

# VF y desarrollables reinosa para ctrl f

<input checked="" type="checkbox"/> Hecha	<input type="checkbox"/>
---	--------------------------

## Vof TODOS CHEQUEADOS

Un vector se puede utilizar para representar cualquier grafo restringido.

**Verdadero'**

Cuando un grafo no es unívoco, debe ser categorizado como un grafo irrestricto.

**VERDADERO.** DADO QUE PERMITE CUALQUIER TIPO DE RELACIONES'

Un motor de base de datos cuenta con su propio User Manager, File Manager y Disk manager, por lo cual no tiene contacto con el sistema operativo sobre el cual corre.

**FALSO.** MAS ALLÁ DE TENER ESOS COMPONENTES REQUIERE COMUNICARSE CON EL SO PARA ACCEDER A LOS DISPOSITIVOS'

Las únicas restricciones posibles aplicables en el modelo relacional, son por tabla.

**FALSO.** POR EJ EXISTEN LAS RESTRICCIONES POR COLUMNA COMO EL NOT NULL Y LOS CHECK'

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

**VERDADERO'**

En una base de datos relacional, los índices garantizan la unicidad de claves.

**FALSO.** ESO LO REALIZAN LAS PK PUEDE HABER INDICES CON CLAVES DUPLICADAS'

A veces una Primary Key en una dentro de una Base de datos Relacional puede permitir valores repetidos.

**FALSO'**

Un grafo es biunivoco cuando tiene un solo predecesor y un solo sucesor .

**VERDADERO'**

Si un arbol binario esta completo y balanceado todas las hojas estan en el mismo nivel.

**FALSO.** PUEDE QUE NO TIENE QUE ESTAR PERFECTAMENTE BALANCEADO Y NO SOLO BALANCEADO.'

Las tecnicas más comunes utilizadas en datamining son las redes neuronales, los arboles de decision y las reglas de inducción.

**VERDADERO** , SON LAS TECNICAS QUE SE UTILIZAN'

Si en un arbol el subarbol derecho e izquierdo tienen la misma profundidad y la misma cantidad de elementos, entonces está perfectamente balanceado.

**FALSO** , TIENE QUE OCURRIR EN TODOS LOS NIVELES'

En SQL, una subconsulta en el SELECT siempre debe retornar una fila y una columna.

**VERDADERO'**

El comando CHECK es una restricción al modelo .

**VERDADERO'**

El heap es un arbol n-ario el cual debe cumplir la norma de que el sea mayor que sus hijos en todos los niveles

**Falso'**

Si una columna posee la constraint UNIQUE, entonces una sola fila como maximo puede contener NULL en dicha columna

'  
**Veradero'**

Un arbol B o bien un B+ tiene un alto costo de mantenimiento, y por ello,  
implementa  
un factor de carga en cada nodo

'  
**Verdadero'**

Una foreign key FK es una relacion entre uno o mas campos de una tabla con  
uno o mas  
campos de otra de igual tipo.

'  
**Falso'**

en SQL, una subconsulta en el SELECT siempre debe retornar una fila y una  
columna

'  
**Verdadero'**

A veces una primary key PK en una tabla dentro de una base de datos  
relacional puede  
permitir valores repetidos

'  
**Falso'**

En una base de datos relacional, los indices garantizan la unicidad de claves

'  
**Falso'**

La cantidad de nodos de un arbol de expresion siempre es par

'  
**Falso'**

Si un arbol binario esta completo y balanceado todas las hojas estan en el  
mismo nivel

'  
**Falso'**

Una subconsulta en el HAVING, siempre debe retornar una fila y una columna

'  
**Falso'**

La ejecución sin filas de resultado de una query dentro de un trigger genera la cancelación de la transacción

**Falso'**

El barrido de pre orden se puede hacer en todo tipo de árboles pero el simétrico

es solo para árboles binarios

**Verdadero'**

Todo grafo de grado 2 es un árbol binario

**Verdadero'**

Un vector se puede utilizar para representar cualquier grafo restringido

**Verdadero'**

Una tabla sin Primary Key no puede tener una Foreign Key

**Falso'** → xq una tabla no puede no tener PK

Un Data Mart es un Datawarehouse afectado a un Departamento o sector de la empresa

**Verdadero'**

Solo los árboles binarios perfectos son balanceados

**Falso'**

Si un árbol está balanceado entonces está completo

Falso

En una tabla de una BD relacional, no se puede definir más de un trigger por evento (insert delete update)

Falso

Los índices en una base de datos relacional solo sirven para aumentar la velocidad de acceso a los datos

Verdadero → Solo sirven para evitar el acceso secuencial

Un sistema de Data Warehousing no permite la integracion de bases de datos heterogeneas

Falso

En SQL, la clausula having, solo puede ser usada acompañada por un group by

Verdadero

La re-expresion de caracteres al aplicar huffman implica la disminucion de 8 bitd para la expresion de todos los caracteres

Falso

Un motor de bases de datos relacional construye sobre tres niveles, fisico, conceptual y externo

Verdadero

En la implementacion de un arbol b, todos los nodos de datos que contienen claves se encuentran en el mismo nivel

Verdadera

En una base de datos relacional una tabla siempre debe tener una PK

Falso, puede crearse sin PK

Cualquier arbol puede ser representado computacionalmente mediante un vector

Verdadero

En el modelo estrella, primero se deben cargar las tablas de dimensiones y luego la tabla de hechos

Verdadero

En una base de datos relacional, los indices garantizan unicidad de claves

falso

A veces una primary key en una tabla dentro de una bd relacional puede permitir valores repetidos

falso

El metodo de arbol b no es aplicable a archivos con gran volumen de datos

Falso

El comando check es una restriccion al modelo

Verdadero

Si en un arbol el subarbol derecho e izq tienen la misma profundidad y la misma cantidad de elementos, entonces esta perfectamente balanceado

Falso → tiene que ocurrir en todos los niveles

un vector se puede utilizar para representar cualquier grafo restricto

Verdadero

Un motor de BD cuenta con su propio, user manager, file manager y disk manager, por lo cual no tiene contacto con el Sistema Operativo sobre el cual corre

Falso

Un grafo es biunivoco cuando tiene un solo predecesor y un solo sucesor

Verdadero

Si un arbol binario esta completo y balanceado todas las hojas estan en el mismo nivel

Falso

Las funciones de hashing no poseen funciones inversas

Verdadero

Una tabla sin Primary Key no puede tener una Foreign Key

Falso

Las tecnicas mas comunices utilizadas en el Datamining son las redes neuronales, los arboles de decision y las reglas de induccion

Verdadero

Un vector puede representar computacionalmente cualquier grafo irrestricto

Falso

Una tabla de dimension en el Modelo Star es el equivalente a una cara en un cubo de informacion multidimensional

Verdadero

Cuanto mas chico se tome el factor de split de un arbol B, mayor sera la cantidad de niveles del mismo

Verdadero

La maxima cantidad de nodos que puede contener un arbol esta acotada por el grado del mismo

Verdadero

Una Foreign Key es una restriccion al modelo

Verdadero

LA integridad Referencial en un sistema de gesgtion de bases de datos relacional es la restriccion que garantiza la no duplicidad de filas

Falso

Un motor de base de datos relacional utiliza los conceptos de algebra relacional para la resolucion de consultas

Verdadero

Si una palabra se escribe igual de izquierda a derecha como de derecha a izquierda, entonces su codigo de huffman se escribe igual de izquierda a derecha como de derecha a izquierda

Falso

En una base de datos relacional, no se puede utilizar un select con un outer (left o right) join entre dos tablas, si no existe entre ellas una relacion fisica (FK)

Falso

En un modelo de base de datos se dice que una entidad es independiente si y solo si no recibe una relacion de otra entidad

veradero

Un motor de base de datos se constituye sobre tres niveles: fisico, conceptual y externo

Verdadero

El tiempo de ejecucion de cualquier algoritmo de clasificacion siempre depende aunque sea en parte de la cantidad de elementos a ordenar

Verdadero

Un constraint de tipo CHECK puede ser reemplazado por un trigger

Verdadero

El algoritmo de Huffman obtiene los codigos comprimidos parseando un arbol binario balanceado

Falso

En un motor de bases de datos relacional, se aplica la metodología transaccional, por lo cual hasta que no se verifica con éxito la última transacción no se realiza ningún cambio en la base de datos

Verdadero

Los arboles B garantizan mayor velocidad que hashing para el acceso a los datos

Falso

Un comando SELECT no acepta mas de tres niveles de recursividad interna

Falso

En un arbol de Huffman, si el codigo del caracter 'm' es 1011, entonces no puede ningun otro caracter poseer el codigo 1011111

Verdadero

## #desarrollables

Describir el concepto ed colision de hashing, porque ocurre y que formas hay para mitigarla

Una colisión ocurre cuando se aplica la función hashing a dos elementos de un conjunto del dominio, y se obtiene el mismo resultado. Lógicamente, no pueden almacenarse dos registros en una misma posición

Esto puede ocurrir porque el conjunto dominio (los elementos) es mayor que el conjunto imagen (el rango de valores que puede generar la función hashing), siendo la relacion entre el dominio y la imagen un uno a muchos.

La función de hashing debe armarse de forma tal que a pesar de que el problema no sea completamente evitable, se pueda evadir al máximo la frecuencia de la ocurrencia. Esto se logra cuando la función tiene buena dispersión.

Algunas de las formas de mitigarlas son el encadenamiento y el direccionamiento abierto.

El encadenamiento consiste en que cada casilla en el arreglo referencia una lista de los registros insertados que colisionan con la misma casilla. Se



encuentra la casilla correcta y se agrega al final de la lista correspondiente. Para borrar, se busca y quita de la lista. Es dinámico

El direccionamiento abierto consiste en la búsqueda de otra posición dentro de la tabla en el caso que un registro no se pueda ubicar en el índice calculado por la función hash. Para seleccionar esta nueva posición, existen varios métodos, tres de ellos son: el sondeo lineal, el sondeo cuadrático y el hashing doble.

En el sondeo lineal se busca secuencialmente en la tabla hasta encontrar una posición vacía. En el caso de llegar al final de la tabla se vuelve a la primera posición. El sondeo cuadrático busca posiciones a una distancia específica del punto inicial de sondeo. Esta variación permite una mejor distribución de las claves colisionadas. La función es  $F(i) = i^2$ . Como mencione antes, a mayor dispersión menor ocurrencia de colisiones.

El hashing doble vuelve a aplicar la función hash a la clave, con una función hash diferente a la primera. El resultado es el tamaño del salto a hacerse. Para que sea efectiva la segunda función debe arrojar resultados mayores a 0.

'Un grafo se define como  $G(V, A)$  donde  $V$  es el conjunto de vértices (o nodos) y  $A$  el conjunto de aristas que las une. Un grafo sirve para modelizar problemas matemáticos, el más conocido: el problema del viajero que aún no tiene solución por ser de complejidad NP.'

Su clasificación puede ser por restricción:

- Restritos: Solo pueden modelizar relaciones que cumplan las propiedades de transitividad, simetría y reflexividad
- Irrestritos: Pueden modelizar cualquier tipo de relación sin importar las restricciones arriba mencionadas  
Por su direccionalidad.
- Dirigidos: Las aristas que unen los vértices tienen un sentido de recorrido, eso significa que no son bidireccionales.
- No dirigidos: Las aristas pueden ser recorridas en un sentido o el otro y ambas son válidas.

Hay problemas de la realidad donde es necesario grafo dirigido y otros que no. Ninguno es 'mejor' que el otro

La representación computacional puede ser estática o dinámica.

Las representaciones estáticas pueden ser:

Matriz de adyacencia: Conecta vertice con vertice, como resultado resulta en una matriz de  $n \times n$  siendo  $n$  el numero de vertices. Es muy ineficiente en espacio porque su tamaño es fijo y es funcion de la cantidad de vertices  $n$  ( $n \times n$ )

Matriz de Incidencia: Las filas son vertices y las columnas aristas. Si el grafo es no dirigido contiene como máximo dos 1 por columna. Es de tamaño fijo  $n \times m$  siendo  $n$  los vertices y  $m$  las aristas. Ineficiente como la anterior para grafos dispersos

Las dinámicos suelen representarse en:

Listas de adyacencia: Cada vertice es un nodo con una lista de adyacencia asociada a él. De esta forma el tamaño crece dinámicamente conforme aumentan los caminos entre vertices. Es más eficiente en terminos de complejidad espacial porque no guarda datos sobre vertices no conectados (a diff de la las repr. estaticas)

Listas de Graal: Es una mejora de las listas de adyacencia que utiliza punteros para optimizar el uso de la memoria. También consiste en un nodo con un puntero a una lista de nodos a los cuales puede ir este vertice. Si es un grafo no dirigido, el vertice destino tambien va a tener al propio en su lista de adyacencia (o un puntero a el)

Representación de Platz: Utiliza structs para representar un nodo con su informacion y un puntero al siguiente nodo, tien otro struct para representar la arista y de donde viene y hacia donde va (y un puntero a la siguiente arista)

## Describir el concepto de colision en hashing

Una colision de da cuando se aplica la funcion hashing a dos elementos de un conjunto de dominio, y se obtiene el mismo resultado.

Esto puede pasar porque el conjunto de dominio es mayor que el conjunto imagen, siendo que la relacion entr valores del dominio es muchos a uno con la de las imagenes.

LA funcion para el hsashin deberia armarse de manera tal que a pesar de que el problema no puede eliminarse, se pueda evitar al maximo la frecuencia de esta ocurrencia, es decir que la funcion tenga buena dispersion.

## Cual es el objetivo del algoritmo de Dijkstra

El objetivo es dado un vertice de inicio encontrar el camino mas corto hacia el resto de los vertices

## Describe la búsqueda de una clave única en un árbol B.

La búsqueda de una clave única en un árbol B consiste en ubicarse en el nodo raíz del árbol.

Los pasos a seguir son:

1. Seleccionar como nodo actual la raíz del árbol. Si el árbol está vacío la búsqueda falla.
2. Comprobar si la clave se encuentra en el nodo actual. Si esta termina la búsqueda
3. Si la clave no está presente, hay 3 posibilidades a considerar:
  - Si el elemento  $x$  a buscar es menor que  $C(\text{clave})$ , entonces se continúa buscando en el subárbol  $S$ .
  - Si  $x$  es más grande que  $C_{n-1}$ , se continúa buscando en el subárbol  $S_{n-1}$ .
  - Si existe un  $i$  tal que  $1 \leq i \leq n - 1$  para el cual  $C_i \leq x \leq C_{i+1}$  entonces se continúa buscando en el árbol  $S_i$ .

## 'Qué beneficios brinda la aplicación de la normalización al diseño de un modelo de bases de datos?'

La normalización evita redundancias (datos duplicados) y a tambien problemas de actualización de los datos en la tabla, reduciendo anomalías en la manipulación de datos.

Asegura la integridad de la informacion, mostrando como se relacionan las tablas entre si

Mejora la localizacion de la informacion y tambien su seguridad

Deja datos precisos, únicos y relevantes según las necesidades del sistema.

A su vez, como el volumen de los datos disminuye se facilita y agiliza el acceso, mantenimiento y consultas a los ellos.

## Describa y cite al menos un ejemplo de una representacion computacional de un grafo irrestricto

Los grafos irrestrictos son grafos en los que no se aplica ninguna restriccion a la relacion modelada, pudiendo o no modelar relaciones que sean reflexivas, simetricas o transitivas.

Un ejemplo seria la lista de adyacencia.

Se define la lista enlazada para cada nodo, que contiene los nodos a lo que se puede acceder. El nodo 'A' va a tener una lista enlazada asociada en la que va a haber un elemento con referencia al nodo 'B' si 'a' y 'b' estan conectados por una arista. Si el grafo no es dirigido, en la lista enlazada de 'b' tambien habra una referencia al nodo 'a'

## Describa la arquitectura interna de un DBMS, niveles o capas y funcionalidad y funcionamiento de cada uno de ellos

La arquitectura de un DBMS esta conformaada por tres niveles/capas

1. Nivel interno (fisico): En este nivel se gestiona la persistencia de los datos en la DBMS. La forma fisica de los datos y los metodos de acceso
2. Nivel Conceptual (logico): la vista comun de la base de datos, donde se encuentra la logica de la DBMs. Define las reglas y es donde se encuentra el analizador sintactico y semantico. Se transforman los datos y se establecen sus relaciones a nivel logico
3. Nivel Externo (vista del usuario final): La capa con la que interactua el usuario final (sea usuario, lenguaje de programacion o aplicacion)

En esta estructura tambien se consideran transformaciones como:

Transformacion conceptual interna: define la correspondencia entre la vista conceptual y la base de datos , especificando la representacion en el nivel interno

Transformación externa/conceptual: a partir de una vista externa, implica la correspondencia con la vista conceptual

## 'La tabla de hechos en una arquitectura de Data Warehouse contiene información primaria del dominio del negocio.'

Contiene las claves primarias (PK) de cada tabla de dimension a la que se une para explotar y analizar un aspecto del negocio.

Cada tabla de dimensión analiza un aspecto de esos datos ofreciendo consultas históricas, que permiten consumir los datos de forma exploratoria sin tener que pensar queries sql, ver si compilan, etc.

Data warehouse está orientado al sujeto, a ayudarlo a que tome decisiones y pone énfasis en la historia de las operaciones de modo que es posible reconstruir hechos a partir de la información contenida.

Otra característica importante es la de integración, al ser una recolección de distintas fuentes de información (BD SQL, NOSQL, archivos..) debe consolidar la información de forma consistente para poder explorarla.

// debe ser otra pregunta esta

En su origen tenía una implementación no SQL, pero las grandes compañías de bases de datos vieron el nicho y agregaron soporte para este tipo de bases de datos multidimensionales.

Otra diferencia importante con las bases de datos operacionales es que la información aquí no está normalizada, porque su objetivo no es la performance, sino ofrecer información al cliente que le sirve para el desarrollo de su negocio

## 'Cual es el objetivo de los grafos?'

Modelizar un problema específico a través de un modelo abstracto. Los elementos que participan del problema son los nodos, y las relaciones que existen entre esos participantes son los arcos.

Pueden ser modelizados de forma Estática o Dinámica

En el caso estático, se construyen sobre estructuras rígidas que usan el concepto de contigüidad, como los vectores y matrices. Puede cambiar, pero no constantemente. Contempla la existencia de todas las relaciones posibles entre todos los vértices existentes. La representación puede ser a través de una matriz de adyacencia o de incidencia

En el caso dinámico, el espacio usado por la representación cambia en función de los cambios del grafo. Puede ser representada por la lista de adyacencia.

## Desarrolle los conceptos de OLTP y OLAP, identificando características propias y diferencias entre ambos

OLTP → modelo de base de datos relacional y transaccional que se basa en la ejecución de transacciones (implicando altas, bajas y modificaciones). Tiene un nivel de detalle que el modelo OLAP no considera. Este modelo refiere a la parte más operativa o de procesamiento de los datos de una organización. Los datos están normalizados y son volátiles.

OLAP → Modelo de base de datos relacional basado en el análisis. La aplicación consume lo que analiza de los datos, pero no los transforma ni actualiza. Trabaja con información, no datos, y se utiliza para la toma de decisiones.

Ambos son bases de datos relacionales y trabajan con lo que se almacena en ellas, pero con perspectivas distintas.

## Describir un objeto que pueda ser utilizado para la funcionalidad de integridad

La Foreign Key, que permite referenciar a una columna o grupo de columnas de una tabla a partir de una columna o grupo de columnas en otra tabla. (2pts)

## Describir un objeto que pueda ser utilizado para la funcionalidad de integridad

Los constraints pueden usarse para la integridad.

La constraint de primary key sirve para asegurar la integridad de la entidad. Obliga a cada fila a tener un valor único no nulo en un campo/conjunto de campos.

La constraint FK garantiza la integridad referencial. Todos los valores no nulos para un campo existan como PK en otra tabla indicada.

Para la integridad semántica se usan las constraints NOT NULL, CHECK, UNIQUE y DEFAULT.

## Cuales son las principales funcionalidades de un DBA

El DBA es el tecnico de informacion responsable de dirigir todas las actividades relacionadas con el mantenimiento de un entorno de base de datos exitoso. Debe asegurarse que la BD de una organizacion y sus aplicaciones funcionen funcional y eficientemente.

Ayudan a definir el mejor nivel de disenio conceptual de un motor de base de datos, en colaboracion con el desarrollador del sistema, y define tambien la mejor forma de almacenar los datos, y que se puede y no hacer a nivel programacion en funcion de la configuracion del DBMS.

## | 'Que se entiende por grafo conexo?':

Grafo es el que todo par de vertices esta conectado por un camino. Esto significa que hay al menos una conexion entre todos los nodos que conforman el grafo, sea indirecta o directa (a traves de mas de un arco entre ambos)

## | 'Qué es y cuál es la diferencia entre paso y camino?'

Un camino entre dos nodos se establece cuando existe una vinculacion directa o indirecta entre ambos, cuando estan vinculados por uno o mas arcos, mas alla del sentido.

Un paso existe cuando hay un camino, pero con un sentido preestablecido.

La diferencia esta en que el paso esta asociado a la direccion

## | Ejemplifique un caso de uso de triggers

Cuando se quiere proteger el borrado o actualizacion de uno o varios registros. En este caso se implementa un trigger que bloquee la consulta desencadenante, y hacer las validaciones necesarias antes de borrar. si no se cumple la validacion no se borra, emitiendo un mensaje, y en el caso contrario se realiza el borrado.

## | Explique 1 escenario donde un arbol b seria mas eficiente que una tabla de hash

En situaciones donde manejamos grandes volumenes de datos, ya que el Hash en situaciones donde el dominio es mayor que la imagen puede generar

colisiones. Los Arboles B son mejores tambien para rangos y accesos secuenciales.

## | utilidad de buscar un paso o camino en un grafo

LA utilidad de poder buscar un camino o paso es que partiendo de un vertice se pueda llegar a otro a traves de ciertos arcos. Se suele usar muchas veces para encontrar el camino mas rapido entre dos vertices. Asi, podemos encontrar la solucion optima.

Una aplicacion de la vida real seria encontrar como llegar desde un punto a otro en un mapa.

## | 'Desarrolle cual es la importancia de la selección del pivote en el metodo Quicksort en funcion de los diferentes conjuntos de datos a ordenar.'

De ahí marca el punto de partida para ordenar la tabla.

El orden de complejidad, y el tiempo de respuesta dependen de la seleccion del pivote. Si el pivote elegido genera un arbol desbalanceado, en el que una mitad es sustancialmente mas grande que la otra, es decir el peor de los casos, el algoritmo va a ser de orden  $O(N^2)$ . Si se eligiera un pivote que genera un arbol balanceado, o uno en el que ambas mitades son similares el roden es de  $O(n \log n)$ . Este seria el mejor de los casos.

## | 'Defina el concepto de transacción en un Sistema de Base de Datos'

Una transacción es un conjunto de operaciones sql que se ejecutan como un único bloque, si falla una operación entonces fallan todas, si una transacción tiene éxito, todas las modificaciones de los datos realizadas durante la transacción se confirman y se convierten en una parte permanente de la base de datos. Sí una transacción encuentra errores y debe cancelarse, se borran todas las modificaciones de los datos, volviendo al estado anterior previo a la transaccion.

Para el DBMS, es el equivalente a una sola instruccion.

## | 'Describir la arquitectura ANSI Sparc'



Consiste en tres niveles:

El interno / físico → almacenamiento

El conceptual / lógico (las reglas del dominio, el manejo de información)

Externo / de usuario → la presentación de lo que ve el usuario

Desde el punto de vista del almacenamiento, el nivel externo son las vistas individuales de los usuarios. Excluye datos irrelevantes, y los datos que el usuario no tiene permisos para acceder. El nivel conceptual es la forma de describir los datos que se almacenan, y como están relacionados entre sí, sin especificar cómo se almacenan físicamente los datos. Por último, el nivel interno implica la forma en la que la base de datos se representa físicamente en el sistema informático.

Desde el punto de vista de la funcionalidad, el nivel externo es el nivel en el que el usuario interactúa con el DBMS. Usuario engloba al usuario final o cualquier aplicación/entorno de programación. El nivel conceptual es donde radica la lógica del DBMS, donde se definen las reglas. Aquí está el analizador sintáctico y semántico. En el nivel interno es donde se maneja la persistencia de la información.

'Enumere y explique al menos tres algoritmos de clasificación u ordenamiento que conozca.'

Quicksort

Es un ordenamiento con intercambio de partición

Si un elemento de un vector está ubicado de tal forma que todos los que están antes que él son menores y todos los que están después son mayores ⇒ está en su posición final cuando el vector esté ordenado.

El algoritmo usa dos punteros: DOWN y UP

Y un flag que indica cuál puntero se venía decrementando/incrementando.

Se puede definir como un procedimiento recursivo.

Tiempo de ordenamiento de  $n$  elementos en el tiempo proporcional de :  $n \cdot \log(n)$

HeapSort

Objetivo → dado un conjunto de claves ingresadas en forma aleatoria, se busca ordenar el conjunto minimizando el tiempo de búsqueda

Este algoritmo consiste en almacenar todos los elementos del vector a ordenar

en un montículo (heap), y luego extraer el nodo que queda como nodo raíz del montículo (cima) en sucesivas iteraciones obteniendo el conjunto ordenado. Basa su funcionamiento en una propiedad de los montículos, por la cual, la cima contiene siempre el menor elemento (o el mayor, según se haya definido el montículo) de todos los almacenados en él.

### MergeSort

El método MergeSort es un algoritmo de ordenación recursivo con un número de comparaciones entre elementos del array mínimo.

Su funcionamiento es similar al Quicksort, y está basado en la técnica divide y vencerás.

De forma resumida el funcionamiento del método MergeSort es el siguiente:

- Si la longitud del array es menor o igual a 1 entonces ya está ordenado.
- El array a ordenar se divide en dos mitades de tamaño similar.
- Cada mitad se ordena de forma recursiva aplicando el método MergeSort.
- A continuación las dos mitades ya ordenadas se mezclan formando una secuencia ordenada.

## Describir la tabla de frecuencia de huffmann detallando sus columnas

### 5 columnas

Status: Indica si dicho caracter ya fue tomado para ponerlo en el arbol que arma el algoritmo

Caracter: muestra el caracter como lo vemos nosotros.

Frecuencia: indica la cantidad de veces que aparece el caracter en el archivo

Codigo: El codigo con el que va a sr reemplazado el caracter al momento de la compresion (en bits)

Direccion del arbol: indica si en el arbol va a la izquierda o a la derecha.

'Describa el barrido simétrico o in orden de un arbol binario, ejemplifique.'

Entreorden o simétrico:  
recorrer el subárbol izquierdo  
Informar la raíz  
recorrer el subárbol derecho

Ejemplo → Arbol binario de 3 niveles con cada nivel teniendo la maxima cantidad de nodos permitido por el arbol binario. La representacion algebraica seria ((A,B),(A,C),(B,D),(B,E),(C,F),(C,G)). El barrido daria como resultado DBEAFCG

'Describe de forma clara y concisa los terminos Datamining, Datawarehouse y OLAP y cómo se relacionan entre sí.'

//mi respuesta

Data Mining → El conjunto de tecnicas que se usan para obtener informacion implicita u oculta en bases de dato con volumenes altos, como las Data Warehouse. Buscan patrones de comportamiento entre los datos, para permitir anticipar futuros acontecimientos.

Data Warehouse → Base de datos corporativa cuya caracteristica principal es la integracion y el filtrado de informacion de una o varias fuentes, para procesarla para su analisis desde diferentes puntos de vista y con una gran velocidad de respuesta. Coleccion de datos historicos integrados. Diseñada para la toma de desiciones estrategicas

OLAP → On Line Data Processing. Un tipo de tecnologia que analiza y relaciona datos ya procesados. Procesa la informacion para convertirla en conocimiento. Almacena la informacion de forma desnormalizada y esta es persistente, es decir no volatil. Estan optimizadas para el analisis

El Data Warehouse utiliza tecnologias OLAP

'Desarrollar el concepto de JOIN, enumere y explique cada uno de los tipos. Ejemplifique'

Permite unir dos o mas tablas, expando la consulta mas alla de lo acotado por el FROM. A nivel teoria de conjuntos, es la interseccion entre dos conjuntos.

INNER JOIN solo muestra las filas que coincidan,

OUTER JOIN mostrara todas las filas de la tabla dominante matcheen o no con los datos de la otra tabla

Puede ser left, right o full

'Describir el nivel de aislamiento repeatable read y el concepto de lectura repetible '

Repeatable read → Se establecen bloqueos para todos los datos usados en la consulta, impidiendo que otros usuarios la actualicen. Permite insertar nuevas filas fantasmas en los datos que otro usuario establezca, para que se incluyan en lecturas posteriores de la misma transacción.

Su nivel de aislamiento es inferior al predeterminado porque permite la simultaneidad.

'Mencionar que funcionalidades se pueden usar en un stored procedures y que no se pueden hacer en una función.'

En una función no se pueden invocar procedimientos, ni pueden ser invocadas por aplicaciones externas.

Los stored procedures pueden modificar el estado de los datos, y no retornan valores sino true o false según se la haya podido ejecutar.

Enumere las ventajas y desventajas de una implementación estática y dinámica de grafos

Ventajas de la implementación estática de grafos:

- son más rápidos en búsquedas porque son posiciones contiguas de memoria que son de rápido acceso

- Ocupan menos espacio, solo reservan espacio para una cantidad determinada de elementos con el tamaño del dato a guardar.

- Son útiles cuando el grafo no cambia o no crece, ya que se reserva el espacio una vez

Desventajas de la implementación estática:

- Si se reserva un tamaño menor al necesario no se puede agregar, y si se reserva más espacio del necesario este espacio es desperdiciado

- El tamaño es predefinido

- Insertar un elemento puede implicar mover varios elementos del grafo.

## Ventajas de implementacion dinamica

Se pueden seguir agregando elementos

Son utiles cuando el grafo es dinamico, crece o cambia constantemente.

Se libera la memoria si un nodo es eliminado.

Insertar un elemento implica mover solo los punterios adyacentes al nodo

## Desventajas de la implementacion dinamica

Ocupa mas espacio, por cada nodo debo reservar espacio para uno o varios punterios hacia el y los siguientes nodos, y eventualmente hacia un nodo anterior.

'Enumere una forma de implementación estáticos sobre grafos y describalo.'

## Matriz de adyacencia o incidencia

### Matriz de Adyacencia

Dado un grafo  $G = (V, A)$  con  $n$  vértices  $\{V_1, \dots, V_n\}$  su matriz de adyacencia es la matriz de orden  $n \times n$ ,  $A(G) = (A_{ij})$  donde  $A_{ij}$  es el número de aristas que unen los vértices  $V_i$  y  $V_j$ .

### Matriz de incidencia :

Dado un grafo simple  $G = (V, A)$  con  $n$  vértices  $V = \{V_1, \dots, V_n\}$  y  $m$  = aristas  $A = \{A_1, \dots, A_m\}$ , su matriz de incidencia es la matriz de orden  $n \times m$ ,  $B_{ij}$  donde  $B_{ij}=1$  si  $V_i$  es incidente con  $A_j$  y  $B_{ij}=0$  en caso contrario.

'En que difieren las DB multidimensionales con las relacionales: ventajas y desventajas'

la principal diferencia que tienen las bases de datos multidimensionales frente a las relacionales es que el objeto de las primeras es agilizar las consultas en grandes cantidades de datos, mientras que el objetivo de las bases de datos relacionales es asegurar las propiedades ACID en las transacciones que recibe.

### **Ventajas multidimensionales:**

- consultas agiles en grandes volumenes de datos
- capacidades analiticas (consulta por particiones, calculo de percentiles, etc)
- segmentar los datos segun un criterio propuesto por el usuario.

**Desventajas multidimensionales:**

- es imposible realizar cambios de estructuras

**Ventajas relacionales:**

- propiedades ACID
- flexibilidad en los modelos de datos pese a las reglas normales

**Desventajas relacionales:**

- los joins entre tablas pueden ser muy costoso
- los datos deben ser estructurados

## | Describir las ventajas y desventajas de los índices

**Ventajas:**

Permiten mantener los datos ordenados en base a una clave

Al acceder por clave indexada, se minimizan las lecturas a disco, que es una operacion costosa

Disminuye el tiempo que se necesita para hacer una consulta porque se evita hacer fullscans sobre la tabla. (EJemplo → si es un indice de arbol b+, con hacerse un count sobre la tabla indexada es suficiente sin hacer el fullscan)

Distintos tipos de índices resuelven distintos tipos de necesidades. Hashing es mas eficiente para búsquedas por valores exactos y Arbol B es mas eficiente con búsquedas por rango

**Desventajas:**

Cada operacion de insert/delete/update puede llegar a requerir mas tiempo de ejecucion porque debe actualizarse el índice tambien

Los índices son estructuras adicionales a las tablas por lo que se ocupa un espacio adicional y mas memoria

Como la actualización de los índices es en serie, en dos sesiones no se puede actualizar un mismo índice simultáneamente.

Cuando se actualiza un índice se bloquean las consultas a la tabla conectada a ese índice.

## Mencionar las diferencias entre una view y un snapshot o vista materializada

Una vista es una tabla virtual creada en base a una consulta. Esa consulta va a ser ejecutada cada vez que se haga una consulta sobre la vista, por lo que siempre disponemos de los datos referenciados por la consulta con los valores actualizados.

Un snapshot es un objeto también creado en base a una consulta, pero además almacena el resultado de la consulta en el momento de su creación. El snapshot no devuelve resultados actualizados en el caso de que se modifique la tabla sobre la que se hizo la consulta original, salvo que se refresque el snapshot.

## 'Escribir método de ordenamiento de merge sort:'

Algoritmo de ordenamiento recursivo, el método se basa en dividir en dos subconjuntos iguales la secuencia principal, y estas dos partes se vuelven a dividir y los subconjuntos se dividen nuevamente. Así recursivamente hasta que la diferencia entre los grupos sea indivisible.

Una vez hecho esto se empieza a ordenar las secuencias ya divididas y se las une (merge) de forma ordenada. Tiene la complejidad computacional de  $O(n \log n)$ , por lo que ese sería el tiempo límite de espera, más allá de que tan desordenada esté la entrada. El aspecto negativo es que no es un algoritmo in situ, necesita memoria extra para formar las subsecuencias que luego se reordenan y unen.

## 'Explique el objetivo de los índices y sus características.'

Los índices son estructuras opcionales asociadas a una tabla. Busca permitir mantener los datos en función a una clave, agilizando el acceso a los datos de una tabla.

Los índices son lógicamente y físicamente independientes de los datos en la tabla asociada. Se puede crear o borrar un índice en cualquier momento sin afectar a las tablas base o a otros índices. A su vez, no garantizan la unicidad de las claves.

### TIPOS DE INDICES

Btree Index → Estructura de índice estándar y más utilizada.

Hash Index (MySQL) → Están implementados en tablas de hash y se basan en otros índices Btree existentes para una tabla. Si una tabla entra íntegramente en memoria, la manera más rápida de ejecutar consultas sobre ella es usando un hash index.

Functional Index / Function based Index

Son índices cuya clave deriva del resultado de una función.

En general las funciones deben ser funciones definidas por un usuario.

Reverse Key Index (Oracle)

Invierte los bytes de la clave a indexar. Esto sirve para los índices cuyas claves son una serie constante con por ej.

Crecimiento ascendente. para que las inserciones se distribuyan por todas las hojas del árbol de índice.

Explique el concepto de durabilidad en transacciones ACID y de un ejemplo donde no se cumpliría y sería válido

La durabilidad en una transacción ACID implica que el conjunto de operaciones que se ejecuta de forma atómica (como única unidad), una vez que fue aceptada y plasmada en la base de datos mediante un commit no puede deshacerse más allá de que falle el sistema, porque esta fue replicada a nivel disco en forma de registros.

Esto significa que una vez que una transacción es comprometida podemos encontrar sus resultados alojados en la tabla de registros de forma inmediata

Un ejemplo donde la durabilidad no se da pero que sea válido a pesar de eso sería si durante la transacción se llega a un estado donde el dominio lógico determine que se desarrolló una situación errónea, por lo que se realiza un 'rollback transaction', lo que genera que todos los cambios realizados dentro de la transacción hasta el momento sean eliminados. En un caso así, la transacción no fue persistida y no hubo durabilidad

Describir el método de ordenamiento Heap Sort

Se basa en una estructura de datos 'heap', es decir un montículo binario, que permite implementar una cola de prioridad usando un árbol binario con la regla de que la clave de cada nodo tiene que ser mayor o igual a las claves de sus hijos. Así genera que, una vez ordenado la clave más grande está siempre en la raíz.



Tiene dos fases. Primero se ordena el árbol binario, formando el montículo. Después, se desarma el montículo y se obtienen los valores ordenados a partir de la raíz de mayor a menor.

El orden de complejidad máximo del algoritmo es  $O(n \log n)$ .

## Describir la funcionalidad de la consistencia en las bases de datos relacionales

La consistencia es una de las cuatro características que componen las siglas ACID.

La consistencia asegura que solo se empieza aquello que pueda terminarse, se pasa de un estado íntegro a otro estado íntegro. Es decir, asegura que se ejecuten aquellas operaciones que no van a romper reglas y directrices de la integridad de la base de datos.

Busca garantizar que la información que se presenta al usuario sea siempre la misma.

## 'Defina el concepto de transacción en un sistema de base de datos'

Una transacción es un conjunto de operaciones que se ejecutan como una única unidad.

Estas transacciones deben cumplir 4 propiedades fundamentales conocidas como ACID → atomicidad, consistencia, aislamiento y durabilidad

Atomicidad: cualquier cambio que produce una transacción es atómico, o sea, se ejecutan todas las operaciones o no se ejecuta ninguna.

Consistencia: asegura que una transacción no rompa la integridad de la base de datos

Aislamiento: asegura que no se afectaran entre sí las transacciones que se ejecuten de manera concurrente

Durabilidad: asegura la persistencia de una transacción, o sea, una vez que la transacción queda aceptada no podrá deshacerse aunque falle el sistema

## 'Explicar los conceptos de Data Warehouse y Data Mining y el objetivo de utilizar cada uno de ellos.'

DW: son sistemas donde se almacenan y extraen datos de diversas fuentes para la toma de decisiones.

DM: es la extracción de información oculta y predecible de grandes BDs, ayuda a las compañías a concentrarse en la información más importantes de sus bases y predecir nuevas tendencias.

'Explique y desarrolle los diferentes niveles de aislamiento de una base de datos relacional.'

Niveles de aislamiento:

**Read uncommitted:** no asegura lockeos por select, lo que mejora el rendimiento pero afecta la integridad.

no lockeos por select ⇒ mejor rendimiento

si lecturas sucias

si lectura no repetibles

si lectura fantasma fantasmas

Es el menos restrictivo

**Read committed:** asegura que no exista lecturas sucias pero no asegura lecturas repetibles, ya que una vez que leyó los datos, libera el lockeo. En una misma transacción puede tener dos llamados a un mismo select y este arrojar resultados distintos.

no lecturas sucias

si lecturas no repetibles, si datos fantasma

**Repeatable read:** asegura que no existan lecturas sucias y que las lecturas puedan ser repetibles, pero no evita lecturas fantasmas.

no lecturas sucias

no lectura no repetible

si lectura fantasma

**Serializable read:** asegura que no existan lecturas sucias, lecturas fantasmas y que las lecturas puedan ser repetibles. El problema es que se implementa un nivel de bloqueo que puede afectar a los demás usuarios.

no lectura sucia

no lectura no repetible

no lectura fantasma  
tantos bloqueos pueden afectar a los demás usuarios

'Explique las propiedades ACID de una BD y su relación con las transacciones.'

Las propiedades ACID garantizan que las transacciones se realicen de forma confiable en una BD.

Atomicidad: la transacción debe ejecutarse en su totalidad o no debe ejecutarse en absoluto.

Consistencia: luego de la transacción, la BD debe quedar consistente satisfaciendo

todas las restricciones de integridad.

Aislamiento (isolation): cuando las transacciones se ejecutan concurrentemente, sus efectos deben ser aislados y no deben interferirse entre sí.

Durabilidad: una vez que la transacción termina, su efecto no debe perderse aunque el sistema falle.

'Defina el concepto de transacción, cite un ejemplo de cuando el motor de BD ejecutaría un Rollback en una transacción.'

Transacción es un conjunto de instrucciones las cuales deben ejecutarse todas o ninguna, manteniendo la integridad de los datos.

Un Rollback puede ejecutarse en el caso de que se quiera borrar una fila que está relacionada con otra tabla por medio de una FK, por lo cual fallaría. Esto en el caso que no esté configurada en "Delete on cascade".

En que consiste el grado de un grafo

Consiste en la cantidad de arcos que se relacionan con un vertice en específico.

Puede ser positivo (la cantidad de arcos que Salen) o negativo (la cantidad de arcos que llegan al vertice)

'Mencione dos objetos que tengan que ver con la seguridad, descríbalos e indique de qué modo puede utilizarlos para dicha funcionalidad.'

Objetos relacionados con la seguridad:

Vistas: es una consulta que se presenta como una tabla (virtual). Se puede por ej, para algunos usuarios crear una vista de una tabla donde obtenga solo parte de las cantidad real de columnas que tiene la tabla.

Triggers: son objetos que se relacionan a tablas, y permiten administrar mejor la BD. Se puede por ej, crear un trigger en una tabla que ante un delete de n filas, no haga caso al delete, o escriba un histórico.

Stores procedures: es un conjunto de instrucciones que se almacenan y ejecutan en la BD. Se puede por ej, crear un SP para obtener un listado de los usuarios que accedieron a determinadas tablas, horarios, etc.

'Defina el concepto de lectura sucia, repetible y fantasma.'

**Lectura sucia:** Cuando a una transacción se le permite la lectura de una fila que fue modificada por otra transacción concurrente pero que no ha sido cometida todavía

**Lectura repetible:** ocurre cuando en el curso de la transacción se lee una fila dos veces, y los valores coinciden.

**Lectura no repetible:** ocurre cuando en el curso de la transacción se lee una fila dos veces, y los valores no coinciden.

**Lectura fantasma:** se produce cuando durante una transacción se ejecutan dos consultas idénticas y los resultados de la segunda son distintos a los de la primera.

Relacione y explique los conceptos de transacción, lockeos y aislamiento

— armada en base a varias respuestas chequeadas

Las transacciones son un bloque de sentencias SQL que están delimitadas por el inicio de la transacción, y su fin. El bloque debe cumplir con las propiedades

ACID (atomicidad, consistencia, aislamiento y durabilidad)

Los niveles de aislamiento son utiles cuando se realizan transacciones concurrentes, en los que se quiera leer y escribir simultaneamente sobre un mismo dato o conjunto de datos.

Se puede determinar el nivel de aislamiento con el que van a trabajar las transacciones de la forma read uncommitted, read committed, repeatable read y serializable read.

Read Uncommitted: No asegura lockeos por select. Esto mejora el rendimiento pero afecta la integridad. Tiene lecturas sucias, lecturas no repetibles y lecturas fantasmas. Este nivel no tiene bloqueos exclusivos ni compartidos

Read Committed: Asegura que no existan lecturas sucias pero no asegura lecturas repetibles. Libera el lock despues de leer los datos ya que mantiene bloqueos compartidos para evitar lecturas no actualizadas, pero se pueden modificar los datos antes del final de la transaccion

Repeatable Read: Asegura que no existan lecturas sucias y que las lecturas puedan ser repetibles, pero no evita lecturas fantasmas ya que establece un bloqueo para los datos usados en la transaccion, no dejando que otros usuarios actualicen pero permite insertar nuevas filas fantasmas.

Serializable Read: Asegura que no existan lecturas sucias, lecturas fantasmas y que las lecturas sean repetibles. Su problema es que se implementa un nivel de bloqueo que puede afectar a los demas usuarios, ya que impide que otros usuarios actualicen o inserten filas en el conjunto de datos hasta que la transaccion finalice. Es el nivel mas restrictivo

Los lockeos se usan para cumplir con los diferentes niveles de aislamiento. Pueden ser a nivel BD, nivel Tabla, nivel Pagina, nivel Fila o nivel Clave de Indice, y de tipo compartido, exclusivo o promovible.

A mayor nivel de aislamiento, menos efectos de simultaneidad pero se requieren mas recursos del sistema y se aumentan las posibilidades de que una transaccion bloquee a otra

'Que es un Data Mart, explique por qué motivo posee datos desnormalizados.'

Un Data Mart es un subconjunto de un Data Warehouse, con el propósito de ayudar en un área específica dentro del negocio para que se puedan tomar

mejores decisiones. Los datos en un Data Mart están desnormalizados para disminuir la cantidad de tablas necesarias, reduciendo costos

Donde el data warehouse es el almacén de datos integrados, los Marts son las vistas multidimensionales de cada área.

| 'Explique el concepto de transacción y su relación con las propiedades ACID.'

Una transacción es un conjunto de órdenes que se ejecutan atómicamente. Las propiedades ACID garantizan que las transacciones se realicen de forma confiable en una BD. (ver 3)

| 'Explique Algoritmo de Huffman, en que se basa para la compresión de datos.'

Es un algoritmo de compresión de datos sin pérdida, muy eficiente con los archivos de texto.

Huffman se basa en asignar códigos de distinta longitud de bits a cada uno de los caracteres distintos de un archivo.

Si se asignan códigos más cortos a los caracteres que aparecen más a menudo se consigue una mejor compresión del archivo.

| 'Detalle las estructuras del algoritmo de Huffman'

El algoritmo de Huffman se usa para la compresión o encriptación de datos mediante el estudio de la frecuencia de aparición de caracteres. Se basa en el uso de un árbol binario donde las hojas representan los símbolos del conjunto de entrada.

Para conseguir el código de Huffman asociado a cada símbolo únicamente hay que seguir las aristas que unen la raíz con la hoja determinada.

Estructuras utilizadas

Tabla de Huffman: Es una matriz compuesta por 5 columnas (estado, carácter, frecuencia, código, dirección en el árbol) y una fila por cada carácter distinto.

Un árbol binario completo compuesto por el carácter, la frecuencia, punteros a

dos hijos y un puntero al padre.  
Una pila.

## | 'Detallar el objeto BD Constraint y su relacion con integridad'

### Integridad de Entidad

La integridad de entidades es usada para asegurar que los datos pertenecientes a

una misma tabla tienen una única manera de identificarse, es decir que cada fila de cada tabla tenga una primary key capaz de identificar unívocamente una fila y esa no puede ser nula

**PRIMARY KEY CONSTRAINT:** Puede estar compuesta por una o más columnas, y deberá representar unívocamente a cada fila de la tabla.

No debe permitir valores nulos (depende del motor de base de datos).

### Integridad Referencial

La integridad referencial es usada para asegurar la coherencia entre datos de dos tablas.

**FOREIGN KEY CONSTRAINT:** Puede estar compuesta por una o más columnas, y estará referenciando a la PRIMARY KEY de otra tabla.

Los constraints referenciales permiten a los usuarios especificar claves primarias y foráneas para asegurar una relación PADRE-HIJO (MAESTRO-DETALLE).

### Integridad Semántica

La integridad semántica es la que nos asegura que los datos que vamos a almacenar tengan una apropiada configuración y que respeten las restricciones

definidas sobre los dominios o sobre los atributos.

- DATA TYPE
- DEFAULT
- UNIQUE
- NOT NULL
- CHECK

### Integridad Semántica

**DATA TYPE:** Este define el tipo de valor que se puede almacenar en una columna.

**DEFAULT CONSTRAINT:** Es el valor insertado en una columna cuando al insertar un registro ningún valor fue especificado para dicha columna. El valor default por default es el NULL.

Se aplica a columnas no listadas en una sentencia INSERT.

El valor por default puede ser un valor literal o una función SQL (USER, TODAY, etc.)

Aplicado sólo durante un INSERT (NO UPDATE).

UNIQUE CONSTRAINT: Especifica sobre una o más columnas que la inserción o actualización de una fila contiene un valor único en esa columna o conjunto de columnas.

NOT NULL CONSTRAINT: Asegura que una columna contenga un valor durante una operación de INSERT o UPDATE. Se considera el NULL como la ausencia de valor.

CHECK CONSTRAINT: Especifica condiciones para la inserción o modificación en una columna. Cada fila insertada en una tabla debe cumplir con dichas condiciones.

Actúa tanto en el INSERT, como en el UPDATE.

Es una expresión que devuelve un valor booleano de TRUE o FALSE.

Son aplicados para cada fila que es INSERTADA o MODIFICADA.

Todas las columnas a las que referencia deben ser de la misma tabla (la corriente).

No puede contener subconsultas, secuencias, funciones (de fecha, usuario) ni pseudocolumnas.

Todas las filas existentes en una tabla deben pasar un nuevo constraint creado para dicha tabla. En el caso de que alguna de las filas no cumpla, no se podrá crear dicho constraint o se creará en estado deshabilitado.

## | 'Que es una tabla temporal'

Son tablas creadas cuyos datos son de existencia temporal.

No son registradas en las tablas del diccionario de datos(se refiere a la lista de tablas que tengo).

No es posible alterar tablas temporarias. Si eliminarlas y crear los índices temporales que necesite una aplicación.

Las actualizaciones a una tabla temporal podrían no generar ningún log transaccional si así se configurara.

## | 'Explique las Reglas de Integridad según el Modelo Relacional'



Regla de integridad de la entidad: ninguno de los atributos que componen la clave primaria puede ser nulo.

Regla de la integridad referencial: si en una relación hay una clave foránea, sus valores deben coincidir con los valores de la clave primaria.

'Describe ventajas y desventajas de cada uno de los métodos de creación de índices, Árbol B y Tablas de Hashing.'

Hashing: busca establecer una relación directa entre el valor de los datos y el valor de la clave con una función hash, que aplicada a una clave devuelve el subíndice de la tabla.

Ventajas: acceso directo a los datos (más rápido que Árbol B para acceso a datos). No se utiliza espacio extra para su implementación.

Desventajas: el principal problema es cuando la función hash devuelve un valor igual para dos o más claves, lo que se llama colisión.

Cuanto más datos, más colisiones. No es bueno para búsquedas secuenciales.

Árbol B: la estructura de Árbol B parte del concepto de árboles n-arios de búsqueda. Está pensado para disminuir la cantidad de accesos a disco y la posibilidad de mantener en memoria la parte que se está utilizando y el resto conservarlo en disco.

Ventajas: se utiliza para grandes volúmenes de datos y es mejor que hash para archivos secuenciales.

Desventajas: es más lento que Hashing para la creación de índices, dado que es necesario crear toda la estructura en memoria.