**BASES DE DATOS DISTRIBUIDAS**

Un sistema de bases de datos distribuida consiste en un conjunto de bases de datos lógicamente relacionadas y distribuidas sobre una red de computadoras interconectadas que se encuentran en varios lugares geográficos distintos y se administran de forma local y global. Por ejemplo, un sistema bancario compuesto por cuatro sucursales situadas en cuatro ciudades diferentes, cada sucursal posee su propia computadora con una base de datos que alberga todas las cuentas abiertas en dicha sucursal. Así, cada una de estas instalaciones se considera un sitio. También hay un sitio único, que mantiene la información relativa a todas las sucursales del banco.

Hay varios **factores que han hecho que las Bases de Datos evolucionan Bases de Datos Distribuidas**:

* En el mundo de los negocios se ha dado el fenómeno de la globalización
* A la vez los negocios de las empresas son cada vez más descentralizados geográficamente.

**COMPONENTES DE UNA BASE DE DATOS DISTRIBUIDA:**

1. **LDBMS:** Sistema Gestor local de datos. Funciona como un sistema estándar de administración de la base de datos, responsable de controlar los datos locales en cada sitio que tenga la base de datos. Cuenta con su propio diccionario de datos locales, así como los subsistemas usuales para el control de concurrencia recuperación y optimización de consumo el
2. **DDBMS**: Sistema Gestor Distribuido de la Base de Datos. Implica un conjunto de programas que operan en diversas computadoras y puede estar distribuido en los servidores locales. Estos programas pueden ser subsistemas de un único gestor de un mismo fabricante o podría consistir en una colección de programas de diferentes proveedor.
3. **Red de Comunicaciones**: es el software en cada nodo que lo liga con una red que incluye la descripción completa de los nodos y las líneas de red. De igual manera, identifica para cada nodo el procesamiento realizado, la capacidad de almacenamiento, potencia de procesamiento y su estado actual. Identifica para cada vínculo los nodos que conecta, el tipo de vínculo, el ancho de banda, los protocolos requeridos y su estado actúa
4. **Diccionario de datos global**: es una recopilación de información acerca de la base de datos distribuida. Incluye una lista de todos los aspectos de sus datos (bajo la forma de metadatos), su ubicación y otra clase de información sobre cualquier dato almacenado en cualquier parte del sitio.

Los diccionarios pueden ser de tipo:

* **Centralizado:** es global en una única sed. El inconveniente es que se transforma en un cuello de botella.
* **Descentralizado** es local en cada nodo. El inconveniente es que la búsqueda no local es costosa.
* **Esquema mixto** es global en un único nodo y local en cada sede.

**CARACTERÍSTICAS DE LA BASES DE DATOS DISTRIBUIDAS**

* Los datos deben estar físicamente en más de un computador, o sea distintas sedes.
* Las sedes deben estar interconectadas. Cada sede es un nodo de la red.
* Los datos deben tener un esquema lógico global y único para poder ser recuperados y actualizados.
* En una única operación se debe poder acceder a los datos que se encuentran en más de una sede.
* Todas las acciones que se necesiten realizar sobre más de una sede deben ser transparentes al usuario.

**INDEPENDENCIA DE DATOS**

Los datos en las bases de datos distribuidas deben contar con la característica de independencia. Esa independencia puede ser tanto lógica como física:

* **Independencia lógica de datos** se refiere a la inmunidad de las aplicaciones de usuario a los cambios en la estructura lógica de la base de datos, permitiendo que un cambio en la definición de un esquema no afecte a las aplicaciones del usuario. Por ejemplo, agregar un nuevo atributo a la relación, la creación de una nueva relación, el reordenamiento lógico de algunos atributos, etcétera.
* **La Independencia física** de los datos se refiere al ocultamiento de los detalles sobre las estructuras de almacenamiento a las aplicaciones de usuario. La descripción física de datos puede cambiar sin afectar a las aplicaciones. Por ejemplo, los datos pueden ser movidos de un disco a otro o la organización de los datos en el disco puede cambiar.

**TRANSPARENCIA**

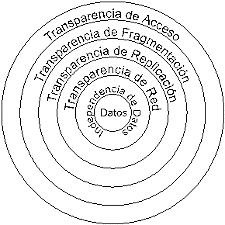
Otra característica de las bases distribuidas es la transparencia. Su objetivo es ocultar al usuario los detalles de diseño, es decir, el usuario no tiene que saber que se encuentra trabajando con un sistema distribuido mediante independencia de datos.

Debe asegurar que 2 sitios no utilicen el mismo nombre para elementos de datos diferentes.

Una solución a este problema es exigir que todos los nombres se registren en un servidor de nombres central.

Un enfoque alternativo muy utilizado propone que cada sitio anteponga su propio identificador de sitio a cualquier nombre que genere, garantizando así que 2 sitios diferentes no generen nunca el mismo nombre, dado que cada sitio tiene un identificador único.

**NIVELES DE TRANSPARENCIA**



El propósito de establecer una arquitectura de un sistema de bases de datos distribuida es ofrecer un nivel de transparencia adecuado para el manejo de la información. Para ellos estructura por niveles:

* A partir de la Independencia de datos **el primer nivel soporta la transparencia de red**: es la ocultación al usuario de los componentes de este sistema distribuido y están separados. Por lo tanto, el usuario percibirá que él es un único sistema y no varios componen.
* El **segundo nivel permite la transparencia de replicación de datos** y se refiere a que, si existen réplicas de datos, estas deben ser controladas por el sistema gestor, no por el usuario. La función principal de la transparencia de replicación es la de mantener la consistencia entre las copias. La replicación es necesaria para obtener un mayor rendimiento debido a que dispone de copias locales y mayor disponibilidad ya que los datos son accesibles siempre al tener varias copias. La principal desventaja, es que hay que mantener actualizadas todas las copias de ese dato replicado y esto lleva el problema de la denominada propagación de las actualizaciones.
* **El tercer nivel permite la transparencia de fragmentación**. El acceso a la base de datos es de forma global, es decir, el usuario no necesita especificar los nombres de los fragmentos ni las ubicaciones de los datos. Los usuarios deben comportarse como si los datos en realidad no estuvieran distribuidos, el sistema maneja la conversión de consultas de usuarios definidas sobre relaciones globales a consultas definidas sobre fragmentos. Las respuestas a consultas fragmentadas resultan en una sola respuesta global.
* **El cuarto nivel permite la transparencia de acceso o localización** por medio de un lenguaje de manipulación de datos que le permite acceder al usuario a los datos sin tener en cuenta la ubicación de estos (No necesitas saber dónde está ese dato para utilizarlo). Se consigue cuando los administradores de transacciones distribuidas pueden determinar la localización de los datos y emitir acciones a los administradores apropiados. Esto se puede Ejecutar cuando los administradores de transacciones tienen acceso a los directorios de localización de datos. Los administradores de transacciones, los sistemas, necesitan conocer si los datos cambian de lugar ya que las transacciones ignoran la modificación en la localización.

**SGBDD : SISTEMA GESTOR DE UNA BASE DE DATOS DISTRIBUIDA**

Es el software que gestiona esa base de datos y suministra mecanismos de acceso que hace que la distribución sea transparente al usuario.

**Características**

* Administra una base de datos distribuida como si estuviera almacenada en una sola ubicación.
* Se usa para crear recuperar actualizar y eliminar bases de datos distribuidas.
* Sincroniza la base de datos periódicamente proporcionando mecanismos de acceso, en virtud de los cuales la distribución se vuelve transparente para el usuario.
* Asegura que los datos modificados en cualquier sitio se actualicen globalmente.
* Se utiliza en áreas de aplicación donde numerosos usuarios procesan que acceden a grandes volúmenes de datos simultáneamente. Por ejemplo, organizaciones que tienen oficinas en distintas ubicaciones geográficas.
* Mantiene la integridad de los datos y la seguridad de acceso.
* Mantiene el diccionario global.
* Controla la concurrencia de transacciones.

**TIPOS DE BASES DE DATOS DISTRIBUIDAS**

En un sistema distribuido ideal, los sitios comparten un esquema global común. Aunque algunas relaciones se puedan almacenar sólo en alguno de estos sitios, todos los sitios deberían ejecutar el mismo software de gestión de bases de datos y los sitios deberían conocer la existencia de los demás. Si una base de datos distribuida se construye partiendo de cero realmente se debería lograr estos objetivos anteriores, sin embargo, en la realidad una base de datos distribuida se tiene que construir enlazando múltiples sistemas de bases de datos que ya existen, cada uno con su propio esquema y posiblemente ejecutando diferentes software de gestión de bases de datos. A veces tales sistema reciben el nombre de sistemas o bases de datos distribuidos que pueden ser homogéneos o heterogéneos:

* En los **sistemas homogéneos** todos los sitios tienen idénticos software de sistemas gestores. Son conscientes de la existencia de los demás sitios y acuerdan cooperar en el procesamiento de las solicitudes de usuarios.
* **Las bases de datos heterogéneas** tienen sitios diferentes, puede que utilicen esquemas diferentes, por ejemplo, relacional y orientado a objetos y diferentes software de gestión de sistemas. Puede que unos sitios no sean conscientes de la existencia de los demás y puede que solo proporcione facilidades limitadas para la cooperación en el procesamiento de transacciones. Se supone que los sistemas con bases de datos múltiples ofrecen la ilusión de un solo sistema de bases de datos integrado, por lo que hay que utilizar un modelo de datos común. Se caracteriza por manejar diferentes gestores en los nodos locales, que pueden ser de tipos diferentes, múltiples esquemas de bases de datos, etcétera. La integración de todos ellos se realiza mediante software. Otra dificultad es proporcionar un esquema conceptual común. Cada sistema local ofrece su propio esquema conceptual, el sistema de bases de datos múltiples debe integrar esos esquemas independientes en uno común. La integración de los esquemas es una tarea complicada sobre todo por la heterogeneidad semántica. La integración de los esquemas no es la mera traducción directa de unos lenguajes de definición de datos a otro. Puede que los mismos nombres de atributos aparezcan en diferentes bases de datos locales con significados distintos, puede que los tipos de datos utilizados en un sistema no estén soportados por los demás. Todas estas diferencias deben registrarse de manera adecuada en el esquema conceptual global para poder generar funciones de traducción correctas.

**ALMACENAMIENTO DISTRIBUIDO**

Es un problema fundamental en las bases de datos distribuida es cómo guardar los datos de manera redundante, de manera que incluso si falla un nodo de la arquitectura en particular los datos se puedan recuperar a partir de otras fuentes. A partir de unas relacion R o tabla es posible utilizar las siguientes técnicas:

1. **Replicación** donde el sistema mantiene varias réplicas idénticas de una relación r o tabla. Cada réplica se almacena en un sitio diferente.
2. **Fragmentación** una relación se particionada en varios fragmentos y cada fragmento es almacenado en un sitio diferente.
3. Una combinación de ambas **replicación y fragmentación** una relación es particionada en varios fragmentos y el sistema mantiene varias copias de estos.

**REPLICA**

El sistema conserva copias idénticas de la relación y guarda cada réplica en un sitio diferente. La replicación de datos se usa para crear instancias adicionales de datos en diferentes partes de la base. Usando esta táctica una base de datos distribuida puede evitar el tráfico excesivo, porque se puede acceder a datos idénticos en forma local y facilita el suministro de datos desde cualquier sección a otra si los datos de esta última sección se ven comprometidos por cualquier tipo de error. En general la réplica mejora el rendimiento de las operaciones de lectura y aumenta la disponibilidad de los datos, sin embargo, las transacciones de actualización suponen una mayor sobrecarga: el sistema debe asegurar que todas las réplicas de la relación sean consistentes. Cada actualización sobre la relación o tabla debe ser propagada a los sitios, conteniendo las respectivas réplicas. El control de las actualizaciones concurrentes de los datos replicados resulta más complejo que en los sistemas centralizados. En las bases de datos distribuidas, las actualizaciones pueden configurarse según la necesidad de los datos correctos momento a momento, o durante cualquier período de tiempo pudiendo asumir los siguientes formatos: de forma periódica o en forma continua.

* En la **réplica periódica** se actualizan los datos en la réplica y recién en periodos determinados de tiempo, se sincronizan con el repositorio principal. Este es el caso en que no se tenga una necesidad perentoria de datos actualizados en la base
* Cuando es muy necesario disponer de los datos actualizados en los diferentes sitios de la base, **la actualización continua** es la que se utiliza. Cada actualización de los datos en la réplica genera una sincronización en el repositorio principal en tiempo real algo

**FRAGMENTACIÓN**

Al utilizar la técnica de fragmentación, el sistema divide la relación original o tabla en varios fragmentos y guarda cada fragmento en un sitio diferente. Debe respetarse 3 reglas:

1. **La integridad** que impida la no pérdida de información, o sea, si una relación o tabla es descompuesta en fragmentos cada dato de la tabla original debe ser encontrado en uno de los fragmentos.
2. **La reconstrucción** que implica preservar dependencias. Deberá ser siempre posible reconstruir la relación global a partir de los fragmentos.
3. **El desacoplamiento.** Para entender la tercera regla es necesario conocer que hay 3 esquemas de fragmentación de las relaciones: **fragmentación horizontal, vertical y mixta** . Si la relación original es descompuesta horizontalmente en fragmentos, un mismo dato no puede estar en 2 fragmentos diferentes. Si la relación original es descompuesta verticalmente su clave primaria debe estar en todos los fragmentos.

Forma

Descripción generada automáticamente

**LA FRAGMENTACION HORIZONTAL** se usa comúnmente para situaciones en que las ubicaciones específicas generalmente sólo necesitan acceso a la base de datos para gestionar los registros correspondientes a su dominio.

Veamos un ejemplo: A partir de la tabla original CLIENTES, donde sus atributos son: el código, el nombre, la categoría de compra mayorista minorista, la sucursal donde compra y el saldo actual de la deuda.



Hay 6 registros en este ejemplo, aplicando fragmentación horizontal la relación r la tabla original clientes se divide en varios subconjuntos.

Tabla

Descripción generada automáticamente

Cada registro de la relación original debe pertenecer como mínimo a uno de los fragmentos, de modo que se pueda reconstruir la relación original si fuera necesaria. Tenemos en el ejemplo 2 fragmentos: se separaron los clientes minoristas y los mayoristas. Podría haber sido también una fragmentación de acuerdo con la sucursal y tendríamos 3 fragmentos de clientes minoristas de la sucursal tigre, clientes minoristas de la sucursal Belgrano y clientes minoristas de la sucursal Quilmes.

**CON LA FRAGMENTACIÓN VERTICAL** la tabla original se fragmenta en función de los atributos .

Tabla

Descripción generada automáticamente

Tenemos entonces los fragmentos: clientes por sucursal y clientes de deuda. Este tipo de formato funciona bien para situaciones en las que las sedes de la base de datos distribuida interactúan con las mismas cuentas, pero orientadas a diferentes dominios, como por ejemplo, la información de contacto del cliente y consulta de cuestiones financieras.

Tabla

Descripción generada automáticamente**EN LA FRAGMENTACION MIXTA** se aplican ambas fragmentaciones en forma secuencial, y es indistinto el orden

En nuestro ejemplo, la tabla original clientes es fragmentada horizontalmente y tenemos el fragmento clientes minoristas. A partir de este fragmento se aplica fragmentación vertical y tenemos entonces clientes minoristas sucursal y clientes minoristas deuda. En todos los casos se conserva la clave primaria para poder acceder a los datos de los fragmentos y para reconstruir la tabla original si fuera necesario.

**TRANSACCIONES DISTRIBUIDAS**

Para asegurar las propiedades ASCII (atomicidad, consistencia, aislamiento y durabilidad) de las transacciones que se ejecutan en un nodo, el sistema gestor de la base de datos distribuida contiene 2 subsistemas:

* **El Sistema Coordinador de transacciones**
  + Coordina la ejecución de las diferentes transacciones (tanto locales como globales) iniciadas en un nodo.
  + Comienza la ejecución de una transacción descomponiéndola en un número de sus transacciones y las distribuye en los sitios apropiados para su ejecución.
  + Coordina la terminación de una transacción, lo que puede resultar en que las transacciones estén cometidas en todos los sitios o abortadas en todos los sitios.
  + Es el encargado de la ejecución del protocolo de compromiso en 2 fases: “handshaking" and two phase commit. La atomicidad de las transacciones es un aspecto crítico de la construcción de un sistema distribuido de base de datos para no llevarlo a un estado de inconsistencia. El protocolo de compromiso de 2 fases asegura que tales situaciones no se produzcan. La idea básica de este protocolo recordemos es que el coordinador comprometa la transacción solo si la transacción alcanza el estado preparada en cada sitio donde debería ejecutarse. En otro caso, por ejemplo, si la transacción se canceló en algún sitio, el sistema coordinador cancela la transacción. Todos los sitios donde la transacción se ejecutó deben acatar la decisión del sistema coordinador. Si un sitio falla cuando una transacción se está ejecutando, cuando el sitio se recupere del fallo, debería estar en posición de cancelar la transacción y volver todos los registros al estado anterior.
  + El control de concurrencia es otra característica de una base de datos distribuida. Como una transacción puede acceder a elementos de datos de varios sitios, los administradores de transacciones de esos sitios pueden necesitar coordinarse para implementar el control de concurrencia. Si se utiliza el bloqueo, como casi siempre sucede en la práctica, éste se puede realizar de forma local en los sitios que contienen los elementos de datos accedidos. Pero también existe la posibilidad de un interbloqueo que involucre las transacciones originadas en múltiples sitios.
* **El Gestor transacciones** gestiona la ejecución de las transacciones o subtransacciones que acceden a los datos almacenados en el sitio local. Estas transacciones pueden ser locales o formar parte de una transacción global. Es el encargado de definir la estructura de las transacciones, mantener la consistencia en la Base de Datos cuando se ejecuta una transacción o se cancela la ejecución, mantener protocolos de fiabilidad, implementar algoritmos para el control de la concurrencia y sincronizar las transacciones que se ejecuten simultáneamente. El sistema gestor recibe solicitudes de procesamiento de transacciones del sistema coordinador y las traduce en acciones.

**PROPAGACIÓN DE ACTUALIZACIONES**

Para propagar cualquier modificación que se haga sobre un registro a todas las copias almacenadas de ese registro, se cuenta con 2 estrategias:

* **la propagación inmediata** desde la sede primaria donde se produjo la actualización a todas las réplicas en tiempo real.
* **la propagación diferida** en la sede primaria se encarga de actualizar la siguiente copia y ésta a la siguiente en forma secuencial con posterioridad a la modificación.

Diagrama

Descripción generada automáticamente