

PreguntasFBD2.pdf



Zukii



Fundamentos de Bases de Datos



2º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada**

5 ECO TIPS FOREVER GREEN

SIGUE LOS CONSEJOS DE FOREVER GREEN
PARA CUIDAR EL MEDIO AMBIENTE

1 2 3 4 5

REUTILIZA ENCIENDE APROVECHA EVITA USA BICICLETA



FOREVER GREEN

¿ AÚN NO
NOS
CONOCES?
VISÍTANOS



1

REUTILIZA**y recicla
todo lo que puedas.**

utilizar siempre objetos
que puedan tener una
segunda vida

2

ENCIENDE**Las luces justas.**

para ahorrar utiliza
los horarios más baratos:

- **llano** de 8:00 a 10:00 h,
de 14:00 a 18:00 horas y
de 22:00 a 00:00 horas
- **valle** de 00:00 a 08:00 h.

3

APROVECHA**todo el agua.**

no desperdices los
primeros litros esperando
a que salga caliente.

4

EVITA**contaminar el
entorno.**

no depositar en el medio
ambiente materiales o
productos que son espe-
cialmente agresivos para
la salud y la naturaleza.

**5 ECO TIPS
FOREVER GREEN**

SIGUE LOS CONSEJOS DE FOREVER GREEN
PARA CUIDAR EL MEDIO AMBIENTE



FOREVER GREEN

5

USA BICICLETA**transporte público o
comparte vehículo
cuando lo utilices.**

Así se reduce el CO2

**¿AÚN NO
NOS CONOCES?**

VISÍTANOS SOMOS

FOREVER GREEN

1. Las páginas que componen un archivo almacenado no tienen por qué estar consecutivas en el disco. (V)
2. El índice no denso mejora el barrido ordenado completo del fichero por la clave física. (F)
3. El acceso directo a registros no permite realizar la lectura secuencial de datos en un rango (V)
4. Se pueden montar tantos índices densos como se necesiten. (V)
5. En un índice multinivel el índice de primer nivel (nodo hoja) puede ser denso o no denso. (V)
6. El acceso directo a registros garantiza siempre que encuentre una tupla con una sola lectura de bloque (F)
7. El acceso directo a bloques o cubos produce menos lecturas en disco que el acceso directo a registros. (V)
8. El índice denso es adecuado para consultas por rango de valores del campo clave (V)
9. Los bloques usados para almacenar los datos de la BD pueden ser de distinto tamaño, dependiendo del tamaño de los registros que se almacenen en ellos. (F)
10. El índice no denso permite realizar preguntas de tipo existencial sin acceder al fichero de datos. (F)
11. El hashing dinámico es el método de acceso que mejor distribuye los datos en el disco y, por tanto, el que menos desperdicio ocasiona (V)
12. Se pueden montar tantos índices no densos como sea necesario (F)
13. Para montar un índice denso, los registros tienen que estar ordenados físicamente por algún campo. (F)
14. El índice denso ocupa el mismo tamaño que el propio fichero que indexa (F)
15. En ficheros no ordenados físicamente, no se pueden montar índices no densos. (V)
16. El índice no denso es mucho menor que el denso cuando caben varios registros en un bloque (V)
17. Las sentencias CREATE TABLE y CREATE INDEX de SQL generan nuevos conjuntos de páginas (archivos almacenados en el nivel interno). (V)
18. El rendimiento de un índice no denso desciende considerablemente cuando se realizan inserciones o borrados (F)
19. En acceso directo a registros, si se produce una colisión, habrá un hueco en el fichero maestro que nunca se va a aprovechar (V)
20. El índice denso no es rentable cuando se actualiza o se inserta con mucha frecuencia (V)
21. El índice no denso es el único mecanismo de indexación posible cuando los datos están ordenados físicamente (F)
22. El hashing dinámico hace falta una estimación del número de datos a insertar para cada tabla hash (V)

23. El gestor de disco forma parte del SGBD (F)
24. Lo normal es que cada archivo almacenado del nivel interno se almacene en un fichero físico separado. (F)
25. Para búsquedas basadas en el valor de la clave, los árboles B garantizan el acceso a un número fijo de páginas de base de datos.
26. El agrupamiento por defecto en el nivel interno es intra-archivos (V)
27. Conociendo el RID de un registro no hace falta más que un acceso a disco para recuperarlo. (V)
28. Se puede montar un árbol B sobre cualquier campo clave utilizando un índice denso como conjunto secuencia. (V)
29. En una base de datos puede haber más de un índice primario. (F)
30. En una organización secuencial no es necesario que los registros mantengan ningún orden en particular. (V)
31. Todas las páginas de una base de datos tienen la misma estructura. (F)
32. Los árboles B se montan en memoria para no tener que acceder a disco más que una vez para llegar al registro. (F)
33. En un índice denso el número de elementos es el mismo que el del archivo principal.
34. En el nivel interno de una BD hay que tener en cuenta también el nivel físico que gestiona el acceso a físico (V)
35. Cuando la clave de un índice es compuesta (C1, C2) resulta eficiente el uso del índice para buscar por C1 o por C2.
36. El número de accesos a disco que hacen falta para obtener una página depende del tamaño de la página y del tamaño del bloque físico.
37. Las actualizaciones en los archivos almacenados no tienen por qué determinar la actualización de los índices no densos. (V)
38. En el hashing extendido una mala elección en el tamaño de las páginas puede obligar a reorganizar completamente a estructura. V
39. La organización multilistas es independiente de las técnicas de acceso al archivo almacenado. (V)
40. Conviene que estén relacionados el tamaño de los bloques físicos y de las páginas para mejorar el rendimiento de sistema de almacenamiento. (V)
41. El orden de un árbol influye directamente en el número de niveles. (V)
42. El orden de un árbol B fija el número de punteros que salen de un nodo. (V)
43. En el agrupamiento inter-archivo se ubican en una página registros de distinto tipo. (V)
44. El orden de un árbol está determinado por el tamaño de la página que se asigna a los nodos del árbol. (V)

45. Se puede combinar una organización multilistas con un árbol B para gestionar los accesos en una estructura de datos jerárquica. (V)
46. Puesto que es una variante de índice no denso, solo se puede montar un árbol B (sobre la clave física) de un archivo. (F)
47. En el hashing extendido lo mejor es que la pseudollave se ajuste el tamaño del índice que se guarda en memoria(V)
48. La organización multilistas puede servir para conectar fichero y es la base de datos basados en grafos. (V)
49. Un factor de bloqueo mayor a 1 implica tener más de un registro por página. (V)
50. El hashing dinámico es muy eficaz porque la tabla hash va en memoria principal (F)
51. El numero de niveles de un arbol B depende sólo del tamaño del fichero base (F)
52. Se puede combinar una organización multilistas con un hashing extendido para gestionar los acceso en una estructura de datos jerárquica (V)
53. La actualización de los archivos puedo no influir en la actualización de los índices no densos (V)
54. Puesto que son formas de índice no denso sólo se puede montar un arbol B sobre la clave física de un archivo (F)
55. En el hasing extendido lo mejor es que la pseudollave se ajusta al tamaño del índice que se guarda en memoria (V)
56. El orden de un arbol influye directamente en el número de niveles (V)
57. En hashing dinámico hace falta una estimación del número de datos a insertar para dimensionar la tabla hash (V)
58. El índice denso es adecuado para consultas por rangos de valores del campo clave (V)
59. El nivel interno de una base de datos está enteramente gestionado por el S.O. del ordenador (F)

— — MULTI — —

1. El objetivo principal de los mecanismos de indexación y métodos de acceso es:
 - Acceder a los datos de una tabla de forma ordenada
 - Poder listar los datos por rangos en cualquiera de sus atributos
 - **Localizar datos requeridos con el número mínimo de operaciones de lectura en disco**
 - Garantizar que no se duplique la clave primaria
2. La sentencia CREATE TABLE provoca:
 - La creación de un nuevo fichero en disco
 - **La creación de un nuevo archivo almacenado**
 - La inserción de una nueva página en un archivo almacenado existente
 - La inserción de un registro en una pagina
3. El clúster
 - Acelera las consultas que involucran la reunión natural de las tablas que contiene
 - Perjudica la lectura individual de las tablas que contiene
 - Es una estructura inter-archivo
 - **Todo lo anterior es cierto**
4. En un índice denso
 - **El número de registros del índice es igual al número de registros de la tabla indexada.**
 - El número de registros del índice es menos que el número de registros de la tabla indexada
 - El índice ocupa lo mismo que la tabla indexada.
 - Ninguna de las anteriores es cierta
5. Indica la afirmación verdadera
 - El índice no denso es el único mecanismo de indexación posible para los datos están ordenados físicamente
 - Un árbol B solo se puede montar sobre la clave física de un archivo
 - Un fichero no ordenado físicamente, se pueden montar índices no densos
 - **Se pueden montar tantos índices densos como se necesiten**
6. La técnica de acceso directo
 - **No permite realizar la lectura secuencial de datos de un rango**
 - Garantiza siempre que encuentre una fila con una sola lectura de bloque
 - No utiliza área de desbordamiento
 - No necesita una estimación previa del número de registros

7. En la indexación con árboles B
- El orden de un árbol está determinado por el tamaño de página/bloque que se asigna a los nodos
 - Se puede montar un árbol B sobre cualquier campo
 - El orden de un árbol incluye directamente en el número de niveles
 - **Todo lo anterior es cierto**
8. 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**
 - c. 16
 - d. 5
9. Un índice no denso
- Es adecuado para consultas por rangos de valores del campo clave
 - Permite realizar preguntas de tipo existencial sin acceder al fichero de datos
 - **Existe que los registros estén ordenados físicamente**
 - El rendimiento desciende considerablemente cuando se realizan muchas inserciones o borrados en la tabla
10. Las páginas que componen un archivo almacenado
- Tienen que estar consecutivas en disco
 - Pueden ser de distinto tamaño dependiendo del tamaño de los registros que se almacenen en ellas
 - Contienen siempre registros de una misma tabla
 - **Todo lo anterior es falso**
11. El índice invertido.
- Es útil para recorrer una tabla por el campo clave en orden inverso al establecido
 - Se puede montar sobre cualquier campo de la tabla
 - **Ayuda a distribuir mejor los datos en el espacio de almacenamiento**
 - Todo lo anterior es cierto
12. Cuando la cardinalidad del campo por el que se indexa una tabla es muy baja, el mejor mecanismo de indexación es:
- **Un índice de mapa de bits**
 - Un árbol B donde el conjunto secuencia sea denso
 - Un árbol B donde el conjunto secuencia sea no denso
 - Algún mecanismo de acceso directo
13. Con la consulta `select codpro, sum(cantidad) from ventas group by codpro,`
- No se puede crear una vista por estar agrupada

- **Se puede crear una vista, pero no será actualizable**
 - Se puede crear una vista y será actualizable porque solo usa una tabla
 - Ninguna de las anteriores es cierta
14. Sean F y D las tablas procedentes de una entidad fuerte y una débil, respectivamente. Las filas de D se recuperan siempre reunidas con las de F, y rara vez, por separado. La mejor opción sería:
- Crear una vista en el que aparezcan los datos de ambas tablas en el formato adecuado
 - Indexar D por el atributo que tiene en común con F
 - Poner en D como clave externa el atributo que tiene en común con F
 - **Almacenarlas conjuntamente en un clúster**
15. Cuando se necesita acceder a la tabla Alumnos por rangos de NOTAS, el mejor mecanismo es:
- El hashing básico
 - Un índice no denso
 - **Un índice denso**
 - Un índice de mapas de bits
16. En el hashing extendido:
- a) Lo mejor es que la pseudollave tenga muchos dígitos.
 - **b) Lo mejor es que la pseudollave se ajuste al tamaño del índice que se guarda en memoria**
 - c) Hay que reservar de antemano un número fijo de bloques.
 - d) No hay que tener una estimación del número de registros a almacenar.
17. El record identifier (RID):
- a) Es un campo de un índice denso.
 - b) Es un campo de un índice no denso.
 - c) Puede servir para identificar varios registros.
 - **d) Se calcula mediante un algoritmo de direccionamiento**
18. Indica cuál de estas afirmaciones es verdadera:
- a) El tamaño de los bloques físicos y de las páginas deben ser independientes.
 - b) Cada archivo almacenado del nivel interno debe estar en un fichero físico separado.
 - c) El nivel interno de una base de datos está enteramente gestionado por el S.O. del ordenador.

- **d) Las páginas que componen un archivo almacenado no tienen porqué estar consecutivas en disco**

19. Las tablas organizadas por índice (IoT):

- a) No admiten ningún tipo de índice.
- b) No tienen llave primaria.
- c) No pueden recuperarse de forma ordenada.
- **d) Se organizan como un árbol B cuyas hojas contienen las tuplas.**

20. Indica cuál de estas afirmaciones es FALSA:

- a) En el hashing extendido una mala elección en el tamaño de las páginas puede obligar a reorganizar completamente la estructura.
- b) El hashing extendido es muy eficaz porque la tabla hash va en memoria principal.
- **c) El índice no denso es el único mecanismo de indexación posible cuando los datos están ordenados físicamente.**
- d) La actualización de los archivos puede no influir en la actualización de los índices no densos.

21. Un índice primario...

- a) Nunca puede ser denso.
- b) Nunca puede ser no denso.
- **c) Ninguna de las otras es cierta.**

22. En general, para acceder a los datos...

- **a) Ni índices ni acceso directo son mejores en términos absolutos frente al otro.**
- b) El acceso directo es siempre la mejor alternativa si está disponible en el SGBD.
- c) Los índices son siempre la mejor alternativa si están disponibles en el SGBD.

23. En un índice de mapa de bits:

- a) El número de entradas es el doble del número de valores que tiene la clave por la que se quiere indexar el fichero.
- b) El número de entradas coincide con el número de registros que hay en el fichero que se quiere indexar.
- **c) El número de entradas coincide con el número de valores que tiene la clave por la que se quiere indexar el fichero**

24. En la aproximación del método de acceso a la base de datos vista en clase:

- a) El gestor de archivos y el gestor de disco son dos elementos del S.O. que permiten la transformación entre páginas almacenadas y sectores de disco.

- b) El gestor de archivos y el gestor de disco son dos formas de decir lo mismo.
- **c) Ninguna de las otras es cierta.**

25. Cuando se utilizan técnicas de hashing básico...

- a) Deja de ser necesario usar estrategias para solventar colisiones, puesto que estas no pueden presentarse.
- b) En general, no ayuda a conocer zonas del dominio del campo clave donde pueden presentarse más valores.
- **c) Es posible que se produzcan huecos.**

26. En relación con los índices:

- a) No se pueden utilizar si el fichero de datos no está ordenado por la clave del índice.
- b) Cuantos más campos añadamos a la clave de búsqueda, más rápido será el proceso de búsqueda en el índice por uno de esos campos.
- **c) Ninguna de las otras es cierta.**

27. En general, cuanto mayor es el número de colisiones que produce una función hash...

- **a) Las búsquedas tenderán a ser más lentas.**
- b) Menos huecos se producirán.
- c) Ninguna de las otras es cierta.

28. Un objetivo primordial en relación con el método de acceso es...

- a) Evitar la aparición de valores nulos.
- b) Ocultar al usuario el verdadero valor de la clave física.
- **c) Ninguna de las otras es cierta.**

29. Considere las tablas organizadas por índice (IoT):

- a) Son la forma de implementar los árboles B+ en bases de datos.
- b) La clave de búsqueda del índice no tiene nada que ver con la clave física.
- **c) Ninguna de las otras es cierta.**

30. En el nivel interno...

- a) El almacenamiento persistente de los datos se hace con dispositivos de memoria de las primeras posiciones de la jerarquía de memoria para que las operaciones sean más rápidas.
- b) El almacenamiento persistente de los datos se hace con dispositivos de memoria de las primeras posiciones de la jerarquía de memoria para que sea más barata la implantación del sistema.
- **c) Ninguna de las otras es cierta.**

31. Considere que se está usando hashing dinámico. En un momento dado, al insertar un nuevo registro en un cubo con profundidad local igual a la profundidad global:

- **a) En ningún caso habrá que desdoblar la tabla índice.**
- b) Es posible que no haya que desdoblar la tabla índice.
- c) Necesariamente habrá que desdoblar la tabla índice.

32. Considere un fichero secuencial indexado:

- a) Cuanto más grandes sean los registros del fichero índice, más se ayudará a acelerar el proceso de búsqueda.
- **b) Cuanto más pequeños sean los registros del fichero índice, más se ayudará a acelerar el proceso de búsqueda**
- c) Ninguna de las otras es cierta.

33. El hashing dinámico...

- **Va asignando más espacio en disco a zonas del dominio de la clave donde se van presentando más valores en la instancia de la base de datos**
- Asigna más espacio en disco a aquellas zonas del dominio de la clave que teóricamente van a presentar más valores en la instancia de la base de datos.
- No utiliza ninguna estructura auxiliar, aparte del propio fichero que almacena los registros.

34. En un árbol B+:

- **a) Cuanto menor es M, mayor tiende a ser el número de niveles.**
- b) Cuanto mayor es M, mayor tiende a ser el número de niveles.
- c) Ninguna de las otras es cierta.

35. La BD de datos en el nivel interno se puede representar de distintas formas, pero:

- a) Nunca deben ubicarse juntos registros de distinto tipo para facilitar operaciones.
- b) Cuando se ponen juntos los registros del mismo tipo, se optimizan operaciones como las de reunión de tablas.
- **c) Ninguna de las otras es cierta.**

36. Cuando se utiliza hashing dinámico:

- a) No se puede utilizar una función de direccionamiento.
- b) Es imposible que se produzcan colisiones.
- **c) Ninguna de las otras es cierta.**

37. En relación con el método de acceso a la BD, las páginas o bloques de la BD deben tener un tamaño múltiplo de las páginas o bloques del sistema operativo (mínima unidad de E/S)...

- a) Para garantizar que la memoria y el disco duro sean compatibles.

- **b) Para aprovechar bien cada operación de E/S en disco.**
 - c) Para que sea fácil hacer cuentas a la hora de organizar los datos.
38. Cuando se organiza el acceso a los datos de un fichero mediante el uso de índices:
- a) En general, lo mejor es usar tantos índices como configuraciones de consulta pueden plantearse.
 - b) Si el espacio de disco no es un problema, lo mejor es usar tantos índices como configuraciones de consulta pueden plantearse.
 - **c) Ninguna de las otras es cierta.**
39. Al respecto de los índices jerárquicos:
- a) Los árboles B+ no son un ejemplo de índice jerárquico.
 - **b) El número de niveles depende, entre otras cosas, del número de registros del fichero de datos.**
 - c) Los árboles B no son un ejemplo de índice jerárquico.
40. En general, cuando se utiliza un índice denso:
- a) Al realizar una operación de borrado de un registro en el fichero nunca hay que actualizar el índice.
 - b) Al realizar una operación de inserción de un nuevo registro en el fichero nunca hay que actualizar el índice.
 - **c) Ninguna de las otras es cierta.**
41. En relación con la capacidad de consulta:
- **a) Todas las consultas que se pueden resolver con SQL, se pueden resolver con Álgebra Relacional.**
 - b) Todas las consultas que se pueden resolver con Álgebra Relacional, se pueden resolver con SQL.
 - c) No sabe o no contesta.
42. Considere dos tablas T1 y T2 tales que el esquema de T2 está contenido (sin ser igual) en el de T1. Considere la siguiente expresión en el álgebra relacional:
 $(T1/T2) \times T2$
donde "/" representa la división y "x" el producto cartesiano.
El resultado de aplicar dicha operación sobre dos instancias T1 y T2 (respectivamente) de las tablas:
- **a) Producirá como resultado una instancia t contenida en o igual a t1.**
 - b) Producirá siempre como resultado la instancia t1.
 - c) Ninguna de las otras opciones es cierta.
43. Cuando operamos con dos tablas que están conectadas por una clave externa:
- a) Su producto cartesiano siempre devuelve la misma cantidad de tuplas que su reunión.
 - **b) Su producto cartesiano puede devolver más tuplas que la reunión, aunque no siempre.**

- c) Su producto cartesiano puede devolver menos tuplas que su reunión, aunque no siempre.
44. En relación con los operadores fundamentales y no fundamentales del Álgebra Relacional:
- Solo una parte de los operadores no fundamentales pueden reproducirse utilizando operadores fundamentales.
 - **Todos los operadores no fundamentales pueden reproducirse utilizando operadores fundamentales.**
 - Todos los operadores fundamentales pueden reproducirse utilizando operadores no fundamentales.
45. En general, en Oracle, si añadimos un índice a una tabla...
- a) Habrá que reescribir todas las sentencias de consulta que hayamos planteado previamente sobre dicha tabla, puesto que pueden dar errores sintácticos.
 - b) Habrá que reescribir sólo las sentencias de consulta en las que aparezca la tabla más de una vez, puesto que darán errores sintácticos.
 - **c) Ninguna de las otras opciones es cierta.**
46. Piense en el esquema de suministros que hemos utilizado en las prácticas con Oracle. En relación con el comando DESCRIBE...
- a) Se trata de un comando fundamental, puesto que la información que proporciona sobre dichas tablas es imposible de obtener de otra manera.
 - b)
 - c) La información que proporciona el comando DESCRIBE sobre dichas tablas está almacenada en el catálogo de la base de datos.**
- c) Ninguna de las otras opciones es cierta.
47. En el hashing extendido una mala elección en el tamaño de las páginas puede obligar a reorganizar completamente la estructura
- A) Verdadero**
B) Falso
C) NS/NC
48. En el nivel interno de una bases de datos hay que tener en cuenta también el nivel físico que gestiona el acceso a disco
- A) Verdadero**
B) Falso
C) NS/NC