**Verdadero / Falso**

1. El tiempo de ejecución del algoritmo de clasificación Heapsort solo depende de la cantidad de elementos a ordenar.

*FALSO. Depende también de cómo vienen los datos. Si los datos vienen ordenados, el algoritmo va a ser más eficiente.*

1. Un árbol de grado mayor a dos no puede ser representado mediante una representación computacional estática.

*FALSO. Con una matriz.*

1. El algoritmo de Huffman basa la asignación de códigos comprimidos basándose en un árbol binario.

*FALSO: Huffman usa un árbol binario como estructura de datos, y también una tabla*

*de frecuencias donde se guardan los caracteres empleados en el texto. ???*

1. Al algoritmo de Quicksort es siempre más performante que el Heapsort.

*FALSO. Depende también de cómo vienen los datos. Si los datos vienen ordenados,*

*el algoritmo va a ser más ineficiente.*

1. Una subconsulta ubicada en el where siempre debe retornar al menos una fila.

*FALSO: si se pone un “exists” en el where con una subconsulta, esta misma puede*

*no traer datos.*

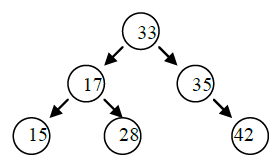
1. Si una columna posee la constraint unique, entonces ninguna fila acepta nulos en dicha columna.

*FALSO. La constraint unique acepta un solo nulo.*

1. Al aplicar un barrido simétrico sobre ABB se obtiene el conjunto de datos ordenados.

*VERDADERO. Esta es la forma de obtener los datos ordenados.*

1. La estructura de ABB (árbol binario de búsqueda) es un árbol completo.

FALSO.

El ABB de la imagen no es completo.

1. En Huffman si el código es 1001, entonces ningún otro caracter puede tener el código 10011.

FALSO. Huffman guarda el código asignado a cada caracter y la longitud del

mísmo.

1. El árbol B+ nunca puede quedar lleno.

FALSO. Usando Load Factor al 100% puede que los nodos queden llenos.

1. La siguiente consulta retorna como máximo una fila.

SELECT distinct 1 FROM tabla1 WHERE campo1=1

UNION ALL

SELECT distinct 1 FROM tabla1 WHERE campo2=2

FALSO. El “union all” muestra todas las filas, sin importar que estén repetidas.

1. Dado el grafo G(P,E) donde P={(a)} y E={(a,a)} decimos que es el mismo árbol.

FALSO. Dicho grafo no es un árbol, ya que para que sea árbol la suma de los nodos

debe ser igual a la suma de arcos más uno.

1. Configurando el Read Commited Isolation Level al ejecutar una transacción, se evita que otras transacciones modifiquen las filas que han sido leídas por la transacción actual.

VERDADERO. Read commited lee las filas y libera el lockeo.

1. El árbol B es más óptimo para la búsqueda de claves puntuales que la estructura de Hashing.

FALSO. Es al contrario, Hashing es mucho más performante ya que genera un

acceso directo al dato.

1. Huffman, el árbol en el que se basa es una árbol principal derecho balanceado.

FALSO. El árbol que usa Huffman puede o no ser balanceado.

1. ABB, cuando un ABB se basa en un árbol AVL su orden de complejidad será el mejor, nlog2n

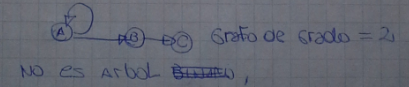
VERDADERO. Es el mejor orden de complejidad que puede tener para AVL.

1. Un vector es una representación computacional estática que puede almacenar un árbol.

VERDADERO. El ejemplo es Heapsort que usa una estructura estática que es un

vector y una lógica que es un árbol. ???

1. Todo grafo de grado 2 es binario.

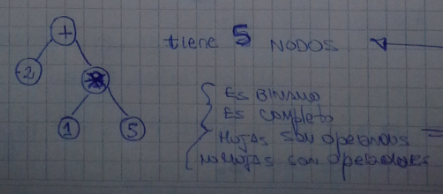
FALSO. 

1. Para reducir el espacio al representar un grafo siempre es más conveniente la forma dinámica que estática.

VERDADERO. El uso de estructuras dinámicas hace un uso eficiente del espacio.

Las estáticas desperdician mucho más, por el las matrices.

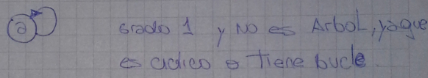
1. La cantidad de nodos de un árbol de expresión siempre es par.

FALSO. 

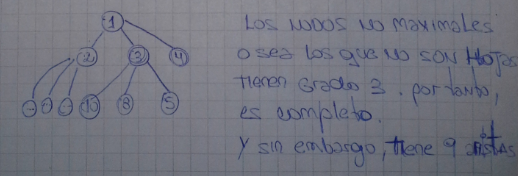
1. El algorimo Heapsort siempre tiene el mismo orden de complejidad para cualquier orden en el que ingresan los datos.

VERDADERO. El orden de complejidad no varía según cómo ingresan los datos.

1. Todo grafo de grado 1 es tambíen un árbol.

FALSO.

1. Si un árbol es completo, la cantidad de arcos es par.

FALSO. 

1. Si un árbol B tiene N claves entonces el grado es N+1.

FALSO. Un árbol de dos nodos puede solo tener un arco, por lo cual es de grado

uno.

1. Es posible asegurar la integridad referencial entre dos tablas de BD distintas.

VERDADERO. Por medio de vinculaciones (link) de BDs.

1. Una primery key y una foreign key son restricciones al modelo físico.

VERDADERO. ambas son restricciones, la PK de unicidad y la FK de restricción

referencial

1. Las únicas restricciones posibles a aplicar en el modelo relacional son por tabla.

FALSO. Existen restricciones a nivel de data type. Validaciones a nivel de campo, por ej, constraint de dominio.

1. La cantidad de nodos de un árbol de Huffman siempre es impar.

FALSO. Huffman utiliza un árbol binario, este puede o no tener cantidad de nodos

impar.

1. Sobre un árbol n-ario con n>2 se pueden realizar los siguientes barridos: preorden, simétrico, posorden y por niveles.

VERDADERO. En cualquier árbol se pueden realizar ??

1. Un árbol binario de búsqueda siempre es más rápido que una lista para ordenar un conjunto de valores.
2. Debido a que el crecimiento de un árbol es exponencial en base al grado del mismo, los tiempos de búsqueda en el mismo son siempre logarítmicos.
3. La reexpresión de caracteres al aplicar huffman implica la disminución de 8bits para la expresión de todos los caracteres.
4. El orden de complejidad de ABB siempre es mejor que el orden de complejidad de Quicksort.

**V/F más viejos**

1. La técnica de Hash abierto puede generar muchas lecturas secuenciales para un valor “clave hash” cuando hay alto grado de repetición de claves de usuario.

VERDADERO.

1. La compactación por algoritmo de Huffman permite redefinir el almacenamiento lógico de símbolos de tal manera que la pérdida de información sea despreciable.

FALSO. A diferencia de otros algoritmos de compactación (jpeg, divx,  
 mp3), Huffman no genera pérdida de información.

1. Una tabla de Hash permite desarrollar un mecanismo indexado para recuperación de claves únicas.

FALSO. Las claves son únicas, no se repiten.

1. Un DBMS que soporte atomicidad, consistencia, Aislamiento y durabilidad puede ser considerado transaccional.

VERDADERO.

1. En SQL, una subconsulta ubicada en el WHERE siempre debe retornar una fila y una columna.

FALSO.

1. Un índice en una base de datos relacional es una restricción al modelo físico.

FALSO. Ya que mejora la velocidad de las operaciones, permitiendo un rápido acceso a los registros de una tabla.

1. En PL-SQL, la única diferencia entre una función y un store procedure es que puede retornar valores.

FALSO.

1. Si en una consulta SELECT, hay al menos una función de grupo (COUNT, SUM, AVG, etc.) siempre debe colocarse la cláusula GROUP BY en dicho SELECT.

FALSO. La función de grupo puede ser aplicada a todo el set de datos que devuelve el select.

1. Los árboles B garantizan mayor velocidad que Hashing para el acceso a los datos

FALSO. Influye el volumen de los datos, los tipos de clave, dado que Hashing genera redundancia y el tipo de acceso, dado que Hashing está preparado para un acceso directo y no secuencial

1. Un árbol de Huffman siempre es completo.

FALSO. No siempre se cumple esa condición.

1. Una tabla no puede tener 2 claves foráneas que referencian a la misma tabla.

FALSO.

1. La implementación de una tabla de Hash es utilizada para recuperar datos  
    por medio de una clave permitiendo un acceso de alta velocidad a los datos.

VERDADERO, justamente está describiendo el objetivo de una tabla de hash.

1. En un motor de base de datos relacional, una función creada por el  
    usuario puede modificar el contenido de una tabla de la base donde fue  
    creada.

FALSO. Una función no permite hacer modificaciones a los datos, solamente consultarlos. Esa es una de las diferencias entre los SP y las funciones.

1. Las "foreign key" brindan la funcionalidad de eliminar la redundancia de datos.

FALSO. Las claves FK permiten hacer una relación Padre-Hijo, Maestro-Detalle. Sirven para proporcionar integridad referencial.

1. La Normalización aplicada al diseño de un modelo de datos relacional, nos permite desarrollar un modelo de manera estructurada independientemente de la performance que ese modelo pueda llegar a obtener

FALSO, siempre que hay normalización hay pérdida de performance.

1. Todos los motores de Bases de Datos utilizan el método de Hashing para el armado de sus índices.

FALSO. Por lo gral usan Arbol-B (en sus diferenes opciones)

1. Las dimensiones en una base de datos multidimensional indican la forma de acceder a la información que se encuentra en la intersección de dichas dimensiones.

VERDADERO??

1. El checksum es una de las técnicas utilizadas para corroborar la integridad de los datos.

VERDADERO.

1. Los triggers son las funciones con las que cuenta un Motor de base de datos que permite controlar la integridad relacional.

VERDADERO. Es una de las herramientas que ofrece SLQ para dicho fin.

1. Datamining son las técnicas y algoritmos utilizados para encontrar información y relaciones ocultas en un Datawarehose.

VERDADERO. Data Mining, la extracción de información oculta y predecible de grandes bases de datos, es una poderosa tecnología nueva con gran potencial para ayudar a las compañías a concentrarse en la información más importante de sus Bases de Información (Data Warehouse).

1. El algoritmo de quicksort tiene en promedio un grado de complejidad O(nlog n) pero en determinada circunstancia puede tener grado de complejidad O(n2) y ser el peor de todos los métodos de clasificación.

FALSO. Es el algoritmo que mejor responde en la mayoría de los casos, con un tiempo promedio de O(nlog n) y O(n2) en el peor de los casos. El más rápido en la práctica. Implementado en gran cantidad de sistemas. No es el peor de los métodos.

1. El método de Hashing es más rápido para el acceso a los datos que el Árbol B.

VERDADERO. Puede ser que al estar en memoria por eso es más rápido ¿?

1. En la implementación de un Árbol B, todos los nodos de datos que contienen claves, se encuentran en el mismo nivel.

VERDADERO. Inserción → Si el nodo hoja tiene menos de M −1 claves en el, simplemente insertamos el elemento en el nodo hoja y damos por terminado. Por tanto quedarán en el nivel inferior siempre.

1. En un índice de un DBMS, armado en un árbol B, el tiempo de acceso a la información depende en parte del tamaño de la clave almacenada.

FALSO. En el peor de los casos, el camino de búsqueda termina en el nodo de la hoja más profunda, por lo tanto el tiempo de ejecución en el peor de los casos esta determinada por la altura del árbol-B.

1. En un modelo de DB OLTP el concepto de transacción está asociado a la atomicidad de procesamiento

VERDADERO. Si bien la transacción se realiza exitosamente se asegura la consistencia de ese único pedazo de datos

1. En un modelo OLAP se puede aplicar normalización en cualquier tabla de dimensión.

FALSO. Si ya que están totalmente renormalizadas pero pierde el modelo de STAR

1. En un modelo OLAP se puede aplicar normalización en cualquier tabla que no sea la Tabla de Hechos.

VERDADERO. Esta tabla no se podría normalizar ya que contiene las foreing key para conectarse con el resto.

1. La compresión que se logra mediante el algoritmo de Huffman es mayor cuando la variedad de caracteres diferentes que aparecen es menor.

VERDADERO. Ya que cuenta las repeticiones de cada carácter

1. La técnica de Hash no puede ser implementada para uso de caché, ya que no garantiza el acceso de alta velocidad.

FALSO.

1. No es posible obtener el mismo resultado de una consulta OLAP a partir del OLTP que dío origen al mismo.

VERDADERO. Le da un tratamiento totalmente diferente a los datos y tiene a su vez orientaciones totalmente diferentes un OLAP está orientado al análisis empresarial, en cambio OLTP esta restringido a tareas específicas. Por lo tanto los resultados no serán los mismos

1. Todos los motores de Bases de Datos utilizan el método de Hashing para el armado de sus índices.

FALSO. También utilizan Árbol-B

1. Un árbol siempre tiene más punteros que elementos de datos.

FALSO. Un árbol con N nodos debe tener N −1 aristas (punteros) porque a cada nodo, excepto la raíz, le llega una arista.

1. Un modelo OLAP, posee al menos una tabla de hechos (FACT TABLE), con campos precalculados.

FALSO. Solo posee una única tabla de hechos

1. Una vista en un RDBMS permite restringir el acceso al modelo además de mostrar una visión simplificada del mismo.

VERDADERO.

1. Una vista es el equivalente a una consulta estática de una o más tablas.

FALSO. Desde el lado del resultado es el mismo, pero por otro lado la Vista guarda el “programa de ejecución” de la consulta, lo cual la hace más rápida a la hora de ejecutarce.

1. Para representar grafos irrestrictos son más perfomantes las representaciones dinámicas que las estáticas.
2. Las funciones de hashing no poseen funciones inversas.

-------------

1. Dado el árbol {(c,a);(c,b);(c,d);(c,e)} su barrido simétrico es a,b,c,d,e.

---------------

1. Un árbol siempre tiene más punteros que elementos de datos.

------------