

# Estrategias de Persistencia

Primer Cuatrimestre 2020.

Clase Nro 2.

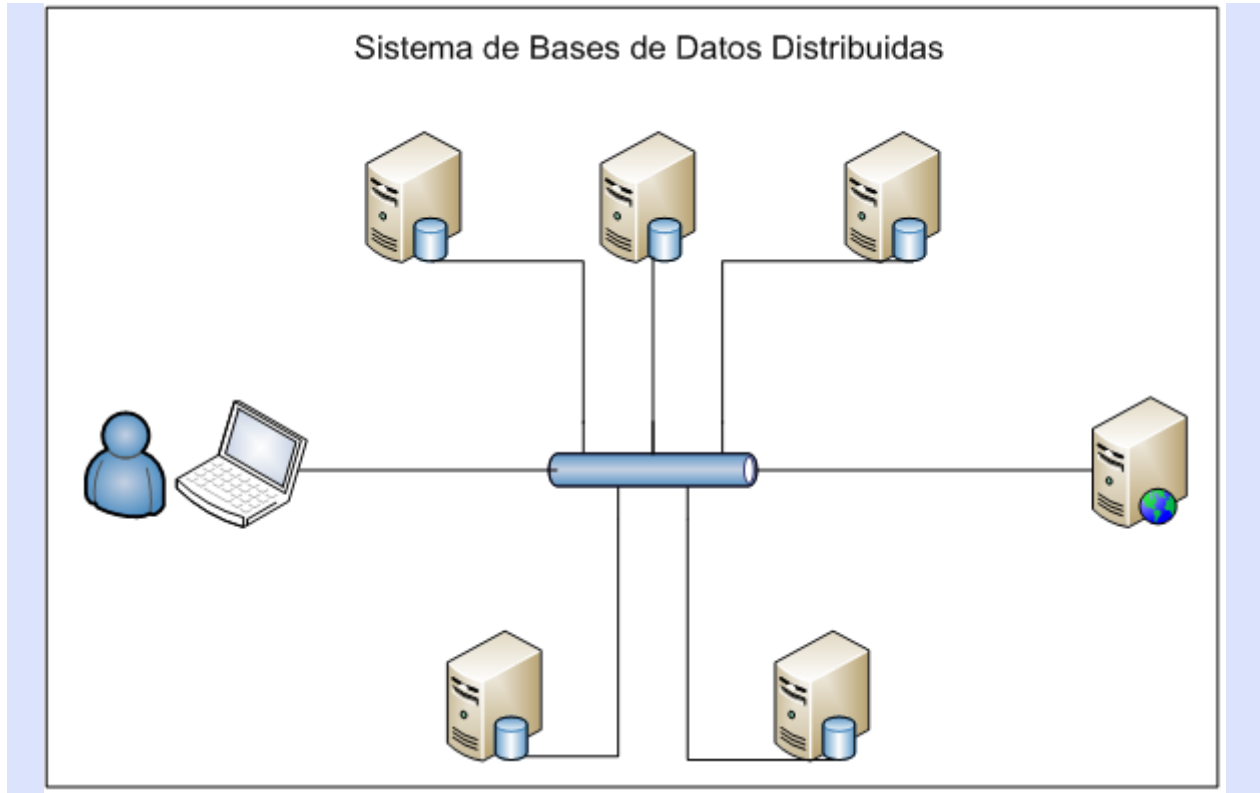
## **Introducción**

En este documento encontraremos un resumen del **capítulo 22** de la bibliografía brindada, sobre conceptos de:

- Base de datos distribuidas
- Ventajas de BDD
- Desventajas de BDD
- Bases de datos homogéneas y heterogéneas
- Almacenamiento distribuido de datos
- Réplica de datos
- Transparencia
- Modos de fallo del sistema
- Pequeño resumen general del capítulo 22.

## ¿Que es una Base de Datos Distribuida?

Una Base de Datos Distribuida (BDD) es, una base de datos construida sobre una red de computadores. La información que estructura la base de datos está almacenada en diferentes sitios en la red, y los diferentes sistemas de información que las utilizan accesan datos en distintas posiciones geográficas.



Por ende una Base de Datos Distribuida es una colección de datos que pertenecen lógicamente a un solo sistema, pero se encuentra físicamente distribuido en varios computadores o servidores de datos en una red de computadoras. Un sistema de bases de datos distribuidas se compone de un conjunto de sitios lógicos, conectados entre sí, mediante algún tipo de red de comunicaciones, en el cual:

Cada sitio lógico puede tener un sistema de base de datos.

Los sitios han sido diseñados para trabajar en conjunto, con el fin de que un usuario de cualquier posición geográfica pueda obtener acceso a los datos desde cualquier punto de la red tal como si todos los datos estuvieran almacenados en la posición propia del usuario. Entonces, la llamada "Base de Datos Distribuida" es en realidad una especie de "objeto virtual", cuyos componentes se almacenan físicamente en varias "bases de datos reales" ubicadas en diferentes sitios. En esencia es la unión lógica de esas diferentes bases de datos.

En otras palabras, cada sitio tiene sus propias "bases de datos reales" locales, sus propios usuarios locales, sus propios SGBD y programas para la administración de transacciones y su propio administrador de comunicación de datos. Así pues, el sistema de bases de datos distribuidas puede considerarse como una especie de sociedad entre los diferentes SGBD individuales locales. Un nuevo componente de

software en cada sitio realiza las funciones de sociedad necesarias; y es la combinación de este nuevo componente y el SGBD ya existente constituyen el llamado Sistema de Administración o Gestión de Bases de Datos Distribuidas - SGBDD. (En ingles DDBMS, Distributed DataBase Management System).

Desde el punto de vista del usuario final, un sistema distribuido deberá ser idéntico a un sistema no distribuido. Los usuarios de un sistema distribuido se comportan en su manipulación de información exactamente como si el sistema no estuviera distribuido. Todos los problemas de los sistemas distribuidos son de tipo interno o a nivel de realización, no pueden existir problemas de tipo externo o a nivel del usuario final.

Los datos que se encuentran distribuidos en varios sitios y que están interconectados por una red de comunicaciones tienen capacidad de procesamiento autónomo de transacciones y hacer procesos locales. Cada sitio realiza la ejecución de al menos una transacción global, la cual requiere accesos a datos en diversos sitios.

Los principios fundamentales de un sistema de datos distribuido son:

1. Autonomía local.
2. No dependencia de un sitio central.
3. Operación continua.
4. Independencia con respecto a la localización.
5. Independencia con respecto a la fragmentación.
6. Independencia de réplica.
7. Procesamiento distribuido de consultas.
8. Manejo distribuido de transacciones.
9. Independencia con respecto al equipo.
10. Independencia con respecto al sistema operativo.
11. Independencia con respecto a la red.
12. Independencia con respecto al SGBD.

Estos doce postulados no son todas independientes entre sí, ni tienen todos la misma importancia que cada usuario le otorgue. Sin embargo, sí son útiles como fundamento para entender la tecnología distribuida y como marco de referencia para caracterizar la funcionalidad de sistemas distribuidos específicos.

1. Autonomía local: Los sitios o posiciones de un sistema distribuido deben ser autónomos. La autonomía local significa que todas las operaciones en un sitio determinado se controlan en ese sitio; ningún sitio A

deberá depender de algún otro sitio B para su buen funcionamiento (pues de otra manera el sitio A podría ser incapaz de trabajar, aunque no tenga en sí problema alguno, si cae el sitio B). La autonomía local significa que existe un propietario y un administrador local de los datos, con responsabilidad local: todos los datos pertenecen a una base de datos local, aunque los datos sean accesibles desde algún sitio distante. Todo el manejo de la seguridad y la integridad de los datos se efectúan con control de la instalación y administración local.

2. No dependencia de un sitio central: La no dependencia de un sitio central, sería lo ideal pero si esto no se logra la autonomía local completa se vería comprometida. La dependencia de un sitio central no es práctica al menos por las siguientes razones: en primer lugar, estos sitios podrían generar un cuello de botella, y en segundo lugar, el sistema sería vulnerable; si el sitio central sufriera un desperfecto, todo el sistema dejaría de funcionar.

3. Operación continua: En un sistema distribuido, lo mismo que en uno no distribuido, nunca debería haber necesidad de apagar o dejar de funcionar. Es decir, el sistema nunca debería necesitar apagarse para que se pueda realizar alguna operación, como añadirse un nuevo sitio o instalar una versión mejorada del SGBD en un sitio ya existente.

4. Independencia con respecto a la localización: La independencia con respecto a la localización, permite que los usuarios finales no sepan donde están almacenados físicamente los datos, sino que trabajen como si todos los datos estuvieran almacenados en su propio sitio local. La independencia con respecto a la localización es deseable porque simplifica los sistemas de información de los usuarios y sus actividades en la terminal. Esto hace posible la migración de datos de un sitio a otro sin anular la validez de ninguno de esos sistemas o actividades. Esta función de migración permite modificar la distribución de los datos dentro de la red, en respuesta a cambios en los requerimientos de desempeño.

5. Independencia con respecto a la fragmentación: Un sistema tiene fragmentación de datos solo si es posible dividir una relación en partes o "fragmentos" para propósitos de almacenamiento físico. La fragmentación es deseable por razones de desempeño: los datos pueden almacenarse en la localidad donde se utilizan con mayor frecuencia, de manera que la mayor parte de las operaciones sean solo locales y se reduzca el tráfico en la red de cómputo. Existen en esencia dos clases de fragmentación, la fragmentación horizontal y la fragmentación vertical; estos tipos de fragmentación son correspondientes a las operaciones relacionales de restricción y proyección, respectivamente. Un fragmento puede ser cualquier sub relación que pueda derivarse de la relación original mediante operaciones de restricción y proyección; la reconstrucción de la relación originada a partir de los fragmentos se hace mediante operaciones de reunión y unión (reunión en el caso de fragmentación vertical, y la unión en casos de fragmentación horizontal).

6. Independencia de réplica: Un sistema maneja réplica de datos si una relación dada se puede representar en el nivel físico mediante varias copias réplicas, en muchos sitios distintos. La réplica es deseable al menos por dos razones: en primer lugar, puede producir un mejor desempeño (las aplicaciones pueden operar sobre copias locales en vez de tener que comunicarse con sitios remotos); en segundo lugar, también puede significar una mejor disponibilidad (un objeto estará disponible para su procesamiento en tanto esté disponible por lo menos una copia, al menos para propósitos de recuperación). La desventaja principal de las réplicas es cuando se pone al día un cierto objeto copiado, deben ponerse al día todas las réplicas de ese objeto. La réplica debe ser "transparente para el usuario final", un sistema que maneja la réplica de los datos deberá ofrecer también una independencia de réplica (conocida también como

transparencia de réplica); es decir, los usuarios deberán comportarse como si sólo existiera una copia de los datos.

7. Procesamiento distribuido de consultas: Este manejo de datos en las consultas permite las consultas eficientes desde diferentes usuarios con las características que determine el sistema; la consulta de datos es más importante en un sistema distribuido que en uno centralizado. Lo esencial es que, en una consulta donde están implicados varios sitios, habrá muchas maneras de trasladar los datos en la red de cómputo para satisfacer la solicitud, y es crucial encontrar una estrategia suficiente. Por ejemplo, una solicitud de unión de una relación Rx almacenada en el sitio X y una relación Ry almacenada en el sitio Y podría llevarse a cabo trasladando Rx a Y o trasladando Ry a X, o trasladando las dos a un tercer sitio Z.

8. Manejo distribuido de transacciones: Este manejo tiene dos aspectos principales, el control de recuperación y el control de concurrencia, cada uno de los cuales requiere un tratamiento más amplio en el ambiente distribuido. En un sistema distribuido, una sola transacción puede implicar la ejecución de programas o procesos en varios sitios (en particular puede implicar actualizaciones en varios sitios). Por esto, cada transacción está compuesta de varios agentes, donde un agente es el proceso ejecutado en nombre de una transacción dada en determinado sitio. Y el sistema necesita saber cuándo dos agentes son parte de la misma transacción. Es importante aclarar que no puede haber un bloqueo mutuo entre dos agentes que sean parte de la misma transacción.

9. Independencia con respecto al equipo: Las instalaciones de cómputo en el mundo real por lo regular incluyen varias máquinas de diferentes marcas comerciales como IBM, DELL, HP, SUN, entre otras; por esta razón existe una verdadera necesidad de poder integrar los datos en todos esos sistemas y presentar al usuario "una sola imagen del sistema". Por tanto conviene ejecutar el mismo SGBD en diferentes equipos, y además lograr que esos diferentes equipos se integren en un sistema distribuido.

10. Independencia con respecto al sistema operativo: Es necesario y conveniente no sólo de poder ejecutar el mismo SGBD en diferentes equipos, sino también poder ejecutarlo en diferentes sistemas operativos y lograr que una versión LINUX y una WINDOWS participen todas en el mismo sistema distribuido.

11. Independencia con respecto a la red: Si el sistema puede manejar múltiples sitios, con equipos distintos y diferentes sistemas operativos, resulta obvia la conveniencia de manejar también varios tipos de redes de comunicación distintas.

12. Independencia con respecto al SGBD: En la independencia con respecto a su manejo, se requiere que los SGBD en los diferentes sitios manejen todos la misma interfaz; no necesitan ser por fuerza copias del mismo sistema.

## Ventajas de las BDD

Las ventajas de utilizar sistemas de Bases de Datos Distribuidas son:

Favorecer la naturaleza distribuidora de muchas aplicaciones, no solamente a nivel local sino incluso en diferentes lugares.

Existe un acercamiento bastante notable a las estructuras organizacionales de las empresas o sectores productivos.

Se consigue compartir los datos, sin perder el control local.

El rendimiento es más alto cuando se distribuye una gran base de datos por múltiples sitios, las consultas locales y las transacciones tienen mejor rendimiento porque las bases de datos locales son más pequeñas. A parte de esta distribución, se puede conseguir lo siguiente en estos sistemas:

1. Reducir el número de transacciones ejecutándose por sitio.
2. Un paralelismo entre las consultas ejecutando varias de estas en sitios diferentes.
3. Descomposición de consultas en sub consultas que pueden ejecutarse en paralelo.
4. Aumento de la fiabilidad y la disponibilidad de los datos.
5. Los procesos de expansión se hacen más sencillos en un sistema distribuido; cuando se quiere añadir un nuevo sitio en nuestro sistema distribuido, el resto de operaciones que se ejecutan en los demás no se verán afectados.
6. La autonomía de estos sistemas es alta.

Para garantizar las ventajas anteriormente descritas, es necesario que el software del SGBDD proporcione las siguientes funciones adicionales a las que un sistema centralizado ya tiene de por sí:

Capacidad para seguir la traza a la distribución y replicación de los datos en el catálogo del SGBDD

Capacidad para acceder a sitios remotos.

Capacidad de implementar estrategias de ejecución para consultas y transacciones que tiene acceso a datos en más de un sitio.

Capacidad de recuperarse de errores de sitios o fallos de la red.

Capacidad de direccionar a qué copia de un elemento de datos replicado acceder y de mantener la consistencia de las copias.

Gestión del directorio o catálogo distribuido de la BDD.

Implementación apropiada de la seguridad de los datos y los privilegios de autorización de los usuarios finales.

## Desventajas de las BDD

Las desventajas de utilizar sistemas de Bases de Datos Distribuidas son:

Hay una menor seguridad en cuanto al control de acceso a los datos: control de réplicas y errores que puedan producirse en la red.

Mayor complejidad en el diseño e implementación del sistema. Además si la replicación de datos no se hace de forma adecuada, las ventajas se pueden transformar en desventajas.

Excesivos costos en el intento de conseguir la transparencia mencionada anteriormente.

Falta de estándares y de experiencia en estos modelos avanzados de BD.

No se puede garantizar al 100 % el rendimiento y la fiabilidad.

## Bases de datos homogéneas y heterogéneas

En los sistemas de bases de datos distribuidas homogéneas todos los sitios emplean idéntico software de gestión de bases de datos, son conscientes de la existencia de los demás sitios y acuerdan cooperar en el procesamiento de las solicitudes de los usuarios. En estos sistemas, los sitios locales renuncian a una parte de su autonomía en cuanto a su derecho a modificar los esquemas o el software de gestión de bases de datos. Ese software también debe cooperar con los demás sitios en el intercambio de la información sobre las transacciones para hacer posible su procesamiento entre varios sitios.

A diferencia de lo anterior, en las bases de datos distribuidas heterogéneas puede que los diferentes sitios utilicen esquemas y software de gestión de sistemas de bases de datos diferentes. Puede que algunos sitios no tengan información de la existencia del resto y que sólo proporcionen facilidades limitadas para la cooperación en el procesamiento de las transacciones. Las diferencias en los esquemas suelen constituir un problema importante para el procesamiento de las consultas, mientras que la divergencia del software supone un inconveniente para el procesamiento de transacciones que acceden a varios sitios.

## Almacenamiento distribuido de datos

Considérese una relación  $r$  que hay que almacenar en la base de datos. Existen dos enfoques del almacenamiento de esta relación en la base de datos distribuida:

- **Réplica.** El sistema conserva varias réplicas (copias) idénticas de la relación y guarda cada réplica en un sitio diferente. La alternativa a las réplicas es almacenar sólo una copia de la relación  $r$ .
- **Fragmentación.** El sistema divide la relación en varios fragmentos y guarda cada fragmento en un sitio diferente. La fragmentación y la réplica pueden combinarse: las relaciones pueden dividirse en varios fragmentos y puede haber varias réplicas de cada fragmento.

En los subapartados siguientes se profundizará en cada una de estas técnicas.

## Réplica de datos

Si la relación  $r$  se replica, se guarda una copia de esa relación en dos o más sitios. En el caso más extremo se tiene una réplica completa, en la que se guarda una copia en cada sitio del sistema. Las réplicas presentan varias ventajas e inconvenientes

- Disponibilidad. Si alguno de los sitios que contiene la relación  $r$  falla, esa relación puede hallarse en otro sitio distinto. Por tanto, el sistema puede seguir procesando las consultas que impliquen a  $r$ , pese al fallo del sitio.
- Paralelismo incrementado. En el caso en el que la mayoría de los accesos a la relación  $r$  sólo resultasen en lecturas, diferentes sitios podrían procesar en paralelo las lecturas que impliquen a  $r$ . Cuantas más réplicas de  $r$  existan, mayor será la posibilidad de que los datos necesarios se encuentren en el sitio en que se ejecuta la transacción. Por tanto, la réplica de los datos minimiza su transmisión entre los diferentes sitios.
- Sobrecarga incrementada durante la actualización. El sistema debe asegurar que todas las réplicas de la relación  $r$  sean consistentes; en caso contrario pueden producirse cálculos erróneos. Por tanto, siempre que se actualiza  $r$ , hay que propagar la actualización a todos los sitios que contienen réplicas. El resultado es una sobrecarga incrementada. Por ejemplo, en un sistema bancario, en el que la información de las cuentas se replica en varios sitios, es necesario asegurarse de que el saldo de cada cuenta concuerde en todos ellos.

## Transparencia

No se debe exigir a los usuarios de los sistemas distribuidos de bases de datos que conozcan la ubicación física de los datos ni el modo en que se puede acceder a ellos en cada sitio local concreto. Esta característica, denominada transparencia de los datos, puede adoptar varias formas:

- Transparencia de la fragmentación. No se exige a los usuarios que conozcan el modo en que se ha fragmentado la relación.
- Transparencia de la réplica. Los usuarios ven cada objeto de datos como lógicamente único. Puede que el sistema distribuido replique los objetos para incrementar el rendimiento del sistema o la disponibilidad de los datos. Los usuarios no deben preocuparse por los objetos que se hayan replicado ni por la ubicación de esas réplicas.
- Transparencia de la ubicación. No se exige a los usuarios que conozcan la ubicación física de los datos. El sistema distribuido de bases de datos debe poder hallar los datos siempre que la transacción del usuario facilite el identificador de esos datos.

## Modos de fallo del sistema

Los sistemas distribuidos pueden sufrir los mismos tipos de fallos que los sistemas centralizados (por ejemplo, errores de software, errores de hardware y fallos de discos). No obstante, en los



entornos distribuidos también hay que tratar con otros tipos de fallos. Los tipos básicos de fallos son:

- Fallos de sitios.
- Pérdidas de mensajes.
- Fallos de enlaces de comunicaciones.
- Divisiones de la red.

En los sistemas distribuidos siempre es posible la pérdida o el deterioro de los mensajes. El sistema utiliza protocolos de control de las transmisiones, como TCP/IP, para tratar esos errores. Se puede encontrar información sobre esos protocolos en los libros de texto estándar sobre redes (véanse las notas bibliográficas). No obstante, si dos sitios A y B no se hallan conectados de manera directa, los mensajes de uno a otro deben encaminarse mediante una serie de enlaces de comunicaciones. Si falla uno de los enlaces, hay que volver a encaminar los mensajes que debería haber transmitido. En algunos casos se puede hallar otra ruta por la red, de modo que los mensajes puedan alcanzar su destino. En otros casos, el fallo puede hacer que no haya ninguna conexión entre los dos sitios. Un sistema está dividido si se ha partido en dos (o más) subsistemas, denominados particiones, que carecen de conexión entre ellas. Obsérvese que, con esta definición, cada subsistema puede consistir en un solo nodo.

### **Resumen (Para profundizar sobre el tema leer capítulo 22 del libro)**

- Los sistemas distribuidos de bases de datos consisten en un conjunto de sitios, cada uno de los cuales mantiene un sistema local de bases de datos. Cada sitio puede procesar las transacciones locales: las transacciones que sólo acceden a datos de ese mismo sitio. Además, cada sitio puede participar en la ejecución de transacciones globales: las que acceden a los datos de varios sitios. La ejecución de las transacciones globales necesita que haya comunicación entre los diferentes sitios.
- Las bases de datos distribuidas pueden ser homogéneas, en las que todos los sitios tienen un esquema y un código de sistemas de bases de datos comunes, o heterogéneas, en las que el esquema y el código de los sistemas pueden ser diferentes.
- Surgen varios problemas relacionados con el almacenamiento de relaciones en bases de datos distribuidas, incluidas la réplica y la fragmentación. Es fundamental que el sistema minimice el grado de conocimiento por los usuarios del modo en que se almacenan las relaciones.
- Los sistemas distribuidos pueden sufrir los mismos tipos de fallos que los sistemas centralizados. No obstante, hay otros fallos con los que hay que tratar en los entornos distribuidos; entre ellos, los fallos de los sitios, los de los enlaces, las pérdidas de mensajes y las divisiones de la red. Es necesario tener en consideración cada uno de esos problemas en el diseño del esquema distribuido de recuperación.

- Para asegurar la atomicidad, todos los sitios en los que se ejecuta la transacción T deben estar de acuerdo en el resultado final de su ejecución. O bien T se compromete en todos los sitios o se aborta en todos. Para asegurar esta propiedad el coordinador de la transacción de T debe ejecutar un protocolo de compromiso. El protocolo de compromiso más empleado es el protocolo de compromiso de dos fases.
- El protocolo de compromiso de dos fases puede provocar bloqueos, la situación en que el destino de una transacción no se puede determinar hasta que se recupere un sitio que haya fallado (el coordinador). Se puede utilizar el protocolo de compromiso de tres fases para reducir la probabilidad de bloqueo.
- La mensajería persistente ofrece un modelo alternativo para el tratamiento de las transacciones distribuidas. El modelo divide cada transacción en varias partes que se ejecutan en diferentes bases de datos. Se envían mensajes persistentes (que está garantizado que se entregan exactamente una vez, independientemente de los fallos) a los sitios remotos para solicitar que se emprendan acciones en ellos. Aunque la mensajería persistente evita el problema de los bloqueos, los desarrolladores de aplicaciones tienen que escribir código para tratar diferentes tipos de fallos.
- Los diferentes esquemas de control de concurrencia empleados en los sistemas centralizados se pueden modificar para su empleo en entornos distribuidos.
  - En el caso de los protocolos de bloqueo, la única modificación que hay que hacer es el modo en que se implementa el gestor de bloqueos. Existen varios enfoques posibles. Se pueden utilizar uno o varios coordinadores centrales. Si, en vez de eso, se adopta un enfoque con un gestor distribuido de bloqueos, hay que tratar de manera especial los datos replicados.
  - Entre los protocolos para el tratamiento de los datos replicados se hallan el protocolo de copia principal, el de mayoría, el sesgado y el de consenso de quórum. Cada uno de ellos representa diferentes equilibrios en términos de coste y de posibilidad de trabajo en presencia de fallos.
  - En el caso de los esquemas de marcas temporales y de validación, el único cambio necesario es el desarrollo de un mecanismo para la generación de marcas temporales globales únicas.
  - Muchos sistemas de bases de datos soportan la réplica perezosa, en la que las actualizaciones se propagan a las réplicas ubicadas fuera del ámbito de la transacción que ha llevado a cabo la actualización. Estas facilidades deben utilizarse con grandes precauciones, ya que pueden dar lugar a ejecuciones no secuenciables.
- La detección de interbloqueos en entornos con gestor distribuido de bloqueos exige la colaboración entre varios sitios, dado que puede haber interbloqueos globales aunque no haya ningún interbloqueo local.
- Para ofrecer una elevada disponibilidad, las bases de datos distribuidas deben detectar los fallos, reconfigurarse de modo que el cálculo pueda continuar y recuperarse cuando se repare el procesador o el enlace correspondiente. La tarea se complica enormemente por el hecho de que resulta difícil distinguir entre las divisiones de la red y los fallos de los

sitios. Se puede extender el protocolo de mayoría mediante el empleo de números de versión para permitir que continúe el procesamiento de las transacciones incluso en presencia de fallos. Aunque el protocolo supone una sobrecarga significativa, funciona independientemente del tipo de fallo. Se dispone de protocolos menos costosos para tratar los fallos de los sitios, pero dan por supuesto que no se producen divisiones de la red.

- Puede que las consultas a las bases de datos distribuidas necesiten tener acceso a varios sitios. Se dispone de varias técnicas de optimización para escoger los sitios a los que hay que tener acceso. Basadas en la fragmentación y en la réplica, las técnicas pueden utilizar técnicas de semirreunión para reducir las transferencias de datos.
- Las bases de datos distribuidas heterogéneas permiten que cada sitio tenga su propio esquema y su propio código de sistema de bases de datos. Los sistemas de bases de datos múltiples ofrecen un entorno en el que las nuevas aplicaciones de bases de datos pueden acceder a los datos de gran variedad de bases de datos ya existentes ubicadas en diferentes entornos heterogéneos de hardware y de software. Puede que los sistemas locales de bases de datos empleen modelos lógicos y lenguajes de definición o de manipulación de datos diferentes y que se diferencien en los mecanismos de control de concurrencia o de administración de las transacciones. Los sistemas de bases de datos múltiples crean la ilusión de la integración lógica de las bases de datos, sin exigir su integración física.