PREGUNTAS TIPO TEST: PARCIAL T4

1. Conviene que estén relacionados el tamaño de los bloques físicos y de las páginas para mejorar el rendimiento del sistema de almacenamiento: **Verdadero (conviene que las páginas tengan un tamaño múltiplo del tamaño del bloque, así las operaciones involucran la mínima cantidad de páginas, una página = mínimo 1 op. E/S)**
2. El gestor de disco forma parte del SGBD: **Falso**
3. El rendimiento de un índice no denso desciende considerablemente cuando se realizan inserciones o borrados: **Falso**
4. El índice no denso es mucho menor que el denso cuando caben varios registros en un bloque: **Verdadero**
5. El índice denso no es rentable cuando se actualiza o se inserta con mucha frecuencia: **Verdadero**
6. El nivel interno de una base de datos está enteramente gestionado por el SO del ordenador: **Falso**
7. El orden de un árbol está determinado por el tamaño de la página que se asigna a los nodos de árbol: **Verdadero**
8. El índice no denso permite realizar preguntas de tipo existencial sin acceder al fichero de datos: **Falso (puede resolverse directamente en un índice denso)**
9. El orden de un árbol B fija el número de punteros que salen de un nodo: **Verdadero**
10. El índice denso es adecuado para consultas por rangos de valores del campo clave: **Verdadero**
11. La organización multilistas puede servir para conectar ficheros y es la base de los modelos de datos basados en grafos: **Verdadero**
12. El índice no denso es el único mecanismo de indexación posible cuando los datos están ordenados físicamente: **Falso**
13. El orden de un árbol influye directamente en el número de niveles: **Verdadero**
14. En el hashing extendido lo mejor es que la pseudollave se ajuste al tamaño del índice que se guarda en memoria: **Verdadero**
15. Se puede combinar una organización multilistas con un árbol B para gestionar los accesos en una estructura jerárquica: **Verdadero**
16. El acceso directo a registros garantiza siempre que encuentro una tupla con una sola lectura de bloque: **Falso**
17. La organización multilistas es independiente de las técnicas de acceso al fichero: **Verdadero**
18. Los árboles B se montan en memoria para no tener que acceder a disco más que una vez para llegar a un registro: **Falso**
19. Se puede montar un árbol B sobre cualquier campo clave utilizando un índice denso como conjunto secuencia: **Verdadero**
20. En el nivel interno de una base de datos hay que tener en cuenta también el nivel físico que gestiona el acceso a disco: **Verdadero**

21. En hashing dinámico hace falta una estimación del número de datos a insertar para dimensionar la tabla hash: **Verdadero (tb necesario en el hashing estático)**
22. El hashing dinámico es el método de acceso que mejor distribuye los datos en disco y, por tanto, el que menos desperdicio ocasiona: **Verdadero**
23. Puesto que son formas de índice no denso sólo se puede montar un árbol B sobre la clave física de un archivo: **Falso (no es una variante de índice no denso)**
24. El hashing dinámico es muy eficaz porque la tabla hash va en memoria principal: **Falso**
25. El acceso directo a bloques o cubos produce menos lecturas en disco que el acceso directo a registros: **Verdadero (no todos los registros están reflejados en el índice)**
26. La actualización de los archivos puede no influir en la actualización de los índices no densos: **Verdadero**
27. Las sentencias CREATE TABLE y CREATE INDEX de SQL generan nuevos conjuntos de páginas (archivos almacenados) en el nivel interno: **Verdadero**
28. En ficheros no ordenados físicamente, no se pueden montar índices no densos: **Verdadero**
29. El número de niveles de un árbol B depende sólo del tamaño del fichero base: **Falso**
30. En un índice multinivel el índice de primer nivel (nodo hoja) puede ser denso o no denso: **Verdadero**
31. El acceso directo a registros no permite realizar la lectura secuencial de datos en un rango: **Verdadero (sólo es adecuado para consultar por valor y los registros no están ordenados por su clave física)**
32. Se puede combinar una organización multilistas con un hashing extendido para gestionar los accesos en una estructura de datos jerárquica: **Verdadero**
33. Se pueden montar tantos índices densos como se necesiten: **Verdadero (solo se puede montar un índice no denso)**
34. Las páginas que componen un archivo almacenado no tienen por qué estar consecutivas en disco: **Verdadero**
35. En el hashing extendido una mala elección en el tamaño de las páginas puede obligar a reorganizar completamente la estructura: **Verdadero**
36. Lo normal es que cada archivo almacenado del nivel interno se almacene en un fichero físico separado: **Falso**

37. El número de accesos a disco que hacen falta para obtener una página dependerá del tamaño de la página y del tamaño del boque físico: **Verdadero**
38. El índice no denso mejora el barrido ordenado completo del fichero por la clave física: **Falso (en todo caso el índice denso)**
39. En el hashing extendido lo mejor es que la pseudollave tenga muchos dígitos: **Falso (muchos digitos = cubos iniciales muy pequeños, mejor pocos dígitos)**
40. En el agrupamiento interarchivo se ubican en la misma página registros de distinto tipo: **Verdadero**

41. Cuando la clave de un índice es compuesta (c1,c2) resulta eficiente el uso del índice para buscar por c1 o por c2: **Falso**
42. El índice denso ocupa el mismo tamaño que el propio fichero al que indexa: **Falso**
43. Todas las páginas de una BD tienen la misma estructura: **Falso ¿?**
44. Conocido el RID de un registro, no hace falta más que un acceso a disco para recuperarlo: **Verdadero**
45. En el acceso directo a registros, si se produce una colisión, habrá un hueco en el fichero maestro que nunca se va a aprovechar: **Verdadero**
46. Un factor de bloqueo mayor que 1 implica tener más de un registro por página: **Falso (no implica, ya que un factor > 1 indica que puede tener más de un registro, pero no por eso debe de tener más de un registro, es más, puede estar vacío)**
47. Los bloques usados para almacenar los datos de la BD pueden ser de distinto tamaño dependiendo del tamaño de los registros que se almacenan en ellos: **Falso**
48. Se pueden montar tantos índices no densos como sea necesario: **Falso**
49. Para montar un índice denso, los ficheros tienen que estar ordenados físicamente por algún campo: **Verdadero**
50. El mantenimiento de un índice no denso es menos costoso que el de un índice denso: **Verdadero (las operaciones de mantenimiento son menos frecuentes en un índice no denso, al no encontrarse todos los registros relajados)**
51. En una organización secuencial no es necesario que los registros mantengan ningún orden en particular: **Falso (registros almacenados consecutivamente por clave física)**
52. En una estructura hash dinámica, al insertar un elemento que supere el tamaño de un cubo, siempre se producirá desbordamiento tanto en ese cubo como en el directorio: **Falso ¿? (hay reorganización y desdoblamiento de cubo/directorio, cuando se llena un cubo se divide en 2)**
53. Las consultas basadas en OR sobre campos indizados mediante índices Bitmaps, obtienen las tuplas que satisfacen la condición directamente de los índices: **Verdadero ¿? (pone que es eficiente para consultas con predicado OR)**
54. El agrupamiento por defecto en el nivel interno es intraarchivos: **Verdadero**
55. Si sé cuántos registros va a tener el archivo almacenado y cuántos valores distintos de la clave, puedo dimensionar adecuadamente el acceso directo a cubos: **Verdadero (hashing estático, necesito conocer la distribución previa de la clave)**
56. El administrador de la BD puede decidir la forma de agrupamiento en páginas de los archivos que corresponden a las tablas de una BD: **Verdadero**
57. En un archivo almacenado puede haber más de un índice primario: **Falso ¿?**
58. En el hashing extendido no se producen desbordamientos: **Falso**
59. La clave de una tabla organizada por índice puede estar definida sobre cualesquiera de sus campos: **Verdadero**
60. No se pueden resolver consultas basadas en AND sobre dos campos indizados mediante índices Bitmaps, usando estos índices: **Falso ¿?**

61. CREATE TABLE y CREATE INDEX generan nuevos conjuntos de páginas en los archivos que corresponden a las tablas de una BD: **Verdadero**
62. La tabla de una tabla organizada por índice (iot) puede estar definida sobre cualquiera de sus campos: **Falso ¿? (clave física)**

63. El objetivo principal de los mecanismos de indexación y métodos de acceso es:
    a. Acceder a los datos de una tabla de forma ordenada
    b. Poder listar los datos por rangos en cualquiera de sus atributos
    c. Localizar datos requeridos con el número mínimo de operaciones de lectura en disco **VERDADERO**
    d. Garantizar que no se duplique la clave primaria
64. La sentencia CREATE TABLE provoca:
    a. La creación de un nuevo fichero en disco
    b. La creación de un nuevo archivo almacenado **VERDADERO**
    c. La inserción de una nueva página en un archivo almacenado existente
    d. La inserción de un registro en una página
65. El cluster:
    a. Acelera las consultas que involucran la reunión natural de las tablas que contiene **V**
    b. Perjudica la lectura individual de las tablas que contiene **V**
    c. Es una estructura interarchivo **V**
    d. Todo lo anterior es cierto **VERDADERO**
66. En un índice denso:
    a. El número de registros del índice es igual al número de registros de la tabla indexada **VERDADERO**
    b. El número de registros del índice es menor que el número de registros de la tabla indexada
    c. El índice ocupa lo mismo que la tabla indexada
    d. Ninguna de las anteriores es cierta
67. Indica la afirmación verdadera:
    a. El índice no denso es el único mecanismo de indexación posible para los datos que están ordenados físicamente
    b. Un árbol B sólo se puede montar sobre la clave física de un archivo
    c. Un fichero no ordenado físicamente, se pueden montar índices no densos
    d. Se pueden montar tantos índices densos como se necesite **VERDADERO**
68. La técnica de acceso directo:
    a. No permite realizar la lectura secuencial de datos de un rango **VERDADERO**
    b. Garantiza siempre que encuentro una fila con una sola lectura de bloque
    c. No utiliza área de desbordamiento
    d. No necesita una estimación previa del número de registros
69. En la indexación con árboles B:
    a. El orden de un árbol está determinado por el tamaño de página/bloque que se asigna a los nodos **V**

b. Se puede montar un árbol B sobre cualquier campo **V**
c. El orden de un árbol influye directamente en el número de niveles **V**
d. Todo lo anterior es cierto **VERDADERO**

70. El hashing dinámico, si el número de registros por bloque es 4 y tengo alrededor de 1000 registros, el número de bits necesario para la tabla hash es:
    a. 4
    b. 8 **VERDADERO**

    nºreg/bloque = 4 entonces 1000/4=250 -> 2^8 = 256

    1000 registros

    c. 16
    d. 5
71. Un índice no denso:
    a. Es adecuado para consultas por rango de valores del campo clave
    b. Permite realizar preguntas de tipo existencial sin acceder al fichero de datos
    c. Exige que los registros estén ordenados físicamente **VERDADERO (porque si la cota superior del rango está en un bloque y la cota inferior está en otro bloque distinto sería ineficiente)**
    d. El rendimiento desciende considerablemente cuando se realizan muchas inserciones o borrados en la tabla
72. Las páginas que componen un archivo almacenado:
    a. Tienen que estar consecutivas en disco
    b. Pueden ser de distinto tamaño dependiendo del tamaño de los registros que se almacenan en ellas
    c. Contienen siempre registros de una misma tabla
    d. Todo lo anterior es falso **VERDADERO**
73. El índice invertido:
    a. Es útil para recorrer una tabla por el campo clave en orden inverso al establecido **F ¿?**
    b. Se puede montar sobre cualquier campo de la tabla **V ¿?**
    c. Ayuda a distribuir mejor los datos en el espacio de almacenamiento **VERDADERO ¿?**
    d. Todo lo anterior es cierto **¿?**
74. Cuando la cardinalidad del campo por el que se indexa una tabla es muy baja, el mejor mecanismo de indexación es:
    a. Un índice de mapa de bits **VERDADERO**
    b. Un árbol B donde el conjunto secuencia (nodos hoja) sea denso
    c. Un árbol B donde el conjunto secuencia sea no denso
    d. Algún mecanismo de acceso directo
75. Con la consulta Select Codpro, sum(cantidad) From Ventas Group By Codpro:
    a. No se puede crear una vista por estar agrupada
    b. Se puede crear una vista, pero no será actualizable **VERDADERO (casi nunca son actualizables. Para que una vista sea actualizable tienen que aparecer en la consulta todos los atributos que sean Not Null (cp),**

no puede haber productos cartesianos, ni group by o having, no funciones de agregación, no distinct, no subconsultas...)

c. Se puede crear una vista y será actualizable porque sólo usa una tabla
d. Ninguna de las anteriores es cierta

76. Sean F y D las tablas procedentes de una entidad fuerte y una débil, respectivamente. Las filas de D se recuperarán siempre unidas con las de F, y rara vez, por separado. La mejor opción sería:
    a. Crear una vista en la que aparezcan los datos de ambas tablas en el formato adecuado
    b. Indexar D por el atributo que tiene en común con F
    c. Poner en D como clave externa el atributo que tiene en común con F
    d. Almacenarlas conjuntamente en un cluster **VERDADERO**
77. Cuando se necesita acceder a la tabla Alumnos por rangos de Notas, el mejor mecanismo es:
    a. El hashing básico
    b. Un índice no denso
    c. Un índice denso **VERDADERO**
    d. Un índice de mapa de bits

78. Sean las siguientes tablas:

Artículo(codar#, doi, título, ncitas, nreferencias)

Autor(codau#, nombre, email, dirección)

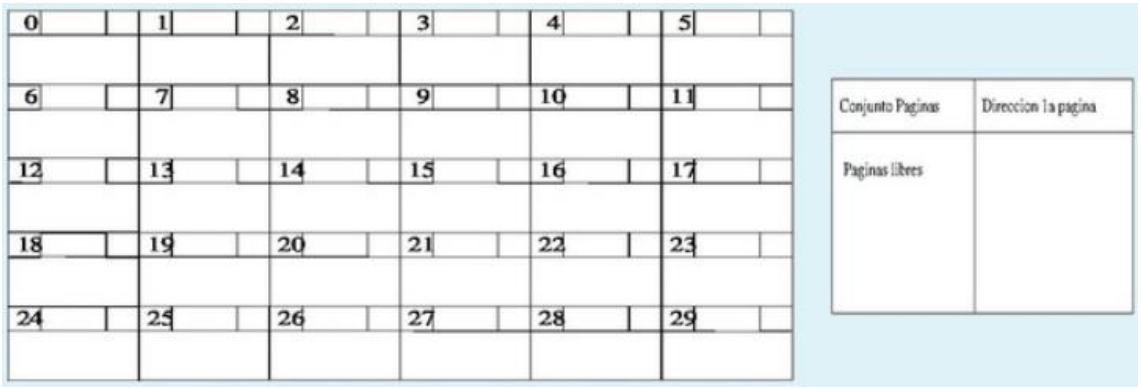
Escrito(codau#,codar#)

las páginas vinculadas a los archivos almacenados de las dos primeras tablas tienen un factor de bloqueo de 1, esto es, cabe un registro por página, mientras que las páginas vinculadas al archivo almacenado de la tabla Escrito tienen factor de bloqueo de 2 (dos registros almacenados por página). La estructura de páginas almacenadas y la estructura de la página 0 es como sigue (donde el recuadro de la esquina superior derecha de cada página albergará el número de página siguiente en la secuencia):

1. Supongamos que se inserta la siguiente secuencia de registros almacenados en Autor: (A01), (A02), (A04), (A05), (A16) y (A17), después la siguiente secuencia de registros almacenados en Artículo: (R01), (R02), (R03) y (R04) y, finalmente, la siguiente secuencia de registros almacenados en Escrito: (A01,R02), (A02,R02), (A04,R02), (A05,R02).

   Seleccione cuáles serán los valores correctos para las siguientes páginas después de la mencionada secuencia de inserciones:
   - Página 0: Página de comienzo de la secuencia de páginas de “Autor”: 1, de “Artículo”: 7, de “Escrito”: 11 y de “Páginas libres”: 13.
   - Página 5: Contenido: A16, puntero a la siguiente página: 6.
   - Página 8: Contenido: R02, puntero a la siguiente página: 9.
   - Página 11: Contenido: (A01,R02) y (A02,R02) , puntero a la siguiente página: 12 .
   - Página 12: Contenido: (A04,R02) y (A05,R02), puntero a la siguiente página: null.
   - Página 13: Contenido: vacía, puntero a la siguiente página: 14 .

79. Partiendo del contenido recogido en la estructura de página del ejercicio “Estructura de páginas. Inserción”, suponiendo que se ha realizado la secuencia de inserciones: en Autor: (A01), (A02), (A04), (A05), (A16) y (A17), en Artículo: (R01), (R02), (R03) y (R04) y, en Escrito: (A01,R02), (A02,R02), (A04,R02), (A05,R02).



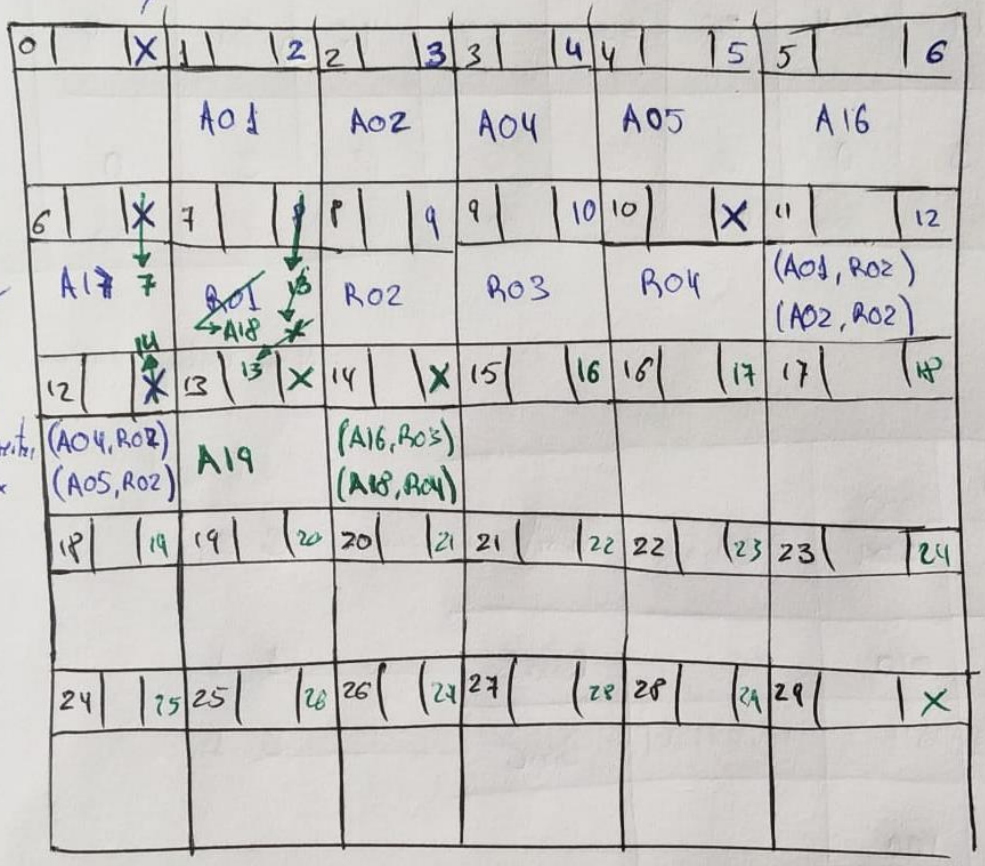
Sobre esta configuración se realizan la siguiente secuencia de actualizaciones en las tablas: se borra (R01), se insertan (A18) y (A19) y los Escritos (A16,R03), (A18,R04).

Seleccione cuáles son los valores correctos para las siguientes páginas después de la mencionada secuencia de actualizaciones:

- Página 0: Página de comienzo de la secuencia de páginas de “Autor”: 1, de “Artículo”: 8, de “Escrito”: 11 y de “Páginas libres”: 15.
- Página 6: Contenido: A17, puntero a la siguiente página: 7.
- Página 7: Contenido: A18, puntero a la siguiente página: 13.
- Página 12: Contenido: (A04,R02) y (A05,R02), puntero a la siguiente página: 14.
- Página 14: Contenido: (A16,R03) y (A18,R04), puntero a la siguiente página: null .
- Página 15: Contenido: vacía, puntero a la siguiente página: 16.

78) 79)

Artículo y Autor ⇒ Bloque 1 = 1 dato por página
Escrito ⇒ Bloque 2 = 2 datos por página



como es de dinámica tabla el sig. a introducir se para x

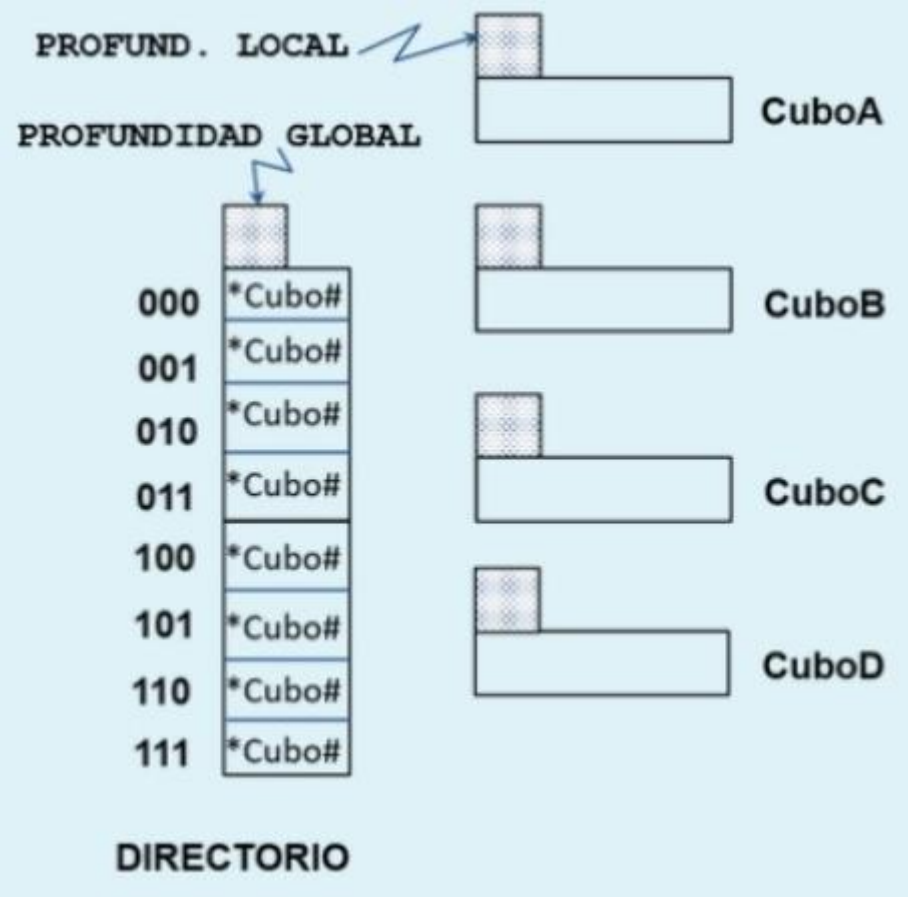
- Cada cuadrito es una página (la 0 siempre es para tabla índice) por lo tanto no apunta a ningún lado.

tabla índice

Conjunto páginas	Dirección la página
Páginas libres	~~13~~ → 15
Autor	1
Artículo	~~7~~ → 8
Escrito	11

80. Supongamos que disponemos de una estructura de “Hashing” Dinámico que alberga hasta 3 registros por página y que usa una función de dispersión

f(x) = x mod 8. Supongamos que insertamos registros con los siguientes valores para la clave: [4,8,19,24,26,35,37,46,51,64]. La estructura generada tiene la configuración:



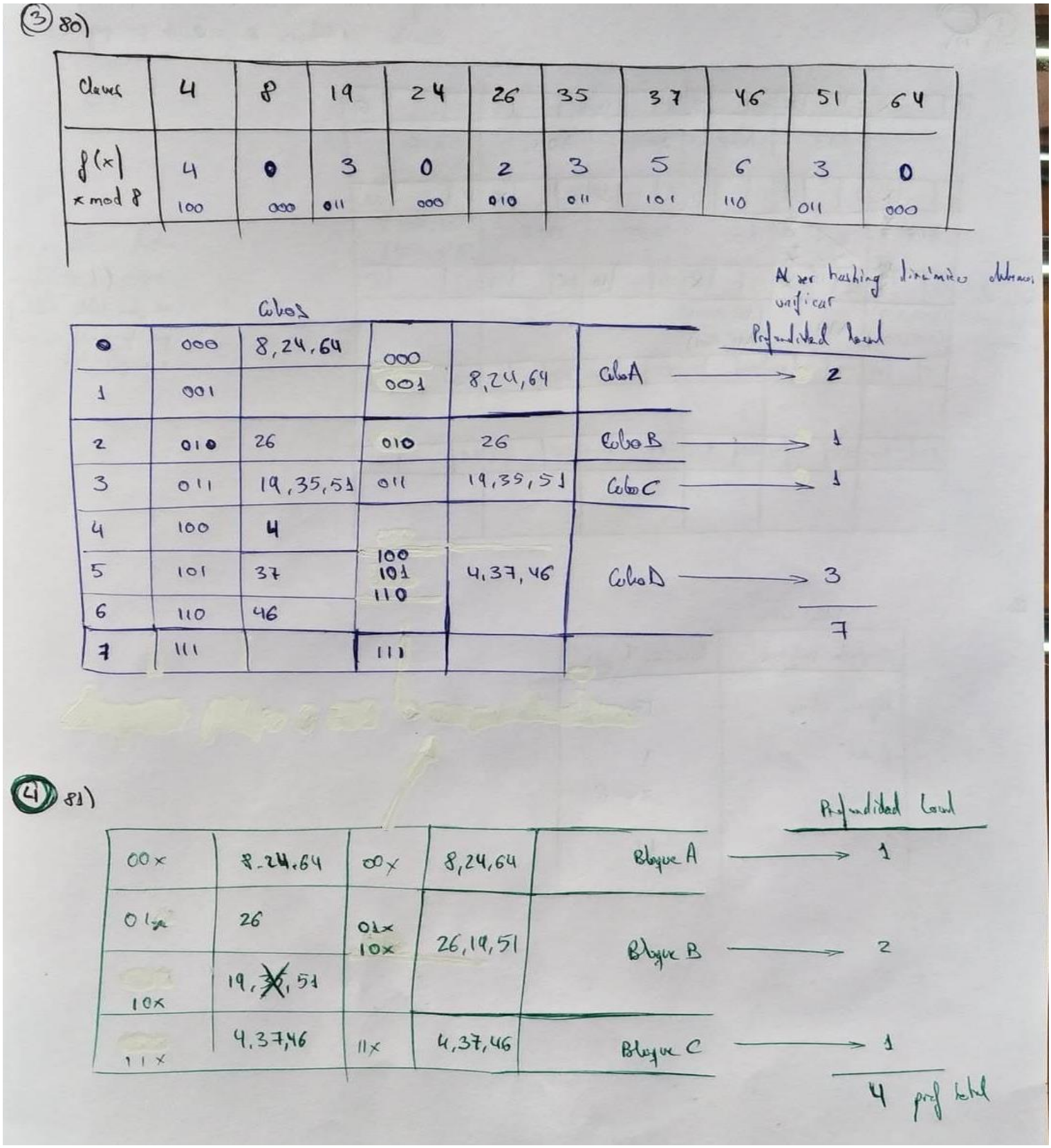
Seleccione cuáles serán los valores correctos para la profundidad Global del directorio, para las entradas del directorio indicadas y para la profundidad local y contenido de cada cubo indicado:

- Profundidad Global del Directorio: 7, profundidades Locales del CuboA: 2, del CuboB: 1, del CuboC: 1 , y del CuboD: 3 .
- Entradas del Directorio: 000: CuboA, 001: CuboA, 010: CuboB, 011: CuboC, 100: CuboD, 101: CuboD, 110: CuboD y 111: nada.
- Claves contenidas en cada cubo: CuboA: 8,24,64, CuboB: 26, CuboC: 19,35,51, CuboD: 4,37,46.

81. Partimos de una estructura de “Hashing” Dinámico que alberga hasta 3 registros por página y que usa una función de dispersión f(x) = x mod 8. Obtenida del ejercicio: “Estructura Hash Extendido. Inserción” después de insertar registros con los siguientes valores para la clave: [4,8,19,24,26,35,37,46,51,64]. Supongamos que, después de eliminar el registro con valor de la clave 35 obtenemos una estructura con la configuración siguiente:

Seleccione cuáles serán los valores correctos para la profundidad Global del directorio, para las entradas del directorio indicadas y para la profundidad local y contenido de cada cubo indicado:

- Profundidad Global del Directorio: 4, profundidades Locales del CuboA: 1, del CuboB: 2 y del CuboC: 1.
- Entradas del directorio: 00x: 8,24,64, 01x: 19,26,51, 10x: 19,26,51 y 11x: 4,37,46.
- Claves contenidas en cada cubo: CuboA: 8,24,64, CuboB: 19,26,51 y CuboC: 4,37,46.



LA PROFUNDIDAD ESTÁ MAL, HAY QUE CONTAR LOS BITS IGUALES PARA LA LOCAL, Y CON CUANTOS BITS SE PUEDE CODIFICAR TODAS LAS ENTRADAS DE CLAVE (EN ESTE CASO 3).

82. El hashing dinámico:
    a. pretente asignar más espacio en disco a zonas del dominio de la clave que teóricamente van a presentar más valores en la instancia de la BD.
    b. pretende asignar más espacio en disco a zonas del dominio de la clave donde se van presentando más valores en la instancia de las BD. **VERDADERO**
    c. NS/NS
    d. pretende asignar más espacio en disco a zonas de clave donde la distribución binaria de valores consecutivos es más densa.
83. En direccionamiento directo, un hueco...
    a. NS/NC
    b. es una parte del rango de entrada que no usa el algoritmo.
    c. es una zona con menos registros en el fichero de salida.
    d. es una parte del rango de salida no asignada por el algoritmo. **VERDADERO**

84. En general, un índice de mapa de bits es adecuado para consultas que ...
    a. es adecuado tanto para consultas que establezcan conjunciones sobre el valor de la clave como para consultas que establezcan disyunciones sobre el valor de la clave
    b. que establezcan conjunciones sobre el valor de la clave, pero no para consultas que establezcan disyunciones sobre el valor de la clave.
    c. NS/NC
    d. establezcan disyunciones sobre el valor de la clave, pero no para consultas que establezcan conjunciones sobre el valor de la clave.
85. En un árbol B+ de orden M...
    a. en cada nivel puede haber como mucho M/2 nodos.
    b. NS/NC
    c. en cada nivel tiene que haber como mínimo M-2 nodos.
    d. un nodo interno puede tener como mucho M hijos. **VERDADERO**
86. En una tabla organizada por índice...
    a. una recuperación completa devuelve siempre las tuplas ordenadas por un campo numérico.
    b. NS/NC
    c. una recuperación completa puede devolver las tuplas sin orden aparente.
    d. una recuperación completa devuelve las tuplas ordenadas por la clave primaria. **VERDADERO**
87. Un objetivo primordial en relación con el método de acceso es ...
    a. evitar la aparición de valores nulos.
    b. organizar los datos de manera que se minimice el número de operaciones de E/S a disco. **VERDADERO**

c. ocultar al usuario el verdadero valor de la clave física.
d. NS/NC