^2) = 1 \text{ bloque / valor} \cdot 200 \text{ valores} = 200 \text{ bloques}$$

De este modo, el cálculo total de bloques para el índice I_B es:

$$B(I_B) = B(I_B^1) + B(I_B^2) = 3 + 200 = 203 \text{ bloques}$$

El tamaño del índice I_B es de 203 bloques.

2. ¿De qué depende el tiempo necesario para reorganizar un Archivo de Acceso Directo?

El tiempo necesario para reorganizar un archivo de acceso directo depende del tiempo necesario en recorrerlo, es decir, del tiempo que se tarda en leer las posiciones del fichero más las colisiones, y el tiempo necesario en generar un nuevo archivo de acceso directo, insertando de nuevo cada registro leído en un nuevo archivo de acceso directo (que dependerá del tiempo que se tarda en calcular la posición del registro y el tiempo que se tarda en insertarlo en una posición o en la zona de desbordamiento).

$$T_Y = T_X + T_{Carga}$$

$$T_X = (m + c) \cdot T$$

$$T_{Carga} = \sum_{i=1}^n T_I(i)$$

3. Sean las relaciones R y S con los siguientes parámetros:

R(a,b,c)	S(c,d)
N(R) = 5000	N(S)=200
V(R,a)=5000	
V(R,b)=3000	
V(R,c)=5	V(S,c)=5
	V(S,d)=40
Size(a)=20	
Size(b)=60	
Size(c)=20	Size(c)=20
	Size(d)=40

Teniendo en cuenta que el tamaño de bloque es de 2KB, que la cabecera es de 20B y que en memoria sólo cabe un bloque, determina el número de operaciones de E/S que supondría la ejecución de la consulta:

$$\Pi_{c,d}(\sigma_{d=d_1}(R \text{ JOIN } S))$$

Teniendo en cuenta que cada operación de E/S representa la transferencia de un bloque de datos entre el disco y la memoria, o viceversa, tendremos que calcular el número de bloques transferidos en total.

El número de bloques que almacenan la relación R es de:

$$B(R) = \text{redondeo hacia arriba} \left(\frac{N(R)}{Bfr(R)} \right)$$

Para ello, hay que calcular el factor de bloqueo de R como:

$$Bfr(R) = \text{parte entera} \left(\frac{B-C}{L(R)} \right)$$

que dependerá de la longitud del registro de R:

$$L(R) = 20B + 60B + 20B = 100B$$

De modo que:

$$Bfr(R) = \text{parte entera} \left(\frac{2048 - 20}{100} \right) = 20$$

$$B(R) = \text{redondeo hacia arriba} \left(\frac{5000}{20} \right) = 250 \text{ bloques}$$

Si aplicamos los mismos cálculos para el número de bloques de la relación S:

$$L(S) = 20B + 40B = 60B$$

$$Bfr(S) = \text{parte entera} \left(\frac{2048 - 20}{60} \right) = 33$$

$$B(S) = \text{redondeo hacia arriba} \left(\frac{200}{33} \right) = 7 \text{ bloques}$$

Como ya se sabe, hay varios algoritmos para resolver la operación de reunión natural, pero la más eficiente, consiste en ordenar por el atributo de reunión aquella relación que no lo está (sólo una de las dos). Entendemos que R no lo está por lo que tenemos que aplicar un algoritmo de ordenación que requiere realizar un número de lecturas de:

$$B(R) \cdot \log_2(B(R)) = 250 \cdot \log_2(250) = 1991,45 \approx 1992 \text{ bloques}$$

Ahora que las dos relaciones están ordenadas por el atributo c, podemos realizar la operación de reunión natural entre las dos. Habrá que leer:

$$250 + 7 = 257 \text{ bloques}$$

y escribir los bloques resultantes de la operación de reunión:

$$L(\text{JOIN}) = 20B + 60B + 20B + 40B = 140B$$

$$Bfr(\text{JOIN}) = \text{parte entera} \left(\frac{B-C}{L(\text{JOIN})} \right) = \text{parte entera} \left(\frac{2048-20}{140} \right) = 14$$

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{Bfr(\text{JOIN})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{14} \right)$$

El número de registros de la operación de reunión natural viene dado por:

$$N(\text{JOIN}) = \frac{N(R) \cdot N(S)}{\max\{V(R, c), V(S, c)\}} = \frac{5000 \cdot 200}{\max\{5, 5\}} = 200000$$

por lo que el número de bloques resultantes que tendremos que escribir en disco será de:

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{200000}{14} \right) = 14286 \text{ bloques}$$

A fin de aplicar la operación de selección sobre el atributo d, tendremos que leer de nuevo:

$$14286 \text{ bloques}$$

y el número de bloques que se transferirán al disco como resultado de la selección será de:

$$B(\text{SELEC}) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{SELEC})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{JOIN})} \right)$$

$$N(\text{SELEC}) = \frac{N(\text{JOIN})}{V(S, d)} = \frac{200000}{40} = 5000$$

por lo que:

$$B(\text{SELEC}) = \text{redondeo hacia arriba} \left(\frac{5000}{14} \right) = 358 \text{ bloques}$$

Por último, para realizar la operación de proyección, tendremos que leer:

$$358 \text{ bloques}$$

y escribir el número de bloques resultantes de dicha operación de proyección:

$$B(\text{PROY}) = \text{redondeo hacia arriba} \left(\frac{N(\text{PROY})}{Bfr(\text{PROY})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{PROY})} \right)$$

$$Bfr(\text{PROY}) = \text{parte entera} \left(\frac{B-C}{L(\text{PROY})} \right) = \text{parte entera} \left(\frac{2048-20}{L(\text{PROY})} \right)$$

$$L(\text{PROY}) = 20B + 40B = 60B$$

Con lo que:

$$Bfr(\text{PROY}) = \text{parte entera} \left(\frac{2048-20}{L(\text{PROY})} \right) = \text{parte entera} \left(\frac{2048-20}{60} \right) = 33$$

$$B(\text{PROY}) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{PROY})} \right) = \text{redondeo hacia arriba} \left(\frac{5000}{33} \right) = 152 \text{ bloques}$$

De modo que el número total de lecturas y escrituras de bloques necesarios para resolver este plan físico es de:

$$1992 + 250 + 7 + 14286 \cdot 2 + 358 \cdot 2 + 152 = 31689 \text{ bloques}$$

4. Propón un plan lógico para la consulta del ejercicio anterior cuyo plan físico sea más eficiente que el calculado para dicho ejercicio. Justifica numéricamente tu respuesta.
-

En primer lugar, llevaremos la selección lo más cerca posible de las hojas del árbol de consulta:

$$\Pi_{c,d}(R JOIN \sigma_{d=d_1}(S))$$

A continuación, hundiremos la proyección lo más posible hasta las hojas para eliminar atributos que no deben ser considerados para la resolución de la consulta:

$$\Pi_c(R) JOIN \sigma_{d=d_1}(\Pi_{c,d}(S))$$

Sin embargo y teniendo en cuenta que el esquema de la relación S es, precisamente, c y d, la proyección sobre S no tiene sentido, por lo que el plan lógico consulta sería:

$$\Pi_c(R) JOIN \sigma_{d=d_1}(S)$$

Para calcular si el número de bloques considerados para la ejecución de esta consulta (plan físico) es mejor que el calculado para el ejercicio 3, calculamos dichos bloques.

En primer lugar, el número de bloques para la selección sobre S es:

$$B(SELEC) = \text{redondeo hacia arriba} \left(\frac{N(SELEC)}{Bfr(SELEC)} \right) = \text{redondeo hacia arriba} \left(\frac{N(SELEC)}{Bfr(S)} \right)$$

$$N(SELEC) = \frac{N(S)}{V(S, d)} = \frac{200}{40} = 5$$

Por lo que:

$$B(SELEC) = \text{redondeo hacia arriba} \left(\frac{N(SELEC)}{Bfr(S)} \right) = \text{redondeo hacia arriba} \left(\frac{5}{33} \right) = 1$$

Para poder realizar la selección es necesario leer los bloques de S y escribir el bloque resultante:

$$7 + 1 = 8 \text{ bloques}$$

Para poder realizar la proyección sobre R es necesario leer los 250 bloques y escribir el número de bloques necesarios para almacenar el resultado:

$$B(PROY) = \text{redondeo hacia arriba} \left(\frac{N(PROY)}{Bfr(PROY)} \right) = \text{redondeo hacia arriba} \left(\frac{N(R)}{Bfr(PROY)} \right)$$

$$Bfr(PROY) = \text{parte entera} \left(\frac{B-C}{L(PROY)} \right) = \text{parte entera} \left(\frac{2048-20}{L(PROY)} \right)$$

$$L(PROY) = 20 B$$

Con lo que:

$$Bfr(PROY) = \text{parte entera} \left(\frac{2048-20}{L(PROY)} \right) = \text{parte entera} \left(\frac{2048-20}{20} \right) = 101$$

$$B(\text{PROY}) = \text{redondeo hacia arriba} \left(\frac{N(R)}{Bfr(\text{PROY})} \right) = \text{redondeo hacia arriba} \left(\frac{5000}{101} \right) = 50 \text{ bloques}$$

Es decir, para calcular la proyección sobre R es necesario leer los bloques R y escribir los bloques del resultado. En total:

$$250 + 50 = 300 \text{ bloques}$$

Para resolver la reunión natural, será necesario ordenar la primera relación (el resultado de la proyección sobre R) por el atributo c para lo cual habrá que leer:

$$50 \cdot \log_2(50) = 282.2 \approx 283 \text{ bloques leídos y escritos para ordenar}$$

Por último, leemos los bloques ordenados de la proyección sobre R junto con los bloques de la selección sobre S, y los mezclamos. El número de bloques del resultado de la reunión natural sería:

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{Bfr(\text{JOIN})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{Bfr(S)} \right)$$

Hay que ver que el esquema resultante de la reunión natural es exactamente igual que el esquema de la relación S, por lo que nos ahorramos algunos cálculos ya realizados.

$$N(\text{JOIN}) = \frac{N(\text{PROY}) \cdot N(\text{SELEC})}{\max\{V(R, c), V(S, c)\}} = \frac{5000 \cdot 5}{\max\{5, 5\}} = 5000$$

con lo que:

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{Bfr(S)} \right) = \text{redondeo hacia arriba} \left(\frac{5000}{33} \right) = 152 \text{ bloques}$$

En resumen, el número de bloques necesarios para ejecutar este plan físico sería:

$$7 + 1 + 250 + 50 + 283 + 50 + 1 + 152 = 794 \text{ bloques}$$

Efectivamente, el número de bloques empleados en esta aproximación es mucho menor de los 31939 bloques del plan físico del ejercicio anterior.

5. Explica qué elementos del nivel interno de Oracle se crearían o modificarían y en qué *tablespaces* para obtener un *cluster* completamente funcional (en el que se puede insertar y consultar) que incluye dos tablas R(a,b,c) y S(c,d).
-

Por una parte, para almacenar las estructuras del clúster, se crearía un segmento, en el que se crearía una extensión, en el que crearía un bloque para contener los registros de las dos tablas del clúster. Estas estructuras se crearían en el tablespace del usuario (*USERS*, en su caso).

Además, se modificaría la tabla o tablas del tablespace *SYSTEM* que contienen la parte del catálogo referente a los clústers, tablas, restricciones sobre columnas y tablas, e índices, consultables por las vistas de catálogo:

- DBA_OBJECTS
- DBA_CLUSTERS
- DBA_TABLES
- DBA_TAB_COLUMNS
- DBA_CONSTRAINTS
- DBA_INDEXES
- ...

6. Propón un tipo de bloque y una estructura de bloque similar a la provista por Oracle para almacenar los registros del clúster del ejercicio anterior en el *tablespace* de datos.
-

El bloque necesario para almacenar esta información sería un bloque con bloqueo fijo y estructura heterogénea y contendría, al igual que un bloque de Oracle:

- cabecera
- directorio de registros
- espacio libre
- grupos de registros formados por:
 - un registro que contienen valores de los atributos comunes del cluster
 - registros de la primera tabla con dichos valores en los atributos comunes
 - registros de la segunda tabla con dichos valores en los atributos comunes