

Preguntas de Parciales

1. **¿Qué entiende por grafo conexo?** Un grafo conexo es aquel donde para todo par de nodos existe al menos un camino/conexión. O sea, de cualquier nodo existe al menos un camino para llegar a cualquier otro. Todos los nodos están conectados al menos un conjunto de aristas.
2. **Cuales el concepto, la funcionalidad, y para que se utilizan los Triggers en una Base de Datos Relacional**

Pregunta 4

Finalizado

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

Cual es el concepto, la funcionalidad y para que se utilizan los Triggers en una Base de Datos Relacional

Los triggers o disparadores son una herramienta del motor dentro de la parte de transact SQL que permiten introducir lógica al motor y desligarlo de la capa de lógica del sistema en casos donde dicho sistema fuese de muchos ingresos/egresos, es decir, mucha información. Son un conjunto de sentencias que se asignan y ejecutan automáticamente a la ocurrencia de cierto evento que suceda en el motor, puede ser una inserción, eliminación y/o actualización asociada a una tabla que puede ser antes de que suceda o, en el caso de SQL server, en vez de que suceda el evento o luego de la ocurrencia. En instead of se busca validar registro por registro para por ejemplo si debiesen insertarse todos o ninguno, validar que todos cumplan, en el after el evento ya sucedió y apunta a realizar una acción/actividad secundaria, referida a otra cosa respecto al evento del trigger aunque también puede utilizarse para rollbackear. Una de sus usos puede ser mantener la integridad del sistema por ejemplo al intentar eliminar algún registro siendo validado previamente por un trigger de eliminación instead of. Los disparadores se utilizan para no repetir código, no tienen parámetros y pueden modificar datos (en el caso de modificar un registro a insertar o actualizar). Funcionan con dos tablas virtuales asociadas que se asemejan a la estructura de la tabla donde se crea el trigger, una tabla inserted para los datos que se vayan a insertar tanto en un insert como en un update y la tabla deleted para los datos que se fuesen a borrar en un delete o update. Si solo la tabla inserted estuviera con datos sería un trigger de insert, si ambas tablas lo están un update, y si solo deleted sería una eliminación.

Al igual que el resto de objetos del motor, para crearse se instancia como `CREATE TRIGGER nombre ON [unaTabla] AFTER/INSTEAD OF/FOR [INSERT-DELETE-UPDATE] ...`

Pregunta 3

Finalizado

Puntúa como 4,00

🚩 Marcar pregunta

Indique un escenario donde la **creación de índices** con Tablas de Hashing sea más eficiente que con **Árboles B**

La creación de índices con tablas de Hashing es más eficiente que con árboles B cuando se trabaja por ejemplo en un mainframe. Puesto que al tratarse de una estructura estática, se tiene reservada la memoria exacta que se va a necesitar usar en el caso de que se utilice una estrategia para evitar colisiones de direccionamiento abierto, o con un extra en caso de que se utilice una estrategia de encadenamiento. Sin embargo, ya se tiene reservada dicha memoria de antemano porque se sabe con exactitud cuánta se va a necesitar, por lo que se requiere de menor interacción con el sistema operativo para solicitar recursos y menor acceso a disco, con lo cual su ejecución se vuelve más rápida en contraste con la de un Árbol B que reserva memoria en varias ocasiones por tratarse de una estructura de comportamiento dinámico y pensada para máquinas con necesidades dinámicas de asignación de recursos. Por esta razón es más eficiente utilizar tablas de hashing en dicho escenario.

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

La normalización evita todo tipo de redundancias (duplicaciones de datos) y a su vez, evita problemas de actualización de los datos en la tabla y protege la integridad de los mismos evitando o reduciendo anomalías en la manipulación de datos; dejando de esta forma datos precisos, únicos y relevantes según las necesidades del sistema. Al disminuir el volumen de los datos, facilita y agiliza notablemente el acceso, el mantenimiento y consultas a los mismos. También reduce el impacto de los cambios en los datos.

Indique claramente en qué caso la creación de índices con tablas de hashing sea más eficiente que con árboles B

La creación de índices con tablas de hashing puede ser más eficiente que con árboles B en los siguientes casos:

Búsqueda exacta: Si se necesita realizar búsquedas exactas, es decir, buscar un valor específico en el índice, las tablas de hashing son más eficientes. En una tabla de hashing, el tiempo de búsqueda es constante $O(1)$, ya que se accede directamente a la posición calculada mediante la función hash. En contraste, en un árbol B, la búsqueda tiene una complejidad de tiempo promedio de $O(\log n)$, donde n es el número de elementos en el árbol.

Inserciones y eliminaciones frecuentes: Si se requiere realizar muchas inserciones y eliminaciones de elementos en el índice, las tablas de hashing son más eficientes. La inserción y eliminación en una tabla de hashing también tienen un tiempo promedio constante $O(1)$ en comparación con el tiempo promedio logarítmico $O(\log n)$ en un árbol B.

Espacio de almacenamiento: Las tablas de hashing suelen requerir menos espacio de almacenamiento que los árboles B. Esto se debe a que las tablas de hashing no necesitan estructuras adicionales, como nodos y punteros, para organizar los datos. En cambio, utilizan directamente una matriz o vector para almacenar los elementos.

Es importante tener en cuenta que los árboles B también tienen sus ventajas en ciertos escenarios, especialmente cuando se necesita soportar búsquedas por rango, mantener los datos ordenados o realizar operaciones de fusión eficientes. Por lo tanto, la elección entre tablas de hashing y árboles B dependerá de las características específicas de la aplicación y los requisitos del sistema.

Relacione y explique los conceptos de transacción, bloqueos y aislamiento

Una transacción es una acción o conjunto de acciones que no debe quedar por la mitad, es la característica de una base de datos que permite mantener la integridad y consistencia de la misma. Se ejecutan atómicamente, es decir, como una unidad. O se ejecutan todas las instrucciones, o no se ejecuta ninguna.

Los bloqueos son bloqueos de acceso que se aplican sobre las tablas o las filas de estas, previenen problemas de concurrencia, es decir, del acceso de dos usuarios a la vez a la misma fila o tabla, de manera que la información con la que trabaja cada usuario sea consistente en cada momento.

El aislamiento consiste en aplicar bloqueos sobre una tabla o alguna/s fila/s con cierto nivel. Existen cuatro tipos de bloqueos.

De menor a mayor bloqueo (son incrementales):

Read Uncommitted:

Permite leer tablas que tienen transacciones trabajando en ellas que aún no han sido committeadas (no han sido confirmadas como terminadas para la base de datos), es decir, permite la lectura de datos que podrían tener dirty reads o phantom records.

Dirty read: Datos que no están consistentes, es decir, de una lectura a otra, se podría obtener diferentes resultados en los datos.

Phantom Records: Filas que no estaban en una primera consulta a la tabla, pero aparecen en la segunda.

Read Committed:

Únicamente permite leer los datos de tablas cuyas transacciones hayan finalizado, en su último estado de consistencia, es decir, si aún no se committeó la transacción, se visualizarán los datos "antiguos" pero no así los que la transacción esté modificando en el momento, una vez terminada y committeada la transacción, ya entonces sí se podrán visualizar los datos actualizados.

Sin embargo, dentro de la misma transacción, si podría obtenerse dos lecturas consecutivas con resultados distintos, si otra transacción hubiera generado cambios en los datos con los que la primera trabaja.

Repeatable Read:

Permite además de lo que permite el Read Committed, que la transacción no lea datos que hayan sido modificados durante su ejecución por otra transacción, se deshace de los dirty reads. Pero no así de los phantom records.

Serializable:

Permite lo mismo que el repeatable read pero además añade la cualidad de no retornar phantom records, por lo que la información es lo más consistente posible.

Comentario:

CUANDO LOS DATOS TIENEN UNA DISTRIBUCION TAL QUE NO IMPACTAN LAS ACOLISIONES, Y ADEMAS CUANDO LA TABLA TIENDE A MANTER LOS MISMO VALORES NO CRECIENDO MUCHO

'Describa 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, si la clave se encuentra ahí se termina la búsqueda, caso contrario, se selecciona entre los hijos al que se encuentra entre dos valores de clave que son menor y mayor que el buscado respectivamente y se repite el proceso hasta encontrarlo. En el caso que se llegue a una hoja y no pueda continuar la búsqueda de la clave, entonces esa clave no se encuentra en el árbol.

Enumerando, los pasos a seguir son:

Seleccionar como nodo actual la raíz del árbol. Si el árbol está vacío la búsqueda falla.

Comprobar si la clave se encuentra en el nodo actual:

Si la clave está presente, se termina la búsqueda.

Si la clave no está presente, hay 3 posibilidades a considerar:

- Si el elemento x a buscar es menor que K_1 (clave) , entonces se continúa buscando en el subárbol T_0 .
- Si x es más grande que K_{n-1} , se continúa buscando en el subárbol T_{n-1} .
- Si existe un i tal que $1 \leq i \leq n - 1$ para el cual $K_i \leq x \leq K_{i+1}$ entonces se continúa buscando en el árbol T_i .

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

Nro de cliente, edad, ingresos, variando según las necesidades de los sujetos que van a consumir la base.

Contiene las primary keys a partir de las cuales cada dimension table se joina para explotar un aspecto del negocio. Cada tabla de dimensión explota 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. Esto es porque el concepto de Data warehouse está orientado al sujeto, a ayudarlo a que tome decisiones y pone énfasis en la historia de las operaciones por lo cual se dice que es variante en el tiempo, 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.

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

La representación computacional puede ser estática o dinámica.

Las representaciones estáticas pueden ser:

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

Matriz de Incidencia: Las filas son vértices y las columnas aristas. Si el grafo es no está 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 dinámicas 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)

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

Nro de cliente, edad, ingresos, variando según las necesidades de los sujetos que van a consumir la base.

Contiene las primary keys a partir de las cuales cada dimension table se joina para explotar un aspecto del negocio. Cada tabla de dimensión explota 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. Esto es porque el concepto de Data warehouse está orientado al sujeto, a ayudarlo a que tome decisiones y pone énfasis en la historia de las operaciones por lo cual se dice que es variante en el tiempo, 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. 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?' Los grafos permiten modelizar matematicamente relaciones. En la computación se utiliza para modelar estructuras de datos no solo estructuras de datos

'Que se entiende por grafo conexo?': grafo que para cualquier par de vertices existira un camino que los conecte entre si

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

El paso es de un vertice a otro mientras que el. Un camino simple es aquel que no repite vertices en su camino. Paso/camino: hay paso entre el nodo x e y si se puede llegar de x a y. Un nodo siempre tiene paso a sí mismo. Paso simple: un paso es simple si se pasa por todos nodos distintos

'Qué beneficios brinda la aplicación de la normalización a diseño de un modelo de base de datos?'

La capacidad de normalizar datos para evitar o reducir anomalías de datos, redundancia de datos y duplicaciones de datos, al tiempo que se mejora la integridad de los datos, la han convertido en una parte importante del conjunto de herramientas del desarrollador de bases de datos durante muchos años. Ha sido una de las características del modelo de datos relacional.

'Describa la búsqueda de una clave única en un árbol B.'

Localizar una clave en un B-árbol es una operación simple pues consiste en situarse en el nodo raíz del árbol, si la clave se encuentra ahí hemos terminado y si no es así seleccionamos de entre los hijos el que se encuentra entre dos valores de clave que son menor y mayor que la buscada respectivamente y repetimos el proceso hasta que la encontremos. En caso de que se llegue a una hoja y no podamos proseguir la búsqueda la clave no se encuentra en el árbol. En definitiva, los pasos a seguir son los siguientes:

Seleccionar como nodo actual la raíz del árbol.

Comprobar si la clave se encuentra en el nodo actual:

Si la clave está, fin.

Si la clave no está:

Si estamos en una hoja, no se encuentra la clave. Fin.

Si no estamos en una hoja, hacer nodo actual igual al hijo que corresponde según el valor de la clave a buscar y los valores de las claves del nodo actual (i buscamos la clave K en un nodo con n claves: el hijo izquierdo si $K < K_1$, el hijo derecho si $K > K_n$ y el hijo i-ésimo si $K_i < K < K_{i+1}$) y volver al segundo paso.

'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. Tomar un elemento cualquiera como pivote tiene la ventaja de no requerir ningún cálculo adicional, lo cual lo hace bastante rápido. Sin embargo, esta elección «a ciegas» siempre provoca que el algoritmo tenga un orden de $O(n^2)$ para ciertas permutaciones de los elementos en la lista.

'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 en cambio si una transacción encuentra errores y debe cancelarse, se borran todas las modificaciones de los datos.

'En qué difieren las Bases de Datos Multidimensionales con las relaciones. Ventajas y desventajas.'

'Describir la arquitectura ANSI Sparc'

Modelo ANSI/SPARC

Nivel externo o VISTAS: percepción de los usuarios respecto de la DB.

Nivel conceptual o LÓGICO: representa de forma entendible la información contenida en una base de datos. Se define mediante un esquema conceptual que se escribe en DDL.

Nivel interno o Físico: cómo se almacenan los datos en disco, representación a bajo nivel de toda la base de datos.

Transformación externa/conceptual:

Dada una determinada vista externa, esta transformación define la correspondencia entre dicha vista externa y la vista conceptual.

Transformación conceptual/interna

Define la correspondencia entre la vista conceptual y la base de datos almacenada, y especifica la representación en el nivel interno de las filas y columnas del modelo conceptual.

'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 \Rightarrow 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 \rightarrow 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.

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

Entreorden o simétrico:

Visitar el subárbol izquierdo

Informar la raíz

Visitar el subárbol derecho

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

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

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, righth o full

'Que entiende por grafo conexo'

Grafo conexo: un grafo se considera conexo si todo par de vértices está conectado por un camino, o sea,

si para cualquiera par de vértices existe al menos un camino posible entre ellos, o dicho de otra forma que existe

al menos una conexión entre todos los nodos que conforman el grafo, sea esta directa (a través de un arco entre ambos) o indirecta (a través de más de un arco entre ambos).

'Describir el nivel de aislamiento repeat able read y el concepto de lectura repetible '

Repeatable read: Lockea en los selects de todos los registros consultados mientras dure la transacción.

Pueden aparecer lecturas fantasmas. Y si tiene lectura repetible.

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

En una funcion no se pueden invocar procedimientos Dentro de una funcion se puede llamar a otra funcion

Se pueden usar funciones dentro de un stored procedure

'Enumere una forma de implementación estática sobre grafos y describirlo.'

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 $n \times m$, $B(G) = (b_{ij})$, 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 volúmenes de datos
- capacidades analíticas (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

'Beneficios brinda la aplicación de la normalización a diseño de un modelo de base de datos'

Una de las principales ventajas de la normalización es que evita todo tipo de redundancias, a su vez, evita problemas de actualización de los datos en la tabla y protege la integridad de los mismos.. de esta manera, se dejan los datos precisos, únicos y relevantes según las necesidades del sistema. al disminuir el volumen de los datos, facilita y agiliza considerablemente el acceso y las consultas a los mismos.

'Escribir método de ordenamiento de merge sort:'

Se trata de un algoritmo recursivo, el método se basa en dividir el conjunto de elementos a ordenar en dos. Estas dos partes se vuelven a dividir y los subconjuntos se dividen nuevamente. Así recursivamente hasta que se queden dos grupos con un size de 1 (caso base). se considerará que ambos grupos están ordenados y lo único que se hará es el "merge" entre ambas partes. se van ordenando y fusionando ambas subsecuencias hasta obtener dos conjuntos ordenados. Es entonces cuando se va a hacer un último merge para que quede el conjunto completo ordenado.

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

Los índices son estructuras opcionales asociadas a una tabla. La función de los índices es la de permitir un acceso más rápido a los datos de una tabla, se pueden crear distintos tipos de índices sobre uno o más campos.

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.

TIPOS DE INDICES

Btree Index

Estructura de índice estándar y más utilizada.

Btree Cluster Index

Este tipo de índice provoca al momento de su creación que físicamente los datos de la tabla sean ordenados por el mismo.

(Informix / SQLServer / DB2)

Bitmap Index (Oracle)

Son utilizados para pocas claves con muchas repeticiones

Cada bit en el Bitmap corresponde a una fila en particular.

Si el bit está en on significa que la fila con el correspondiente rowid tiene el valor de la clave.

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.

'Describa la búsqueda de una clave única en un árbol B'

Si necesitamos buscar un ítem X en un árbol B, debemos comenzar por la raíz..

si el árbol está vacío, la búsqueda falla, de lo contrario, las claves en el nodo

raíz son examinadas para determinar si el elemento que se está buscando está presente

si ese elemento está, la búsqueda termina exitosamente.. si no está, hay 3 posibilidades

1. si el elemento x a buscar es menor que y, se continúa buscando en el subárbol T
2. si x es más grande que Y_{n-1} , se continúa buscando en el árbol T_{n-1}
3. si existe un i tal que $1 \leq i \leq n-1$ para el cual $Y_i \leq X \leq Y_{i+1}$ entonces se continúa buscando en el árbol T_i

el tiempo de ejecución de una búsqueda exitosa depende por la profundidad en que se

encuentra el elemento a buscar dentro del árbol y en el peor de los casos está determinada

por la altura del árbol B

**'Desarrolle cual es la importancia de la selección del pivote en el método quicksort'
'en función de los diferentes conjuntos de datos a ordenar'**

una mala elección sería elegir el primer elemento como pivote, que si bien sería aceptable en una entrada aleatoria nos llevaría al peor caso si al entrada esta ya ordenada o en orden inverso.. Lo mejor sería elegir el elemento central como pivote, el que esta en la posición (izquierda + derecha /2) del array. Otra elección que sería mejor es la de elegir 3 valores (ejemplo, el primero, el del medio y el ultimo) y agarrar el menor de esos 3 como pivote.

'Defina el concepto de transaccion 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 afecten entre si las transacciones que se ejecuten de manera concurrente

Durabilidad: asegura la persistencia de una transacción, o sea, una vez que la transacción quedó 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 bloqueos por select, lo que mejora el rendimiento pero afecta la integridad porque hay lecturas sucias, lecturas no repetibles y lecturas fantasmas. no bloqueos por select => mejor rendimiento
si lecturas sucias
si lectura no repetibles
si lectura fantasma fantasmas

Read committed: asegura que no exista lecturas sucias pero no asegura lecturas repetibles, ya que una vez que leyó los datos, libera el bloqueo. 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 lectura fantasma

Repeatable read: lee los datos y libera el bloqueo dos llamados en una misma transacción puede tener resultados distintos 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.

'Explique y desarrolle 3 objetos de BD que sirven para asegurar la Integridad de datos'

Objetos relacionados con la integridad:

Constraints (unique, not null, check, default)

Triggers

Índices

Vistas (con check options)

Stores procedures

'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".

'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: ocurre cuando se le permite a una transacción hacer una lectura de una fila que ha sido modificada por otra transacción concurrente, pero aún no ha sido confirmada (commitada).

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

'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.

'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 que puede ser usado para compresión o encriptación de datos. Por ej: se usa cuando la cantidad de El espacio en disco es insuficiente o los tiempos de transmisión son prolongados con costos elevados.

Huffman se basa en asignar códigos de distinta longitud de bits a cada uno de los caracteres de un archivo. Si se asignan códigos más cortos a los caracteres que aparecen más a menudo se consigue una compresión del archivo. Esta compresión es mayor cuando la variedad de caracteres diferentes que aparecen es menor. Ej: si el texto se compone únicamente de números o mayúsculas, se conseguirá una compresión mayor.

'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 relación 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.

'Explicar Clave Primaria, Foránea, Candidata y dominio de los datos.'

Clave Primaria: es un campo o combinación de campos que identifica de forma única a cada fila de la tabla.

Clave Foránea: identifica una columna o grupo de columnas en una tabla que se refiere a una columna o grupo de columnas en otra tabla.

Clave Candidata: son claves que podrían servir como clave primaria.

Dominio de los Datos: un dominio describe un conjunto de posibles valores para cierto atributo.

'Desarrolle el concepto de restore y recovery.'

Restore: es la acción de tomar un Back Up ya realizado y restaurar la estructura y datos sobre una base dada.

Recovery: es un mecanismo provisto por los motores de base de datos que se ejecuta en cada inicio del motor de forma automática como

dispositivo de tolerancia de fallas. Sus objetivos son , retornar el el motor al punto consistente más reciente y utilizar logs transaccionales

para retornar al motor a un estado lógico consistente.

'Describa los diferentes conceptos que se relacionan directamente con la funcionalidad de concurrencia.'

Los motores permiten controlar el acceso concurrente a sus recursos a través de bloqueos y de niveles de aislamiento.

'Explique el acceso a datos por hashing into buckets, mencionar de que modo trata' 'las colisiones'

Cuando dos o más claves intentan asignarse a una misma dirección en la tabla hash, cuando dos claves se direccionan a la misma se produce una colisión. cuando hay colisiones se requiere un proceso adicional para encontrar una posición disponible

para la clave:

las cuales son:

- direccionamiento abierto: consiste en poner en la siguiente posicion vacia
- direccionamiento cerrado: no hay problema con que se repita la misma dirección.

¿Cual es el objetivo de crear la tabla de frecuencias en el algoritmo de huffman y porque su importancia?

La tabla de frecuencia es una matriz compuesta por 5 columnas (estado, carácter, frecuencia, código, dirección en el árbol) y una fila por cada carácter distinto. Su objetivo es poder ordenar los caracteres de acuerdo a su frecuencia de mayor a menor y así poder decidir cómo hacer para no repetir esos caracteres, lo cual, al aplicar el algoritmo termina logrando una compresión en base a cuánto se repiten los mismos. Esto funciona debido a que esta tabla le va a asignar códigos de distinta longitud a los caracteres de un archivo y asignándole los códigos de menor tamaño a los caracteres que más se repiten.

'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

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.

'Explicar objetos de BD que nos permiten la integridad referencial, reglas de integridad de entidad y la integridad semántica.'

Integridad semántica = Consistencia.

'Explique el concepto de Fact table y Dimensiones en un Data Warehouse.'

Fact table: es la tabla primaria en cada modelo dimensional, orientada a contener reglas de negocio. Cada Fact table representa una relación de muchos a muchos, y contiene un conjunto de dos o más foreigns keys que hacen referencia a sus respectivas tablas de dimensión.

Dimensiones en DW: llamamos dimensiones a aquellos datos que nos

permiten agrupar o filtrar información. Existe una Dimensión Table que restringe los criterios de selección de la Fact table. Cada dimensión está definida por su clave primaria que sirve como base para la integridad referencial con la fact table.

'Explique la relación existente entre la Constraints UNIQUE y PRIMARY KEY y el Objeto índice.'

Constraint Unique: las restricciones son reglas que el motor aplica de forma automática la Unique es para garantizar que todos los valores de una columna sean únicos. La restricción Check valida que el dato de la columna esté dentro de un rango.

Primary Key: es un campo o combinación de campos que identifica de forma única a cada fila de la tabla.

Objeto Índice: es una estructura de datos que mejora la velocidad de la operaciones por medio de un identificador único de cada fila de la tabla, permitiendo un rápido acceso a los registros de una BD.

Entonces => La PK es el identificador de la tabla que puede estar formado por uno o más campos y no se puede repetir, la Constraint Unique se aplica sobre una columna cualquiera para que no se repitan los datos y el índice crea ???

'Cómo implementaría la integridad referencial entre dos tablas de dos bases de datos en diferentes servidores.' 'Diferencias entre Store procedure, Función y Trigger.'

Los SPs son procedimientos almacenados en una BD, los cuales son ejecutados por un usuario o por otro proceso, y pueden realizar operaciones varias como altas, bajas y modificaciones de tablas. En cambio las Funciones no pueden hacer

modificaciones de tablas, solo hacer consultas y retornar un valor. Los trigger se ejecutan automáticamente ante eventos (insert, delete, update) en las tablas y al igual que en los SPs, puede hacer operaciones varias.

Explique las principales características que diferencian a una Base de datos Operacional de un Datawarehouse.

'Determine los componentes y las variables que intervienen en el diseño de un modelo OLAP.'

Proporciona información fiable sobre los indicadores clave del funcionamiento de una organización para los sectores que toman decisiones. A diferencia de los sistemas transaccionales, OLAP procesa información y aplica inteligencia. Un sistema OLAP se alimenta de uno o varios OLTP que proveen de datos. Todos los campos son calculados. Se usa para tomar decisiones. Se basa en patrones de interés.

OLTP → DTS → OLAP. El DTS se ejecuta periódicamente para generar información.

Se implementa con distintas tecnologías: MOLAP (bases de datos multidimensionales), ROLAP (bases de datos relacionales),

HOLAP (bases de datos híbridas).

MOLAP: ya no se trabaja con el concepto de filas y columnas sino dimensiones. Técnicas de hipercubo (hay dispersión) y multicubo (se saltan los cubos que no contienen información).

ROLAP

'Que características de funcionamiento tiene y que variantes existen del método quicksort de clasificación.'

Es un método de fácil implementación y poco consumo de recursos. Es recursivo pero se usan versiones iterativas para mejorar su rendimiento, también es in-situ, ya que usa solo una pila auxiliar. El QuickSort está basado en la idea de divide y vencerás, en el cual un problema se soluciona dividiéndolo en dos o más subproblemas, resolviendo recursivamente cada uno de ellos para luego juntar sus soluciones para obtener la solución del original.

'Indique por que un índice conformado por una clave numérica corta es más rápido en el acceso a los datos que un índice compuesto o conformado con un string largo. '

Al ocupar más espacio cada nodo del árbol supera el tamaño de una página del File System, de esta forma el Sistema Operativo debe leer varias páginas para levantar un nodo, por eso es mas lento.

'Desarrolle el concepto de trigger en cuanto a ejecución, instancias y funcionalidad.'

Ejecución: un trigger es un procedimiento que se ejecuta ante un acontecimiento (INSERT, UPDATE, DELETE) sobre una tabla determinada.

Tipos: se puede aplicar a la fila que disparó el trigger o a todas las filas.

Before, After & Instead of (en lugar del evento que lo invocó).

Atomicidad: si un error ocurre cuando un trigger se está ejecutando, la operación que disparó el trigger falla, o sea que no se modifica la tabla.

Uso: se usan triggers cuando la integridad referencial y los constraints son insuficientes; reglas de consistencia (no provistas por el modelo relacional); replicación de datos; auditoría; acciones en cascada; autorización de seguridad; los triggers constituyen la herramienta más potente para el mantenimiento de la integridad de la base de datos, ya que pueden llevar a cabo cualquier acción que sea necesaria para mantener dicha integridad; un trigger puede modificar filas de una tabla que un usuario no puede modificar directamente; pueden llamar procedimientos y disparar otros triggers, pero no pueden llevar parámetros. Ventaja: la principal ventaja es que permiten a los usuarios crear y mantener un conjunto de código más manejable para su empleo por todas las aplicaciones asociadas con las base de datos existentes y futuras.

Tablas virtuales: acceso a tablas virtuales de sólo lectura INSERTED y DELETED.

Recursividad: un trigger puede disparar una acción que a su vez, lance otro trigger y así sucesivamente.

'Desarrollar el concepto de Data Warehouse, ejemplifique'

Un Data Warehouse es una colección de datos orientada a sujetos, integrada, variante en el tiempo, no volátil, organizados para soportar necesidades empresariales. Integración de bases de datos heterogéneas (relacionales, documentales, Geográficas, archivos, etc.). Ejecución de consultas complejas no predefinidas visualizando el resultado en forma de gráfica y en diferentes niveles de agrupamiento y totalización de datos. Acceso interactivo e inmediato a información estratégica de un área de negocios. Análisis de problema en términos de dimensiones (por ejemplo el tiempo). Control de calidad de datos para asegurar la relevancia de los datos en base a los cuales se toman las decisiones.

'Desarrolle el concepto de índice asociado a una Base de Datos Relacional.'

Los índices son estructuras físicas y no lógicas. Mejoran la performance de la aplicación, aumentando la velocidad de acceso a los datos. De forma similar al índice de un libro, se guardan parejas de elementos: el elemento que se desea indexar y su posición en la base de datos. Se guardan por un lado los índices (header) ordenados y por otro los datos (cuerpo) en forma secuencial. Se pueden utilizar dos métodos: tablas de hashing y árbol-B.

Tablas de hashing: se guarda en una tabla la clave (es por lo que se ordena), la posición relativa que ocupa en la parte de datos y un puntero a una lista de duplicados. Se indexa por los campos que son accedidos con más frecuencia. Las claves pueden ser compuestas, por ejemplo nombre y legajo, entonces ordeno por nombre y a igual nombre ordeno por legajo (un solo índice). Pero si necesito ordenar por DNI y legajo entonces se necesitan dos índices.

Árbol-B: el número máximo de claves por nodo constituye el orden del árbol-B y el número mínimo de claves permitidas en un nodo es normalmente la mitad del orden excepto para la raíz. En los nodos no terminales se guardan los datos y un apuntador a otro nodo con valores menores o iguales. Todas las hojas tienen la misma profundidad (pertenecen al mismo nivel). Las hojas poseen apuntadores a los registros en el fichero de datos, de forma que el nivel completo de las hojas forma un índice ordenado del fichero de datos.

'Cual es el objetivo de un Trigger. Describa un ejemplo práctico de su utilización.'

Es mantener la integridad referencial entre tablas.

'Defina claramente el concepto de transacción, indicando su utilidad y forma de uso.'

Utilidad: Mantener la consistencia e integridad de los datos, haciendo que las instrucciones que la componen no finalicen en un estado intermedio evitando, de esta manera, posibles fallos en el sistema donde se ejecutan.

Forma de uso:

```
BEGIN TRANSACTION
    UPDATE LEGAJOS
    SET LEGA_APELLIDO = 'RIOS'
    WHERE LEGA_LEGajo = 1
COMMIT
```

'Desde el punto de vista de la Performance, en que consiste el armado de un plan de consulta?'

El plan de consulta es la forma en que el motor va a resolver una consulta enviada para su resolución, para ello existe un optimizador del propio DBMS que de varias opciones se queda con la mejor. Del mismo modo, quien ejecuta la consulta debe optimizarla previamente tratando de que por ejemplo utilice índices de acceso en forma parcial o total para que la consulta tenga mayor performance.

'Que características distintivas brinda un DBMS?'

Sistema de Administración de Bases de Datos (DBMS): Construido sobre sus capacidades en vez de las simples funciones de recuperación y almacenamiento de datos. Estos productos comúnmente incluyen grandes capacidades de backup y recuperación de datos, ingreso al sistema, control de concurrencia y bloqueo, y mecanismos de seguridad.

'A la hora de elegir un Motor de Bases de Datos determinado, que características tendría que analizar para su elección?' Costo, volumen de datos, tiempo de respuesta, control de acceso concurrente y seguridad, redundancia y recuperación de datos.

'Desde el punto de vista de la Performance, en que mejora el armado de un Modelo OLAP.'

OLAP: En el análisis multidimensional, los datos se presentan mediante dimensiones como producto, territorio y cliente. Por lo regular las dimensiones se relacionan en jerarquías, por ejemplo, ciudad, estado, región, país y continente, o estado, territorio y región.

Al procesamiento analítico o análisis multidimensional se le conoce también como procesamiento analítico en línea (OLAP).

Se procesa en una visión multidimensional de los datos empresariales en el Data Warehouse y puede tener un motor de depósito de base de datos multidimensional. De esta forma, dentro de un Data Warehouse existen dos tecnologías complementarias, una relacional para consultas y una multidimensional para análisis. Este concepto nace ante la debilidad fundamental de las bases de datos relacionales para proveer capacidad de análisis y sistemas de soporte a las decisiones. La siguiente definición muestra las mejoras que provee un Modelo Olap:

Fast (Rápido), significa que el sistema está orientado a tener tiempos de respuesta de 5 segundos, con respuestas de 1 segundo a consultas sencillas y, algunas más complejas, con respuestas de hasta 30 segundos. En situaciones típicas, si las respuestas toman más tiempo, el usuario perderá la concentración en lo que busca analizar.

Analysis (Análisis), lo que indica que al usuario debe proporcionársele suficiente funcionalidad para resolver sus problemas, sin la necesidad de contar con el apoyo de sistemas o de una pre-programación. De esta forma, se podrán realizar consultas no definidas, cálculos de diferencias, variaciones y tendencias, consolidar y llevar a cabo análisis de sensibilidad y de búsqueda de metas.

Shared (Compartida), significa que debe haber facilidad de acceso simultáneo, tanto de lectura como de escritura, aunado a un esquema de seguridad adecuado, con el objeto de guardar la confidencialidad (probablemente a nivel de celda).

Multidimensional, ha sido y continuará siendo un requisito, con la habilidad de manejar múltiples jerarquías y dimensiones.

Information (Información), son todos los datos y la información derivada, cuando y donde sea necesaria, en su contexto, tanto de información "suave" como información "dura", interna o externa.

'Que se entiende por Multicubo y Hypercubo y cual es la utilización y diferencias entre ambos.'

Es una forma de almacenamiento en forma de array de los datos de las bases Multidimensionales.

'Cual es el concepto de Data Marts? Para que se utilizan?. Cual es la diferencia entre un Data Marts y un Datawarehouse?'

Data Marts es el mismo concepto que Datawhereshousing pero se utiliza en departamentos, son pequeños proyectos de Data wherehousing. La principal diferencia es el tamaño, dado que los Datawhereshousing tienen grandes cantidades de información almacenada de toda la empresa, por otro lado el data mart solo tiene información de un departamento, esto no quiere decir que sea poca información, pero si lo es en relación, por otro lado, al ser mas pequeño, tiene diferencias, de costo, implementación, accesos, etc.

'Qué se entiende por DTS?'

Son herramientas y objetos que permiten mover datos entre varios orígenes de datos diferentes o iguales. Un DTS puede realizar las siguientes tareas:

- Importación y exportación de datos entre dos orígenes de datos.
- Transformación de datos.
- Copia de objetos de base de datos : tablas, índices, etc.
- Ejecución de instrucciones de Transact-SQL.

DTS tiene una arquitectura OLE DB por lo que puede copiar y transformar múltiples orígenes de datos, entre ellos: SQL Server, Oracle.

'En que consiste la Técnica de Data Mining? Para que se utiliza e indique alguna de las técnicas algorítmicas aplicadas en el mismo.'

Las computadoras son cargadas con mucha información acerca de una variedad de situaciones donde una respuesta es conocida y luego el software de Data Mining en la computadora debe correr a través de los datos y distinguir las características de los datos que llevarán al modelo. Una vez que el modelo se construyó, puede ser usado en situaciones similares donde no se conoce la respuesta.

'Indique las características y funcionalidad de una DataWarehouse.'

Un Data Warehouse es una colección de datos orientada a sujetos, integrada, variante en el tiempo, no volátil, organizados para soportar necesidades empresariales. Integración de bases de datos heterogéneas (relacionales, documentales, Geográficas, archivos, etc.). Ejecución de consultas complejas no predefinidas visualizando el resultado en forma de gráfica y en diferentes niveles de agrupamiento y totalización de datos. Acceso interactivo e inmediato a información estratégica de un área de negocios. Análisis de problema en términos de dimensiones (por ejemplo el tiempo). Control de calidad de datos para asegurar la relevancia de los datos en base a los cuales se toman las decisiones.

'¿Por qué se establece que las Reglas del Negocio deben estar en el Motor de Base de Datos y no en la aplicación cliente?'

Reglas de negocio: cada aplicación debe reflejar las restricciones que existen en el negocio dado, de modo que nunca sea posible realizar acciones no válidas. La información puede ser manipulada por muchos programas distintos que podrán variar de acuerdo a los departamentos de la organización, los cuales tendrán distintas visiones y necesidades pero que en cualquier caso, siempre deberán respetar las reglas de negocio. Es por lo anterior expuesto que las reglas del negocio deben estar en el motor de base de datos.

'¿En qué consisten, para qué se utilizan y cómo pueden implementarse las tecnologías OLAP? Ejemplifique.'

Proporciona información fiable sobre los indicadores clave del funcionamiento de una organización para los sectores que toman decisiones. A diferencia de los sistemas transaccionales, OLAP procesa información y aplica inteligencia. Un sistema OLAP se alimenta de uno o varios OLTP que lo proveen de datos. Todos los campos son calculados. Se usa para tomar decisiones. Se basa en patrones de interés.

OLTP → DTS → OLAP. El DTS se ejecuta periódicamente para generar información.

Se implementa con distintas tecnologías: MOLAP (bases de datos multidimensionales), ROLAP (bases de datos relacionales),

HOLAP (bases de datos híbridas).

MOLAP: ya no se trabaja con el concepto de filas y columnas sino dimensiones. Técnicas de hipercubo (hay dispersión) y multicubo (se saltan los cubos que no contienen información).

'Explique el Modelo Star para el armado de una DataWarehouse. Ejemplifique.'

Es un esquema relacional orientado a la representación de dimensiones que convergen sobre un hecho común o un valor cuantificable. De esta forma se logra implementar en un sistema de base de datos tradicional el concepto de cubos multidimensionales. Existe una tabla central denominada "tabla de hechos" que representa el atributo cuantificable. Alrededor de esta tabla central se definen otras tablas ligadas a esta mediante claves foráneas, y se denominan tablas de dimensiones y son aquellas que definen cualidades del atributo representado por la tabla de hechos. La clave primaria de la tabla de hechos será una clave compuesta por todas las claves foráneas que representan las claves primarias de cada una de las dimensiones asociadas.

'Enumere y explique al menos 3 algoritmos de clasificación que conozca.'

Dicotómico o burbuja: funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada.

Quicksort: elegir un elemento de la lista a ordenar (pivote). Resituar los demás elementos de la lista a cada lado del pivote, de un lado los menores y del otro los mayores (en este momento el pivote está ordenado). Se repite el proceso de forma recursiva para las dos sublistas mientras éstas contengan más de un elemento. Una vez terminado el proceso todos los elementos estarán ordenados. Eficiencia promedio $n \log(n)$; en el peor de los casos n^2 .

Heapsort: el 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) de todos los elementos almacenados en él.

'Enumere ventajas, desventajas y diferencias entre una Base de Datos Orientada a Objetos y una Base de Datos Relacional.'

Ventajas BDOO:

Simplifica la programación.

Flexibilidad: uso de sub-clases que permiten la herencia de atributos, esto permite que una BDOO puede ajustarse a usar siempre el espacio de los campos que son necesarios, eliminando el espacio desperdiciado en registros con campos que no se usan. Manejo de datos complejos: manipula datos complejos de forma rápida y ágil. La estructura de la BDOO está dada por referencias (apuntadores lógicos) entre objetos.

Desventajas BDOO:

Baja performance. Inmadurez del mercado de BDOO. Falta de estándares en la industria orientada a objetos.

¿En qué consiste la técnica de Data Mining? ¿Para qué se utiliza? Indique alguna de las técnicas algorítmicas aplicadas en el mismo.'

Consiste en la extracción de información oculta y predecible de grandes bases de datos. Las herramientas de datamining predicen futuras tendencias y comportamientos, permitiendo tomar decisiones conducidas por un conocimiento acabado de la información (knowledge-driven). Predicción automatizada de tendencias y comportamientos. Descubrimientos automatizados de modelos previamente desconocidos.

Las técnicas más comúnmente usadas son: redes neuronales, árboles de decisión, algoritmos genéticos, etc.

'Desarrollar las ventajas y desventajas de crear mayor o menor cantidad de índices sobre las tablas de un modelo físico.'

Si tuviera que elegir un método de creación de índices, entre Hashing y Árbol B, cuando usaría cada uno de ellos y por qué? Indique las diferentes funcionalidades y características de un DBMS.

Desarrollo el concepto de FUNCION en PL-SQL. Indique ventajas y desventajas sobre otros objetos similares en un DBMS.

Que se entiende por DTS. Indique el concepto y su utilización.

Defina claramente el concepto de transacción, indicando su utilidad y forma de uso.

Cuáles son las diferencias entre una PK y una constraint de Unique ?

Explicar el modelo estrella de un datawarehouse.

Indique las características mandatorias y opcionales de un Motor de Base de Datos Relacional.

¿Qué diferencia conceptual existe entre los índices de archivos de datos tradicionales y los índices creados en una Base de Datos Relacional?

¿Desde el punto de vista de la performance, en qué mejora el armado de un modelo OLAP?

A la hora de elegir un Motor de Base de Datos determinado, ¿qué características tendría en que analizar para su elección?

'¿Qué entiende por diccionario de datos, catálogo de datos o metadata?'

Para representar una base de datos relacional, necesitamos almacenar no sólo las relaciones mismas, sino también una cantidad de información adicional acerca de las relaciones. El conjunto de tal información se llama diccionario de datos o metainformación. El diccionario de datos debe contener información como: nombre de las relaciones en la base de datos; nombre de los atributos de cada relación; dominio de cada atributo; tamaño de cada relación; método de almacenamiento de cada relación; claves y otras restricciones de seguridad; nombres y definiciones de vistas.

Además debe contener información sobre los usuarios de la base de datos y los poderes que éstos tienen: nombre de los usuarios; costo del uso efectuado por cada usuario; nivel de privilegio de cada usuario.

Es de notar que el diccionario de datos no es más que una base datos acerca de la base de datos, y puede almacenarse y manejarse con los mismos métodos.

'¿Qué beneficios brinda la aplicación de la normalización al diseño de un modelo de base de datos?'

Elimina la redundancia de datos. Mejora la performance.

'Defina los conceptos de Data Warehouse y Datamart. Especialmente similitudes y diferencias.'

DataWarehouse: YA ESTÁ EXPLICADO.

DataMart: podemos entender un Data Mart como un subconjunto de los datos del Data Warehouse con el objetivo de responder a un determinado análisis, función o necesidad y con una población de usuarios específica. La diferencia entre un DataWarehouse y un DataMart es que el data mart está pensado para cubrir las necesidades de un grupo de trabajo o de un determinado departamento dentro de la organización. Es el almacén natural para los datos departamentales. En cambio, el ámbito del data warehouse es la organización en su conjunto. Al igual que en un data warehouse, los datos están estructurados en modelos de estrella o copo de nieve y un data mart puede ser dependiente o independiente de un data warehouse. Los orígenes de datos son homogéneos a diferencia del DW.

'Describe y cite al menos un ejemplo de una representación computacional que permite representar computacionalmente un grafo irrestricto.'

De manera estática: con una matriz de adyacencia $n \times n$ (n : cantidad de nodos del grafo), en donde un 1 en una posición ij representa la existencia de una flecha del nodo i hacia el nodo j . Presenta el problema de ocupar mucho espacio para representar lo que realmente existe ya que se reserva espacio para lugares que no tienen datos.

De manera dinámica: utilizando una lista o vector de punteros donde se guardan los nodos que participan del grafo con sus correspondientes datos. Cada elemento contiene un puntero a una lista adicional que contiene "las flechas que salen" del nodo que representan. No se guarda el dato en la lista adicional, guarda los punteros a datos.

'¿Cuál es la diferencia entre la utilización de Multicubos e Hypercubos para la implementación de bases de datos multidimensionales?'

Hypercubos: Se guardan todas las dimensiones en un cubo. Cada intersección del cubo es otro cubo y así sucesivamente. Se pueden agregar dimensiones pero es estático porque no se puede variar la dimensión del cubo. Hay dispersión de datos porque los datos no están compactos por el hecho de que puede haber muchos datos vacíos.

Multicubos: Se genera una lista con los cubos que tienen información, es decir se saltea los cubos vacíos.

'Desde el punto de vista de la performance, ¿en qué consiste el armado de un plan de consulta?'

El optimizador de consultas es el componente del sistema de gestión de base de datos que intenta determinar la forma más eficiente de ejecutar una consulta SQL. Se decide cuál de los posibles planes de ejecución de una consulta dada es el más eficiente. Los optimizadores basados en costos asignan un costo (operaciones de E/S requeridas, CPU, etc.) a cada uno de esos planes y se elige el que tiene menor costo.

El optimizador no puede ser accedido directamente por los usuarios, sino que, una vez enviadas las consultas al servidor, pasan primero por el analizador y recién entonces llegan al optimizador. Una consideración muy importante es el orden de los "join".

'¿Qué características distintivas brinda un DBMS? NO SÉ A QUÉ SE REFIERE'

Describe los componentes básicos de un DBMS. Teniendo en cuenta la administración de datos, la interfaz con el usuario y el procesamiento cliente-servidor

Dos lenguajes: DML y DDL.

3 capas: externa (usuario), lógica (conceptual) e interna (física).

Disk manager (recibe peticiones del file manager y las envía al SO), file manager (decide que página contiene el registro deseado y la solicita al disk manager) y user manager (permite la interacción con el usuario).

Arquitectura cliente-servidor: las aplicaciones corren en el cliente y generan solicitudes para y reciben respuestas del servidor. El servidor realiza el procesamiento de datos y aplica las reglas de negocio establecidas.

Clustering: es la forma de agrupar la información. Hay dos formas, intra-file clustering (todas las tablas conviven con sus elementos) e inter-file clustering (se guarda cada objeto con sus relaciones).

'A la hora de elegir un Motor de Bases de Datos determinado, que características tendría que analizar para su elección?'

En que consiste un Plan de Consulta, cual es su utilización y la ventaja de su implementación.

Indique concepto, tipos, funcionalidades y cual es la mayor ventaja de los triggers.

Que se entiende por Multicubo y Hypercubo y cual es la utilización y diferencias entre ambos.

Cual es el concepto de Data Marts? Para que se utilizan?. Cual es la diferencia entre un Data Marts y un Datawarehouse? Los data marts se ajustan a las necesidades que tiene una parte específica de un negocio, más que a las de toda una empresa. Optimizan la distribución de información útil para la toma de decisiones y se enfocan al manejo de datos resumidos o de muestras. No necesitan ser administrados centralmente por el departamento de sistemas de una organización, sino que pueden estar a cargo de un grupo específico dentro del área de la empresa que los utilice. En ocasiones, los proyectos que comienzan como data warehouses evolucionan a data marts. Cuando las organizaciones acumulan grandes cantidades de datos históricos para el apoyo de decisiones, que rara vez o nunca usan, pueden reducir la información guardada y convertir su data warehouse en un data mart mejor enfocado. Los sistemas que extraen y almacenan datos de diversas fuentes para la toma de decisiones, se denominaban Data Warehouses. En fechas recientes se ha hecho una distinción entre los grandes sistemas para almacenar datos (data warehouses) y los sistemas más pequeños (data marts). En que consisten, para que se utilizan y como pueden implementarse las Tecnologías OLAP. Ejemplifique. Responde con rapidez a las consultas, de modo que el proceso de análisis no se interrumpa y la información no se desactualiza.

_ Tiene un motor de depósito de datos multidimensional, que almacena los datos en arreglos. Esos arreglos son una representación lógica de las

dimensiones empresariales. La tecnología OLAP se aplica en muchas áreas funcionales de una empresa, tales como producción, ventas y análisis de rentabilidad de la comercialización, mezcla de manufacturas y análisis de logística; consolidaciones financieras, presupuestos y pronósticos, planeación de impuestos y contabilidad de costos. OLAP surge como un proceso para ser usado en el análisis de ventas y mercadotecnia, para elaborar reportes administrativos y consolidaciones, para presupuestación y planeación, para análisis de rentabilidad, reportes de calidad y otras aplicaciones que requieren una visión flexible, de arriba a abajo, del negocio. OLAP provee de reportes sumarios que los ejecutivos requieren para tomar decisiones, así como la facilidad de elaborar cálculos complejos, enfoques a detalles operativos y consultas no programadas. OLAP se alimenta principalmente de los sistemas transaccionales y como tales, debe considerar una eficiente administración de la base de datos y proveer un nivel adecuado de seguridad.

1. Si tuviera que elegir un método de creación de índices, entre Hashing y Árbol B, cuando usaría cada uno de ellos y por qué? Que parámetros tendría que tener en cuenta para decidirme por algún método

en especial? Tengo entendido que el volumen de los datos influye en la elección de los métodos pero me gustaría saber la respuesta correcta a esta pregunta. 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

'FUNCIONALIDADES Y CARACTERÍSTICAS DE UN DBMS:'

- Concurrencia: permite el acceso simultáneo de muchos usuarios/procesos.
- Atomicidad: asegura que un conjunto de operaciones se ejecuta todo o falla en su conjunto.
- Consistencia: asegura que la integridad de los datos se mantiene antes y después de realizar una operación.
- Backup y Restore: provee herramientas para realizarlos.
- Gestión de almacenamiento: independiza al usuario de los detalles de bajo nivel de cómo se almacenan los datos. (tratamiento con FS)
- Aislamiento: usuario se abstrae de la implementación física del modelo.
- Facilidad de consultas.
- Transaccionalidad.- Abstracción.
- Procesamiento distribuido: las diferentes capas o sistemas de un DBMS interactúan entre sí para llevar a cabo los objetivos generales de un DBMS.

'QUE DEBE CUMPLIR UN MOTOR DE BD'

- Relación costo/beneficio.- Volúmenes de datos a manejar.- Tipos de bloqueos.- Concurrencia
- Si el sistema es o no orientado a objetos.- Capacidad del equipo de trabajo.- Plataforma (Hardware y SO)

'Características de los DSS Queries (DataWarehouse) > OLAP:'

- Muchas filas son leídas y el resultado no está en una transacción- Los datos son leídos secuencialmente
- SQL complejos son ejecutados- Grandes archivos temporarios son creados
- El tiempo de respuesta es medido en horas y minutos- Hay relativamente poca concurrencia de queries

'CARACTERÍSTICAS DEL Ambiente OLTP: OnLine Transaction Processing'

- Relativamente pocas filas leídas- Alta frecuencia de transacciones- Acceso a los datos a través de índices
- Simple Operaciones SQL- Respuesta de scan medida en segundos- Muchos queries concurrentes

'Existen 3 niveles de Seguridad:' a) Base de datos b) Tabla c) Columna (depende del motor de bd)

'PLAN DE CONSULTA:' Consiste en que el SGBD tiene armado un plan de cómo va a acceder a los datos, esto hace que el acceso sea más rápido. Los SP, Vistas y Funciones implementan un plan de ejecución, es por esto que son más veloces que un query normal. El optimizador de consultas es el componente del sistema de gestión de base de datos que intenta determinar la forma más eficiente de ejecutar una consulta SQL. Se decide cuál de los posibles planes de ejecución de una consulta dada es el más eficiente. Los optimizadores basados en costos asignan un costo (operaciones de E/S requeridas, CPU, etc.) a cada uno de esos planes y se elige el que tiene menor costo.

'Por qué se establece que las Reglas del Negocio deben estar en el Motor de Base de Datos y no en la aplicación cliente?'

- Reglas de negocio: cada aplicación debe reflejar las restricciones que existen en el negocio dado, de modo que nunca sea posible realizar acciones no válidas.
- La información puede ser manipulada por muchos programas distintos que podrán variar de

acuerdo a los departamentos de la organización, los cuales tendrán distintas visiones y necesidades pero que en cualquier caso, siempre deberán respetar las reglas de negocio.

- Es por lo anterior expuesto que las reglas del negocio deben estar en el motor de base de datos.

'Describa los componentes básicos de un DBMS. Teniendo en cuenta la administración de datos, la interfaz con el usuario y el procesamiento clienteservidor.'

- Dos lenguajes: DML y DDL.

- 3 capas: externa (usuario), lógica (conceptual) e interna (física).

- Disk manager (recibe peticiones del file manager y las envía al SO), file manager (decide que página contiene el registro deseado y la solicita al disk manager) y user manager (permite la interacción con el usuario).

- Arquitectura clienteservidor: las aplicaciones corren en el cliente y generan solicitudes para y reciben respuestas del servidor. El servidor realiza el procesamiento de datos y aplica las reglas de negocio establecidas.

- Clustering: es la forma de agrupar la información. Hay dos formas, intrafile clustering (todas las tablas conviven con sus elementos) e interfile clustering (se guarda cada objeto con sus relaciones).

'¿Qué beneficios brinda la aplicación de la normalización al diseño de un modelo de base de datos?'

- Evitar la redundancia de los datos.

- Evitar problemas de actualización de los datos en las tablas.

- Proteger la integridad de los datos.

'Cuál es el objetivo del algoritmo de Dijkstra?'

Es un algoritmo cuyo objetivo es determinar la ruta más corta, desde el nodo origen, hasta cualquier nodo de la red. Su metodología está basada en iteraciones,

con lo cual su complejidad aumenta a medida que aumenta el tamaño de la red.

Para esto, hace uso y define etiquetas a partir del nodo origen y para cada uno de los nodos subsiguientes. Estas etiquetas contienen información relacionada

con un valor acumulado del tamaño de los arcos y con la procedencia más próxima de la ruta.

'Desarrolle el concepto de Datawarehouse, en lo que respecta a utilidad, técnicas de armado, convivencia con modelos transaccionales y acceso a la información.'

Un Data Warehouse es un almacén de información orientada a sujetos, integrada, variante en el tiempo, no volátil, organizada y orientada para la toma de decisiones.

La información proviene de la integración de múltiples fuentes (bbdd relacionales, bbdd documentales, archivos, etc.).

La idea de un data Warehouse es, mediante el data mining, explotar la información contenida y poder sacar buena información para la toma de decisiones.

Los diferentes modelos de construcción son MOLAP (multidimensional), ROLAP (relacional) y HOLAP (híbrido).

Existen los data marts, que permiten contar con vistas multidimensionales de cada área que componen al data warehouse.

Tanto el DW como los data marts pueden ser implementados en un RDBMS utilizando el modelo estrella, en el cual la información está conformada por dos tipos de

tablas: los hechos (definen métricas de un evento específico) y las dimensiones (definen características).

'¿Qué entiende por diccionario de datos, catálogo de datos o metadata?'

✓ Para representar una base de datos relacional, necesitamos almacenar no sólo las relaciones mismas, sino también una cantidad de información adicional acerca de las relaciones.

- El conjunto de tal información se llama diccionario de datos o metainformación.

- El diccionario de datos debe contener información como: nombre de las relaciones en la base de datos; nombre de los atributos de cada relación; dominio de cada atributo; tamaño

de cada relación; método de almacenamiento de cada relación; claves y otras restricciones de seguridad; nombres y definiciones de vistas.- Además debe contener información sobre los usuarios de la base de datos y los poderes que éstos tienen: nombre de los usuarios; costo del uso efectuado por cada usuario; nivel de

privilegio de cada usuario.

Especifique y detalle 1 escenario donde sería adecuado implementar una vista materializada:.

Un escenario adecuado para implementar una vista materializada es cuando se necesita mejorar el rendimiento de consultas complejas y costosas en una base de datos.

Imaginemos un sistema de comercio electrónico que maneja una gran cantidad de datos de productos, transacciones y clientes. Uno de los requisitos frecuentes es generar informes de ventas mensuales que involucren consultas complejas y cálculos agregados sobre estos datos.

En este escenario, crear una vista materializada puede ser beneficioso. Una vista materializada es una copia precalculada y almacenada de los resultados de una consulta compleja. Aquí hay un ejemplo de cómo se podría utilizar una vista materializada en este caso:

Consulta compleja: La consulta original para generar un informe de ventas mensuales podría requerir combinaciones de varias tablas, cálculos de sumas y agrupaciones, y posiblemente operaciones de fecha y hora. Esta consulta puede ser lenta y costosa en términos de tiempo de respuesta debido al gran volumen de datos y la complejidad de los cálculos.

Creación de la vista materializada: En lugar de ejecutar la consulta compleja cada vez que se solicita el informe, se puede crear una vista materializada que almacene los resultados precalculados. Esta vista materializada se actualizaría periódicamente (por ejemplo, cada noche) o en función de ciertos eventos (como la finalización de una transacción importante).

Consulta del informe: Cuando se solicita el informe de ventas mensuales, en lugar de ejecutar la consulta compleja original, se accede directamente a la vista materializada. Dado que los resultados ya están precalculados y almacenados, la consulta se resuelve mucho más rápido y eficientemente. Esto mejora significativamente el rendimiento de la consulta y reduce la carga en el sistema.

Es importante tener en cuenta que, si bien una vista materializada mejora el rendimiento de consultas frecuentes, también implica un mayor costo de mantenimiento, ya que la vista debe actualizarse para reflejar los cambios en los datos subyacentes. Sin embargo, en escenarios donde las consultas complejas son recurrentes y los datos subyacentes no cambian con tanta frecuencia, el beneficio de mejora en el rendimiento puede superar los costos de mantenimiento asociados con la vista materializada.