



RELACIÓN 1

3. 3. Indica las ventajas e inconvenientes de tener registros de longitud variable y razona las respuestas

Ventajas • Permiten almacenar varios tipos de registro en un mismo archivo. • Permite tener tipos de datos con longitud variable como varchar. • Permite un mayor aprovechamiento de los bloques ya que se pueden ajustar los registros mejor al tamaño del bloque e incluso dividir el último, si no cabe en un bloque. • Es la única posibilidad cuando el tamaño del registro excede el tamaño del bloque.

Inconvenientes

Se ha de almacenar la estructura de los registros en las cabeceras de los bloques o en cada uno de los registros con lo que se pierde espacio de almacenamiento. • Algunos registros quedarán divididos en varios bloques al no poderse introducir un registro en el espacio libre del bloque donde deba insertarse. • Requiere el uso de marcas (delimitadores de campos) para indicar donde termina un campo y empieza el siguiente para delimitarlos. • Presenta problemas en la modificación si el registro resultante es de mayor tamaño que el original pudiendo obligar a relocalizar el registro. • No permiten una ubicación inmediata (búsquedas) dentro de los archivos (y de los bloques que lo componen) de almacenamiento debido a que las longitudes de los registros no son siempre similares. • Por la misma razón, la extracción de datos no es inmediata.

4. Indica las ventajas y/o los inconvenientes de claves de búsquedas duplicadas a la hora de montar un índice, y razona las respuestas.

Ventajas: • Índice de poco tamaño y ayuda a la búsqueda del primer elemento con dicha clave. • Permite la búsqueda por varias claves que no sean la clave primaria. • Va muy bien para búsquedas por rangos, es decir, la búsqueda de un intervalo de valores en un atributo.

Inconvenientes: • El primer dato es de acceso directo (acceso mediante el índice) y el resto de los datos del índice se acceden de forma secuencial. Al estar éstos duplicados, se accede al primero de los índices y al resto se van recorriendo de forma secuencial. • No se garantiza el orden al insertar, al tener tanta clave repetida uno no sabe si ese registro se insertará antes o después de otro registro con la misma clave.

5. Indica cuándo crees que es más adecuado usar el bloqueo partido.

(e) cuando hay relación entre el tamaño del registro y el tamaño del bloque.

De otro modo, tendríamos mucho espacio desperdiciado en cada bloque a no ser que el tamaño coincidiera, pero esto no es posible.

Con registros de gran tamaño ocurre lo mismo, es muy probable que el espacio al final del bloque no

nos permita alojar nuestro registro completo, por lo que si usamos bloqueo partido, el desperdicio será menor. Por otro lado, registros de tamaño pequeño no será muy útil pues seguro que al final del bloque entra un último registro y almacenar todo lo necesario para usar bloqueo partido puede no ser útil para el espacio restante de un hipotético último registro que no entrara al final de un bloque. La misma deducción pero a la inversa podemos aplicar para los tamaños de bloque.

otra respuesta

Por descarte, podemos decir que es la "e", ya que en bloque partido si el registro no cabe se guarda en bloques distintos por lo que la opción "a: para guardar registros de gran tamaño", no sería cierta ya que la búsqueda sería difícil. La opción "b: por registros de tamaño pequeño", no sería cierta ya que si son muy pequeños, caben demasiados registros, lo que sería lento. La "c: por bloques de más tamaño" y la "d: por bloques de tamaño pequeño", tampoco son válidas ya que el tamaño del bloque depende del registro.

8. Se tienen registros con un nombre que es un varchar(29), una dirección que es un varchar(255), una fecha que ocupa 10B, un valor para sexo que es un lógico y ocupa 1B, y un tamaño de bloque B=4KB. Calcula el factor de bloqueo y el porcentaje de utilización en caso de tratarse de bloque fijo, si el bloque contiene 10B de cabecera y un directorio de entradas en el bloque.

Los registros están constituidos por los siguientes campos:

nombre --> varchar(29): 29B dirección --> varchar(255): 255B fecha: 10B sexo: 1B

En primer lugar, calculamos el tamaño medio del registro. Como tenemos atributos de tipo **varchar**, esto nos lleva a tener que usar registros de longitud variable en bloques homogéneos de tamaño 4KB=4096B, según el enunciado. En este tipo de bloques, como la cabecera puede almacenar la estructura del registro, no tenemos que incluir los identificadores de atributos ni tampoco los separadores entre identificadores de atributo y valor. Es por ello que en la expresión para el cálculo del tamaño medio del registro, el tamaño medio de los identificadores de atributo vale 0. Además, como cada valor va acompañado de un terminador de valor, tenemos que contabilizarlo:

$$R = a \cdot (A + V + 1) = 4 \cdot (0 + \frac{29 + 255 + 10 + 1}{4} + 1) = 299 \text{ B}$$

Para calcular el **factor de bloqueo** tenemos que tener en cuenta el tamaño del bloque, el tamaño de la cabecera y el tamaño del directorio y el tamaño de cada registro más el terminador. El directorio estará formado por el valor del campo que actúe como identificador, que en nuestro caso es el nombre (29B).

$$Bfr = \frac{B - C - 29 \cdot Bfr}{R + M} = \frac{4096 - 10 - 29 \cdot Bfr}{299 + 1} \Rightarrow Bfr = \frac{4086}{329} = 12.41945289 \Rightarrow Bfr = 12$$

Luego, esto nos dice que se pueden almacenar 12 registros/bloque.

Calculemos ahora el porcentaje de utilización:

$$\%util = \frac{Bfr \cdot R}{B-C-29 \cdot Bfr} \cdot 100 = \frac{12 \cdot 299}{4096-10-12 \cdot 29} \cdot 100 = 95.98715891 \% \approx 96\%$$

9. Se tienen registros con: char(215), integer -2B-, fecha -10B-, real -8B-, R=235B, B=4KB. Supuesta la estructura de longitud variable, una cabecera con 2 punteros -de 4B- más un carácter, calcula el factor de bloqueo para: a) bloqueo fijo b) bloqueo encadenado

Observemos que ninguno de los campos de los registros es de longitud variable. En consecuencia, se usan registros de longitud fija.

Tamaño de bloque = 4KB = 4096B **Tamaño de cabecera** = $2 \cdot 4 + 1 = 9B$

- Factor de bloqueo
 - Bloqueo fijo: $Bfr = \lfloor \frac{B-C}{R} \rfloor = \lfloor \frac{4096-9}{235} \rfloor = 17$
 - Bloqueo encadenado: tenemos en cuenta que hay un puntero de 4B que referencia al siguiente bloque.

$$Bfr = \lfloor \frac{B-C-P}{R} \rfloor = \lfloor \frac{4096-9-4}{235} \rfloor = 17$$

10. Supongamos una relación con 10^6 tuplas, un tamaño de bloque B=4KB, un factor de bloqueo BFr=10, V=10B y P=8B. Calcula el tiempo de búsqueda TF en un índice denso primario si estuviese en memoria. Y supuesto que no cabe en memoria y se monta un índice de segundo nivel, calcula el espacio adicional ocupado.

En un índice denso, el tiempo de búsqueda T_F viene dado por la expresión:

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

donde n es el número de tuplas de la relación y T es el tiempo de acceso a memoria. Este último no nos lo dan, luego, la expresión del tiempo de búsqueda vendrá dado en función de T :

$$T_F = \log_2(n) \cdot T + T = 20.93156857 \cdot T \approx 21 \cdot T$$

En caso de que montemos un índice secundario, el número de entradas del mismo coincidirá con el número de bloques que ocupa el índice denso de primer nivel. Determinemos entonces el factor de bloqueo de este último:

$$Bfr = \lfloor \frac{B-C}{R} \rfloor = \lfloor \frac{4096-0}{18} \rfloor = 227$$

En consecuencia, el número de bloques del índice de primer nivel será:

$$\lceil \frac{n}{Bfr} \rceil = \lceil \frac{10^6}{227} \rceil = 4406 \text{ bloques}$$

Ambos índices tienen el mismo factor de bloqueo. Es por ello que el número de bloques del índice de

segundo nivel tiene que ser:

$$\lceil \frac{n}{Bfr} \rceil = \lceil \frac{4406}{227} \rceil = 20 \text{ bloques}$$

11. Supongamos $Bfr=30$ del fichero de datos, y $Bfrl=100$ del índice. Sean n los registros de datos. Indica cuántos bloques se necesitan para: a) un índice denso b) un índice no denso. Supuesta una ocupación del 100%, vuelve a calcular los casos a) y b) suponiendo que los bloques se ocupan inicialmente al 80%.

Con el 100% de ocupación:

- **Índice denso** En los índices densos el número de registros del fichero de índice coincide con el número de registros del fichero maestro. En este caso estamos hablando de n entradas. Luego, el número de bloques que se necesitan para almacenarlo es:

$$\lceil \frac{n_{\{I\}}}{Bfr_{\{I\}}} \rceil = \lceil \frac{n}{100} \rceil \text{ bloques}$$

- **índice no denso**

Los índices no densos tienen tantas entradas como bloques ocupa el fichero maestro (fichero de datos). Luego, el número de bloques que se necesitan para almacenarlo es:

$$\lceil \frac{n}{Bfr} \rceil = \lceil \frac{n}{30} \rceil \text{ bloques}$$

El factor de bloqueo del índice no denso es el mismo que el del índice denso, y tiene $\frac{n}{30}$ entradas, luego tenemos que el número de bloques que se necesitan para almacenarlo es:

$$\lceil \frac{n_{\{I\}}}{Bfr_{\{I\}}} \rceil = \lceil \frac{\frac{n}{30}}{100} \rceil = \lceil \frac{n}{3000} \rceil \text{ bloques}$$

Con el 80% de ocupación:

- **Índice denso** Ahora, aunque el número de entradas es el mismo, el factor de bloqueo se ha reducido a 80:

$$\lceil \frac{n_{\{I\}}}{Bfr_{\{I\}}} \rceil = \lceil \frac{n}{100 \cdot \frac{80}{100}} \rceil = \lceil \frac{n}{80} \rceil \text{ bloques}$$

- **Índice no denso** Como cambia el factor de bloqueo del fichero maestro ($Bfr = 30 \cdot \frac{80}{100} = 24$), cambia también el número de entradas del índice. Es por ello que tenemos que calcular el número de bloques que se necesitan para almacenar el fichero maestro con el nuevo factor de bloqueo:

$$\lceil \frac{n}{Bfr} \rceil = \lceil \frac{n}{24} \rceil \text{ bloques}$$

El factor de bloqueo del índice no denso es igual que el del denso y como tiene $\frac{n}{24}$, tenemos que el número de bloques necesarios para almacenarlo es:

$$\lceil \frac{n}{\lceil \frac{n}{24} \rceil} \rceil = \lceil \frac{n}{80} \rceil = \lceil \frac{n}{1920} \rceil \text{ bloques}$$

Conclusiones que saco de este ejercicio:

Índices densos

Son índices que tienen tantas entradas como registros tiene el fichero maestro o fichero de datos.

Supongamos que el fichero maestro tiene n entradas (registros de datos). Entonces el índice denso tendrá n entradas también. El factor de bloqueo es el número de registros que caben en un bloque del fichero. Luego, si tengo n registros y se que caben B registros en un bloque del índice, por una regla de tres sencilla deduzco que el número de bloques necesarios para almacenar un índice denso es:

$$\lceil \frac{n}{B} \rceil$$

Índices no densos

Són índices que tienen tantas entradas como bloques ocupa el fichero maestro (fichero de datos), luego, en primer lugar, calculamos el número de bloques que se necesitan para almacenar el fichero maestro:

$$\lceil \frac{n}{B} \rceil \text{ bloques}$$

Ahora, como ya sabemos el número de entradas que va a tener el índice no denso (coincide con el número de bloques del maestro), como tanto índice denso como no denso tienen el mismo factor de bloqueo, el número de bloques necesarios para almacenarlo es:

$$\lceil \frac{\lceil \frac{n}{B} \rceil}{B} \rceil \text{ bloques}$$

13. Supón el siguiente esquema: dpto (d#, nombre, extension, dir) y prof (NRP, nombre, categoría, dpto). Indique la organización que favorecería más la siguiente consulta: SELECT depto, categoria, count(*) FROM prof GROUP BY depto, categoría;

a) un índice denso sobre dpto de prof b) un índice no denso sobre dpto de prof c) un índice denso sobre dpto y categoría de prof d) un hashing sobre dpto y categoría de prof e) un índice denso sobre d# de dpto f) ninguno

Si observamos la consulta, vemos que tenemos que devolver el departamento y la categoría de la tabla prof, además de el número de el número de tuplas con esos valores. Ambos atributos están en la tabla prof y como la consulta se realiza en el orden **dpto, categoría**, la organización que favorecería más la consulta es **un índice denso sobre dpto y categoría de prof**. Este índice estará ordenado por esos valores. De esta forma, en caso de haber registros con los mismos valores, estarán uno de tras de otro en

el índice. Esto favorece al count(*).

14. Suponiendo que se han definido como claves primarias d# y NRP, revisa la consulta anterior y justifica la respuesta

Como en la consulta no se están utilizando dichos atributos, luego no se utilizan los índices que tienen asociados. Es por ello que que sean claves primarias no afecta en nada a la consulta.

15. Para los planes lógicos del ejercicio anterior y si se dispone de las siguientes estadísticas sobre los atributos A, B y C: $T(R)=1000$, $V(R,A)=75$, $V(R,B)=20$, $V(R,C)=80$, indica cuál de ellos escogería un optimizador para ejecutar la consulta. Justifica tu respuesta
16. Indica las diferencias entre plan lógico y plan físico.

Un plan lógico representa una de las posibles secuencias que se pueden seguir para ejecutar una consulta y un plan físico es un plan lógico con el coste asociado a llevar a cabo dicho plan.