

**Nombre del alumno: Johanne Joaquín
Arriaga Díaz**

**Nombre del profesor: Cesar Alfredo
Escobar Sánchez.**

**Nombre del trabajo: Súper nota de
unidad IV.**

PASIÓN POR EDUCAR

Materia: Bases de datos II.

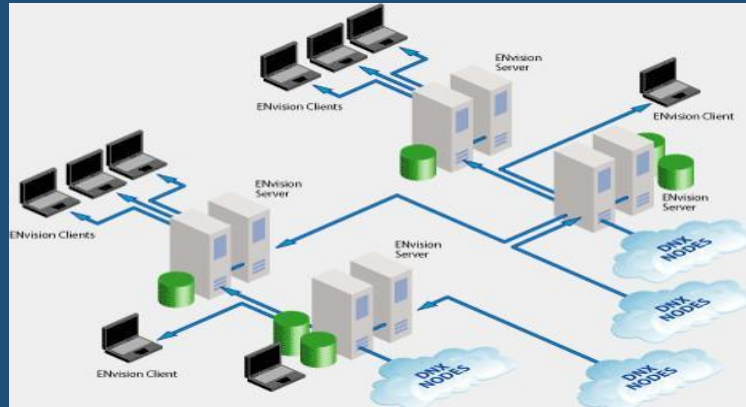
Grado: Octavo cuatrimestre

Grupo: ISC13SDC0119-F

¿Qué son bases de datos distribuidas?

Son un grupo de datos que pertenecen a un sistema pero a su vez esta repartido entre ordenadores de una misma red, a nivel local o cada uno en una diferente localización geográfica, cada sitio en la red es autónomo en sus capacidades de procesamiento y es capaz de realizar operaciones locales y en cada uno de estos ordenadores debe estar ejecutándose una aplicación a nivel global que permita la consulta de todos los datos como si se tratase de uno solo.

-Para tener una base de datos distribuida debe cumplirse las condiciones de una Red Computacional. Una red de comunicación provee las capacidades para que un proceso ejecutándose en un sitio de la red envíe y reciba mensajes de otro proceso ejecutándose en un sitio distinto. Parámetros a considerar incluyen: Retraso en la entrega de mensajes, Costo de transmisión de un mensaje y Confiabilidades de la red.



Arquitectura de Base de Datos

Global Schema: Define los datos que están incluidos en la bd distribuida tal como si la bd no fuera distribuida. Una definición de relaciones globales.

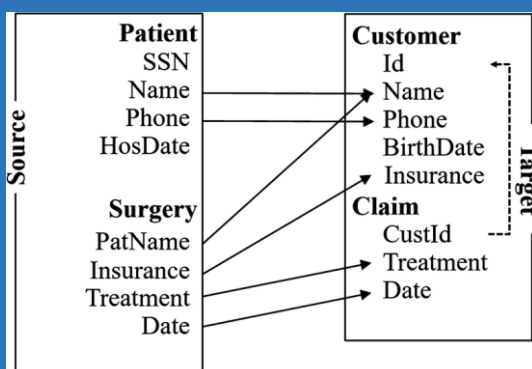
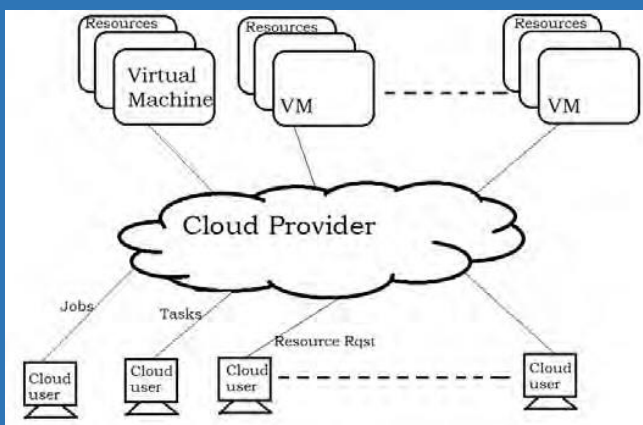
-Fragmentation Schema: Traducción entre relaciones globales y fragmentos. (Una relación global puede consistir de varios fragmentos pero un fragmento está asociado con sólo una relación global)

-Allocation Schema: Define el sitio (o sitios) en el cual un fragmento está localizado. **-Local**

Mapping Schema: Traduce los fragmentos locales a los objetos que son manejados por el

SMBD local Separación entre fragmentación y localización. -Transparencia de Fragmentación

-Transparencia de Localización -Control explícito de redundancia -Independencia de BD locales



Tipos de almacenamiento

1-Replica: El sistema conserva copias o réplicas idénticas de una tabla. Cada réplica se almacena en un nodo diferente. **Ventajas:** Disponibilidad: El sistema sigue funcionando aún con caída de uno de los nodos. Aumento del paralelismo: Varios nodos pueden realizar consultas en paralelo sobre la misma tabla. **Inconveniente:** Aumento de la sobrecarga en las actualizaciones: El sistema debe asegurar que todas las réplicas de la tabla sean consistentes.

2.-Fragmentación: Existen tres tipos de fragmentación la horizontal, la vertical y la mixta.

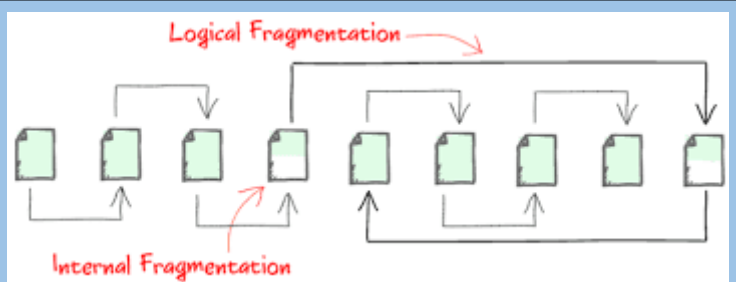
4.-Niveles de Transparencia en una Base de Datos Distribuida: Ofrece transparencia para el manejo de la información. La transparencia adecuada permite ocultar detalles de implementación a capas de alto nivel de un sistema y a otros usuarios. El sistema de bases de datos distribuido permite proporcionar independencia de los datos. La independencia de datos se puede dar en dos aspectos: lógica y física.

1 Lógica: inmunidad de las aplicaciones de usuario a los cambios en la estructura de la base de datos.

2 Física: Ocultamiento de detalles sobre estructuras de almacenamiento a las aplicaciones de usuario.

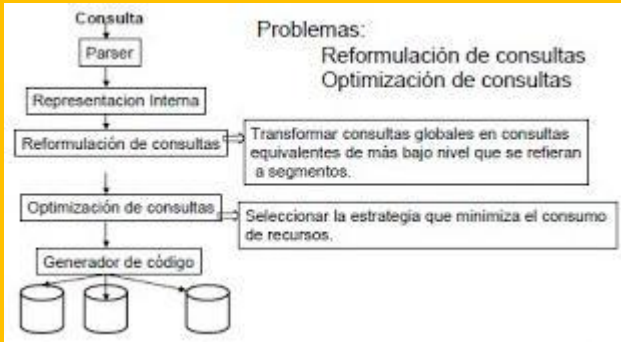
Transparencia sobre la localización de datos: el comando que se usa es independiente de la ubicación de los datos en la red y del lugar en donde la operación se lleve a cabo.

Transparencia sobre el esquema de nombramiento: Lo anterior se logra proporcionando un nombre único a cada objeto en el sistema distribuido. Así, no se debe mezclar la información de la localización con en el nombre de un objeto.



Metodología de procesamiento de consultas distribuidas

Primeramente se debe de contar con heterogeneidad de los datos, para que puedan ser usados para formular consultas. Tenemos los siguientes ejemplos:



- Así como también necesitamos contar con:
- Localización de los datos para generar reglas heurísticas
 - Descomposición de consultas en paralelo en cada nodo
 - Reducir la cantidad de datos a transferir en la red

OPTIMIZACION DE CONSULTAS DISTRIBUIDAS

Para poder optimizar una consulta necesitamos tener claras las propiedades del algebra relacional para asegurar la reformulación de la consulta, al optimizar una consulta obtenemos los siguientes beneficios:

- Minimizar costos
- Reducir espacios de comunicaciones
- Seguridad en envíos de información

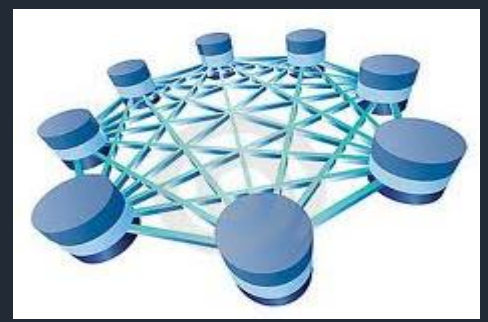
Algoritmos de optimización de consultas en SGBDD: reducir la cantidad de datos a transferir

An infographic titled 'Optimización de consultas distribuidas' (Distributed Query Optimization) for 'Bases de Datos Distribuidas' (Distributed Databases). It features a central globe with various icons representing data and network elements. Surrounding the globe are several callout boxes with text and icons, including 'Introducción', 'Algoritmos de optimización', and 'Estrategias de optimización'. The main title is in red and blue text.



Control de concurrencia

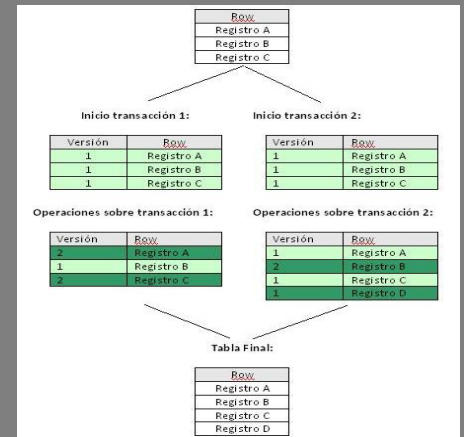
Trata con los problemas de aislamiento y consistencia del procesamiento de transacciones. El control de concurrencia distribuido de una DDBMS asegura que la consistencia de la base de datos se mantiene en un ambiente distribuido multiusuario. Si las transacciones son internamente consistentes, la manera más simple de lograr este objetivo es ejecutar cada transacción sola, una después de otra.



Algoritmos de control de concurrencia

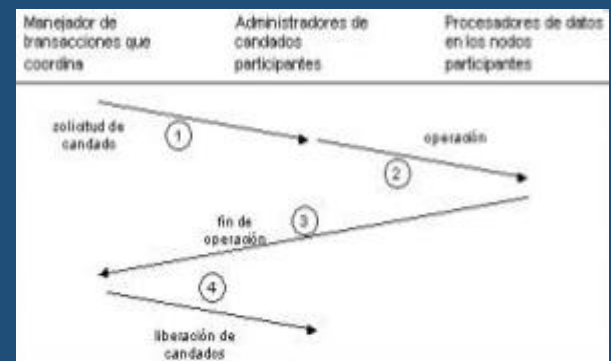
El criterio de clasificación común de los algoritmos de control de concurrencia es el tipo de primitiva de sincronización. Esto resulta en dos clases:

- Algoritmos que basados en acceso mutuamente exclusivo a datos compartidos (candados o bloqueos).
- Aquellos que intentan ordenar la ejecución de las transacciones de acuerdo a un conjunto de reglas (protocolos).



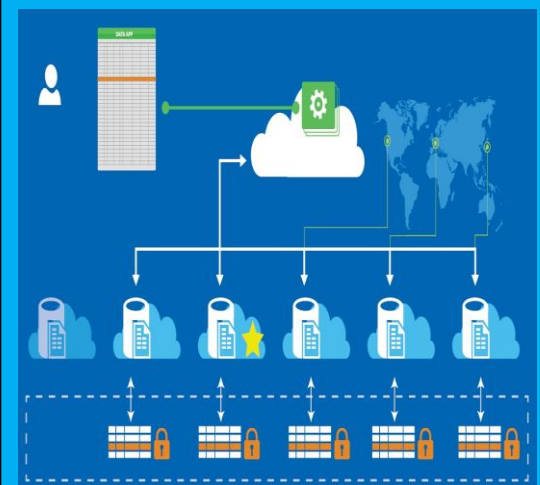
Algoritmo basado en estampas de tiempo

No pretenden mantener la seriability por exclusión mutua. En su lugar, seleccionan un orden a prioridad y ejecutan las transacciones, de acuerdo a ellas. Para establecer este ordenamiento, el administrador de transacciones le asigna a cada transacción T1 una estampa (Un identificador para cada transacción de manera única) de tiempo t_1 (T1) cuando ésta inicia. Cada transacción lleva asociada una marca de tiempo. Cada dato lleva asociado dos marcas de tiempo: uno de lectura y otro de escritura, que reflejan la marca de tiempo de la transacción que hizo la última operación de ese tipo sobre el dato. Para leer la marca de escritura, debe ser menor que el de la transacción y para escribir las marcas de escritura y lectura del dato, deben ser menores que el de la transacción, sino se aborta. Esta técnica es libre de Ínterbloqueos pero puede que haya que repetir varias veces la transacción. En sistemas distribuidos se puede usar un mecanismo como, los relojes de Lamport para asignar marcas de tiempo. El conjunto de algoritmos pesimistas está formado por algoritmos basados en candados, basados en ordenamiento por estampas de tiempo e híbridos. Los algoritmos optimistas se componen por los algoritmos basados en candados y basados en estampas de tiempo.



Algoritmos de cerradura o basados en candados

En los algoritmos basados en candados, las transacciones indican sus intenciones solicitando candados al despachador (llamado el administrador de candados) Los candados son de lectura, llamados compartidos, o escritura, llamados exclusivos. El administrador de transacciones pasa al administrador de candados la operación sobre la base de datos (lectura o escritura) e información asociada, como el elemento de datos accesado y el identificador de la transacción que envía la operación a la base de datos. El administrador de candados verifica si el elemento de datos que se quiere acceder ya ha sido bloqueado por un candado. Si el candado es incompatible con el que esta bloqueado el dato, entonces, la transacción solicitante es retrasada. De otra forma, el candado se define sobre el dato en el modo deseado y la operación a la base de datos es transferida al procesador de datos. El administrador de transacciones es informado luego sobre el resultado de la operación. La terminación de una transacción libera todos los candados y se puede iniciar otra transacción que estaba esperando el acceso al mismo dato. Se usan cerraduras o candados de lectura o escritura sobre los datos. Para asegurar secuencialidad se usa un protocolo de dos fases, fase de crecimiento: se establecen los cerrojos y fase de decrecimiento: se liberan los cerrojos. Pero se pueden producir ínterbloqueos. En los sistemas distribuidos el nodo que mantiene un dato se encarga normalmente de gestionar los cerrojos sobre el mismo.



Candados de dos fases:

Una transacción le pone un candado a un objeto antes de usarlo.

Cuando un objeto es bloqueado con un candado por otra transacción, la transacción solicitante debe esperar. Cuando una transacción libera un candado, no puede solicitar más candados. En la primera fase solicita y adquiere los candados sobre los elementos que va a utilizar y en la segunda fase libera los candados obtenidos uno por uno.

Aborto en cascada: Cuando una transacción aborta después de liberar un candado, y provoca que otras transacciones que hayan accedido el mismo elemento de datos aborten también. Para evitarlo, los despachadores de dos fases implementan candados estrictos de dos fases en los cuales se liberan todos los candados juntos cuando la transacción termina (con compromiso o aborta)



Candados de dos fases centralizados:

En sistemas distribuidos puede que la administración de los candados se dedique a un solo nodo del sistema, por lo tanto, se tiene un despachador central el cual recibe todas las solicitudes de candados del sistema. La comunicación se presenta entre el administrador de transacciones del nodo en donde se origina la transacción, el administrador de candados en el nodo central y los procesadores de datos de todos los nodos participantes. Los nodos participantes son todos aquellos en donde la operación se va a llevar a cabo.

