

# MÓDULO IV

**CARACTERÍSTICAS GENERALES DE LOS SGBD:**  
**CONEXIONES EN LA RED**  
**PARALELISMO**  
**DISTRIBUCIÓN**  
**PERFORMANCE**  
**ESCALABILIDAD**  
**TRANSPARENCIA**  
**CONFIABILIDAD**  
**PROCESAMIENTO DE TRANSACCIONES EN LÍNEA**  
**PROCESAMIENTO PARA TOMA DE DECISIÓN.**

Marcela Russo  
Laboratorio IV

# ***CARACTERISTICAS GENERALES DE LOS SGBD***

- La arquitectura de un SGBD está influenciado en gran medida por el sistema informático subyacente en el que se ejecuta la BD.
- En la arquitectura se reflejan aspectos que condujeron al desarrollo de diferentes estructuras:
  - ***CONEXIÓN EN LA RED***
  - ***ARQUITECTURA CENTRALIZADA***
  - ***PARALELISMO***
  - ***DISTRIBUCION***
- No confundir el SGBD con la arquitectura que se elije para implementarlo. Algunos SGBD solo se puede implantar en una de las arquitecturas y otros en todas ellas.

### ➤ **CONEXION EN LA RED**

- La conexión de varias computadoras en red permite que algunas tareas se ejecuten en un sistema servidor y que otras se ejecuten en sistemas clientes.
- Esta división de trabajo condujo al desarrollo de bases de datos cliente-servidor
- El procesamiento en una arquitectura *Cliente-servidor*, dentro de los DBMS, facilita el acceso web a las bases de datos, por lo tanto cualquier LAN o WAN puede considerarse un sistema Cliente/Servidor desde el momento en que el cliente solicita servicios (datos, impresión, etc).

### ➤ **ARQUITECTURA CENTRALIZADA**

- El SGBD está implementado en una sola plataforma (ordenador) desde donde se gestiona directamente, de modo centralizado, la totalidad de los recursos.
- Es la arquitectura más clásica de los centros de proceso de datos tradicionales.
- Se basa en tecnologías sencillas, muy experimentadas y de gran robustez (hardware).
- Los DBMS corporativos hacen uso de servidores mainframe que usan este tipo de arquitectura

### ➤ **PARALELISMO**

- **Un DBMS paralelo usa una colección de recursos (procesadores, discos y memoria) para realizar trabajo en paralelo.**
- **El trabajo se divide entre recursos para lograr niveles deseados de rendimiento (escalamiento y aceleración) y disponibilidad. Usa los servicios de una red de alta rapidez, sistema operativo y sistema de almacenamiento para coordinar la división del trabajo entre recursos.**
- **Escalamiento**
  - ✓ **Involucra la cantidad de trabajo que puede lograrse mediante el aumento de la capacidad de cómputo.**
  - ✓ **Mide el aumento en tamaño de una labor que puede realizarse mientras se mantiene el tiempo constante.**
  - ✓ **Lo ideal es que sea lineal, en el que el aumento en capacidad computacional en  $n$  veces permite la conclusión de  $n$  veces la cantidad de trabajo en el mismo tiempo, no siendo posible en la mayoría de las situaciones debido a los gastos de coordinación.**
  - ✓ **Se mide como la razón de la cantidad de trabajo concluido con la mayor configuración a la cantidad de trabajo concluido con la configuración original.**

### ➤ **PARALELISMO**

- **Aceleración**

- ✓ **Implica la disminución en tiempo para completar una tarea en lugar de la cantidad de trabajo realizado.**
- ✓ **Con capacidad de computación añadida, mide la reducción en tiempo mientras se mantiene la constante de tareas.**
- ✓ **Las organizaciones necesitan determinar la cantidad de capacidad computacional adicional que garantizará la conclusión del trabajo dentro del tiempo permisible.**
- ✓ **Se mide mediante la razón del tiempo de conclusión con la configuración original al tiempo de conclusión con la capacidad adicional.**

- **Disponibilidad**

- ✓ **Es la accesibilidad de un sistema, usualmente medida como el tiempo productivo del sistema.**
- ✓ **Para computación de alta disponibilidad o resistente al fallo, un sistema experimenta poco tiempo improductivo y se recupera rápidamente de los fallos.**
- ✓ **La computación tolerante al fallo lleva la resistencia al límite en tanto que el procesamiento debe continuarse sin interrupción.**

### ➤ **PARALELISMO**

- **Disponibilidad**
  - ✓ **El costo del tiempo improductivo determina el grado de disponibilidad deseada, puede incluir pérdidas de salarios, mano de obra perdida y equipo inactivo. Para una organización grande, este costo del tiempo improductivo puede ser de cientos a miles de dólares por hora.**
  - ✓ **El procesamiento de bases de datos paralelas puede aumentar la disponibilidad porque un DBMS puede ajustarse dinámicamente al nivel de recursos disponibles.**
  - ✓ **Las fallas de una computadora individual no detendrán el procesamiento en otras computadoras disponibles.**

### ➤ **PARALELISMO**

- **Inconveniente del procesamiento de bases de datos paralelas:**
  - ✓ **Implica grandes costos en software de DBMS y software de coordinación especializado.**
  - ✓ **Existen posibles problemas de interoperabilidad, pues se requiere coordinación entre el software DBMS, el sistema operativo y los sistemas de almacenamiento.**
  - ✓ **Los proveedores de DBMS ofrecen poderosas herramientas para desplegar y administrar la enorme complejidad del procesamiento de bases de datos paralelas.**
  - ✓ **Si no se predican mejoras al rendimiento, será un inconveniente significativo. Las mejoras predecibles en rendimiento permiten a las organizaciones planear capacidad adicional y ajustar dinámicamente la capacidad de acuerdo con los volúmenes de procesamiento anticipados y las restricciones en el tiempo de respuesta**

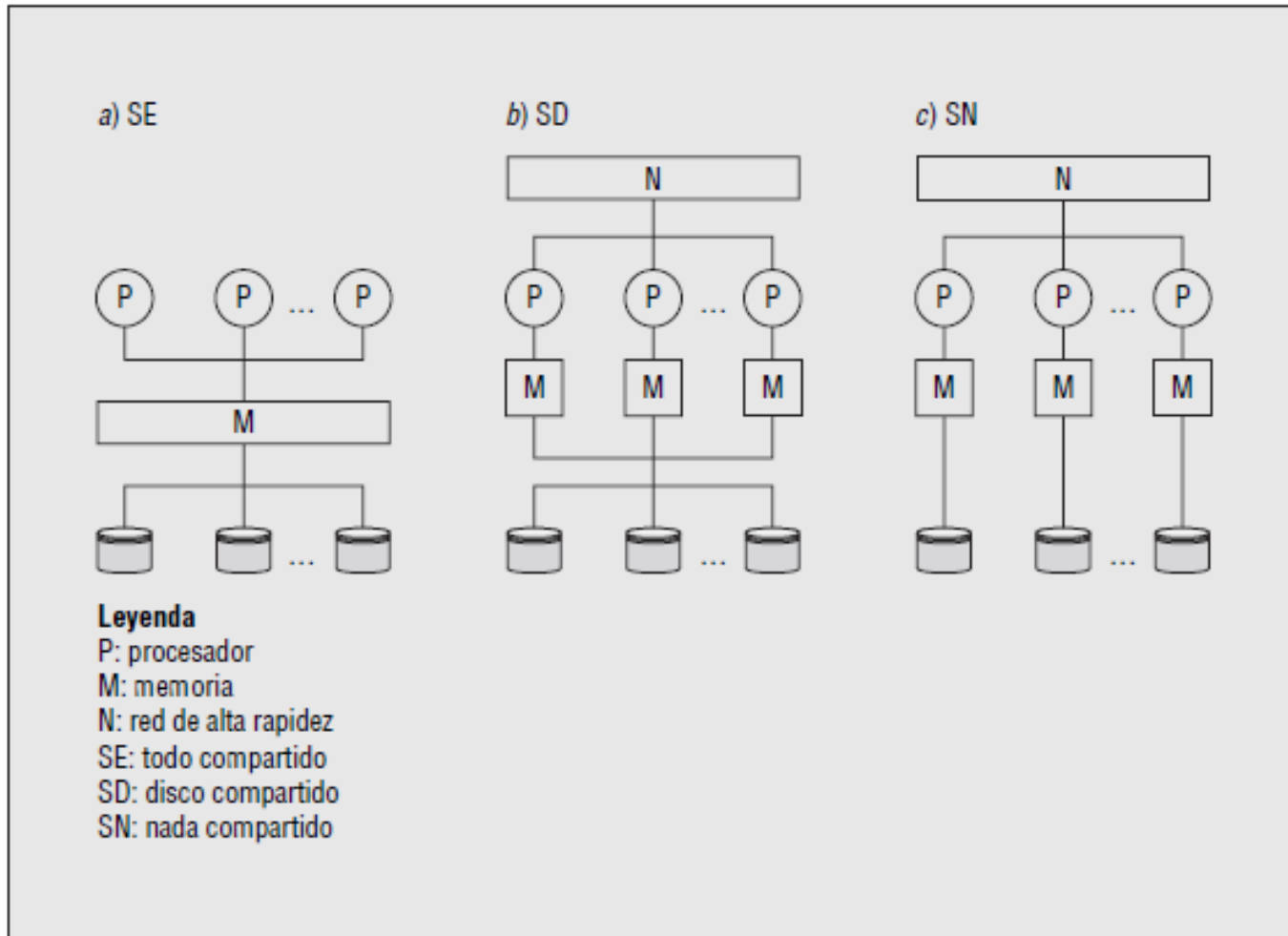
### ➤ **PARALELISMO**

- El grado de recursos compartidos determina las arquitecturas para procesamiento de bases de datos paralelas
- El estándar de clasificación de arquitecturas tiene diferentes enfoques:
  - ✓ **SE: *todo compartido***, memoria y discos se comparten entre una colección de procesadores, usualmente está relacionado como una sola computadora de multiprocesamiento y no como una arquitectura de base de datos paralela. No es necesario el particionamiento, porque los procesadores acceden a todos los datos.
  - ✓ **SD: *discos compartidos***, cada procesador tiene su memoria privada, pero los discos se comparten entre todos los procesadores. No es necesario el particionamiento, porque los procesadores acceden a todos los datos.
  - ✓ **SN: *nada compartido***, los datos se deben particionar entre los procesadores.



## ➤ PARALELISMO

- Grafico de arquitecturas básica de bases de datos paralelas

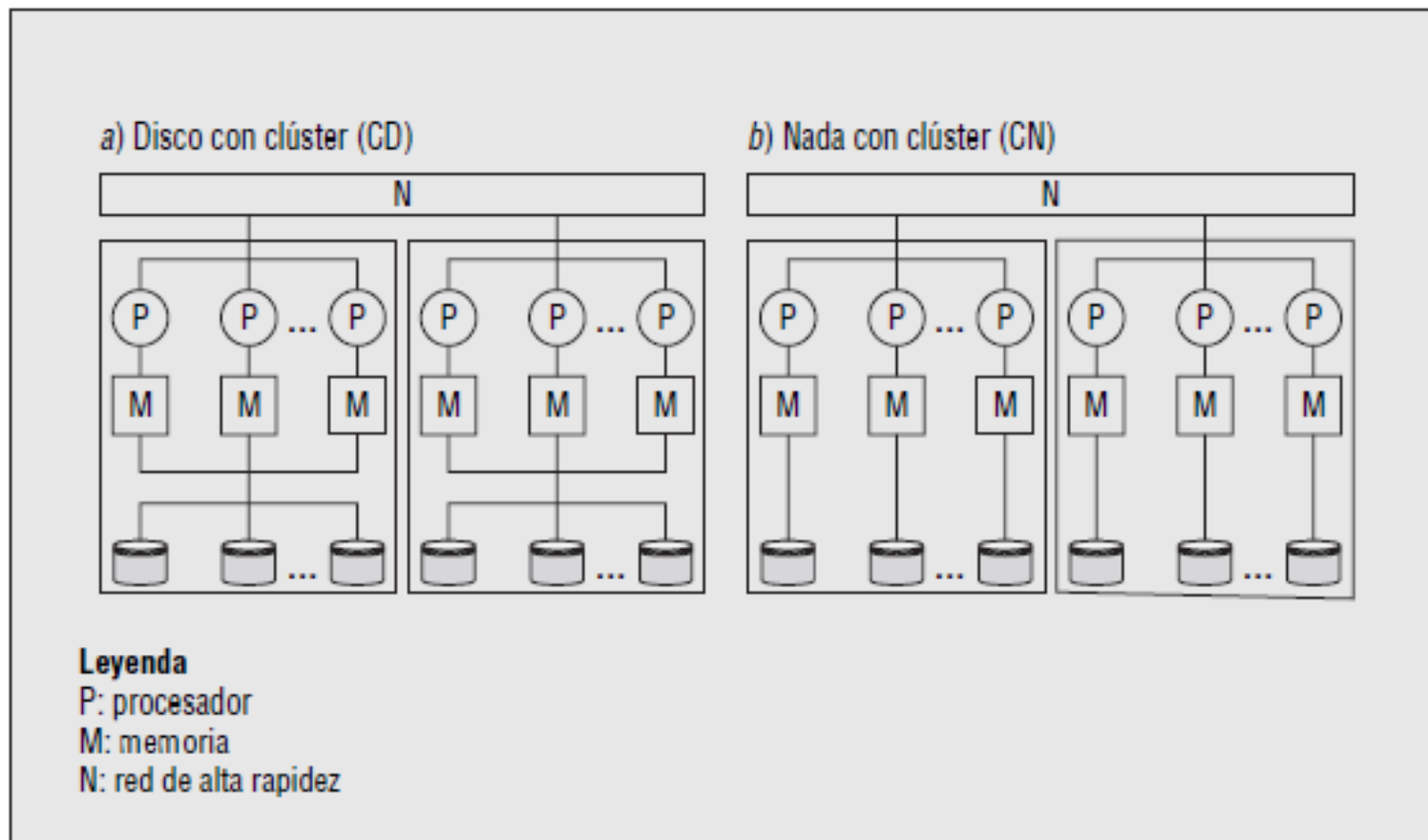


### ➤ **PARALELISMO**

- El procesamiento de bases de datos paralelas divide grandes tareas en muchas tareas más pequeñas y las distribuye entre computadoras interconectadas. Por ejemplo, este tipo de procesamiento puede usarse para realizar una operación de unión en grandes tablas
- El uso de las arquitecturas RAID, es una forma simple de procesamiento de bases de datos paralelas.
- Para flexibilidad adicional, las arquitecturas básicas se extienden mediante *clustering*. Un clúster es un acoplamiento firme de dos o más computadoras, de modo que se comparten como una sola computadora
- En las arquitecturas CD (Disco con clúster) los procesadores en cada clúster comparten todos los discos, pero nada se comparte a través de los clústeres.
- En las arquitecturas CN (Nada con clúster) los procesadores en cada clúster no comparten recursos, pero se puede manipular cada clúster para trabajar en paralelo en la realización de una tarea
- Para mayor flexibilidad, el enfoque de clustering, puede configurar dinámicamente el número de clústeres y la membresía a ellos. Además, cada nodo de procesador en un clúster puede ser una computadora de multiprocesamiento o un procesador individual.

## ➤ PARALELISMO

- Grafico de arquitecturas con clústering de bases de datos paralelas



### ➤ **PARALELISMO**

- La necesidad de procesamiento de consultas en paralelo condujo al desarrollo de los sistemas de bases de datos paralelos.
- En todas las arquitecturas de bases de datos paralelas, el hecho de compartir recursos es transparente a las aplicaciones.
- El código de aplicación (SQL y enunciados en lenguaje de programación) no necesita ser modificado para sacar ventaja del procesamiento de bases de datos paralelas.
- Los conflictos de diseño que influyen en el rendimiento de arquitecturas de bases de datos paralelas son:
  - ✓ **Equilibrio de carga:** involucra la cantidad de trabajo asignado a diferentes procesadores en un clúster. Cada procesador debería tener la misma cantidad de trabajo para utilizar el clúster por completo. Resulta difícil dividir un subconjunto de una base de datos para lograr igual división de trabajo debido a que el sesgo de datos es común entre las columnas de bases de datos
  - ✓ **Coherencia de caché:** implica sincronización entre memorias locales y almacenamiento de disco común. Cuando un procesador direcciona una página de disco, la imagen de esa página permanece en la caché asociada al procesador. Una inconsistencia ocurre si otro procesador cambia la página en su propio búfer. Para evitar inconsistencias, cuando se accede a una página de disco debe hacerse una verificación de otros cachés locales para coordinar los cambios producidos en los cachés de dichos procesadores. Este conflicto se limita a las arquitecturas de disco compartido (SD y CD).

### ➤ **PARALELISMO**

- ✓ **Comunicación interprocesadores:** involucra los mensajes generados para sincronizar acciones de procesadores independientes. El paralelismo particionado, como se usa para las arquitecturas nada compartido, puede crear grandes cantidades de comunicación interprocesadores. En particular, las operaciones conjuntas particionadas pueden generar una gran cantidad de comunicación elevada para combinar resultados parciales ejecutados en diferentes procesadores
- **El procesamiento en paralelo:**
  - ✓ Permite dentro una computadora acelerar las actividades del sistema de BD.
  - ✓ Proporciona a las transacciones respuestas más rápidas.
  - ✓ Da la capacidad de ejecutar más transacciones por segundo.
  - ✓ Las consultas pueden procesarse aprovechando el paralelismo brindado por el sistema informático subyacente.

### ➤ *DISTRIBUCION*

- **Motivación para datos distribuidos:**

- ✓ **Ofrecen algunas ventajas en relación al control de datos, reducción de costos de comunicación y rendimiento mejorado, por lo tanto distribuir una base de datos permite la ubicación de datos de modo que se ajuste a la estructura de una organización**
- ✓ **Las decisiones acerca de compartir y mantener los datos pueden establecerse localmente para proporcionar un control más cercano al uso de datos.**
- ✓ **Los datos deben ubicarse de modo que 80 por ciento de las solicitudes sean locales, debido a que incurren en poco o ningún costo de comunicación y retardos, en comparación con las solicitudes remotas.**
- ✓ **La creciente disponibilidad de datos también puede conducir a rendimiento mejorado, debido a que los datos son más accesibles porque no hay una sola computadora responsable de controlar el acceso y además pueden copiarse de modo que estén disponibles en más de un sitio.**

### ➤ **DISTRIBUCION**

- **Algunos problemas significativos sobre los datos distribuidos están dados por:**
  - ✓ **Un pobre diseño de bases de datos distribuidas puede conducir a enormes costos de comunicación y a pobre rendimiento.**
  - ✓ **El diseño de bases de datos distribuidas es difícil debido a la carencia de herramientas, el número de opciones y las relaciones entre opciones.**
  - ✓ **El procesamiento de transacción distribuida puede agregar considerables gastos, en especial para datos copiados.**
  - ✓ **Los datos distribuidos implican más preocupaciones de seguridad porque son muchos los sitios que pueden manejar datos. Cada sitio debe protegerse adecuadamente de accesos no autorizados.**
- **El procesamiento de bases distribuidas proporcionan autonomía de sitio mientras que las bases de datos paralelas, no. Por lo tanto la autonomía es la diferencia fundamental entre el procesamiento de bases de datos paralelas y distribuidas.**

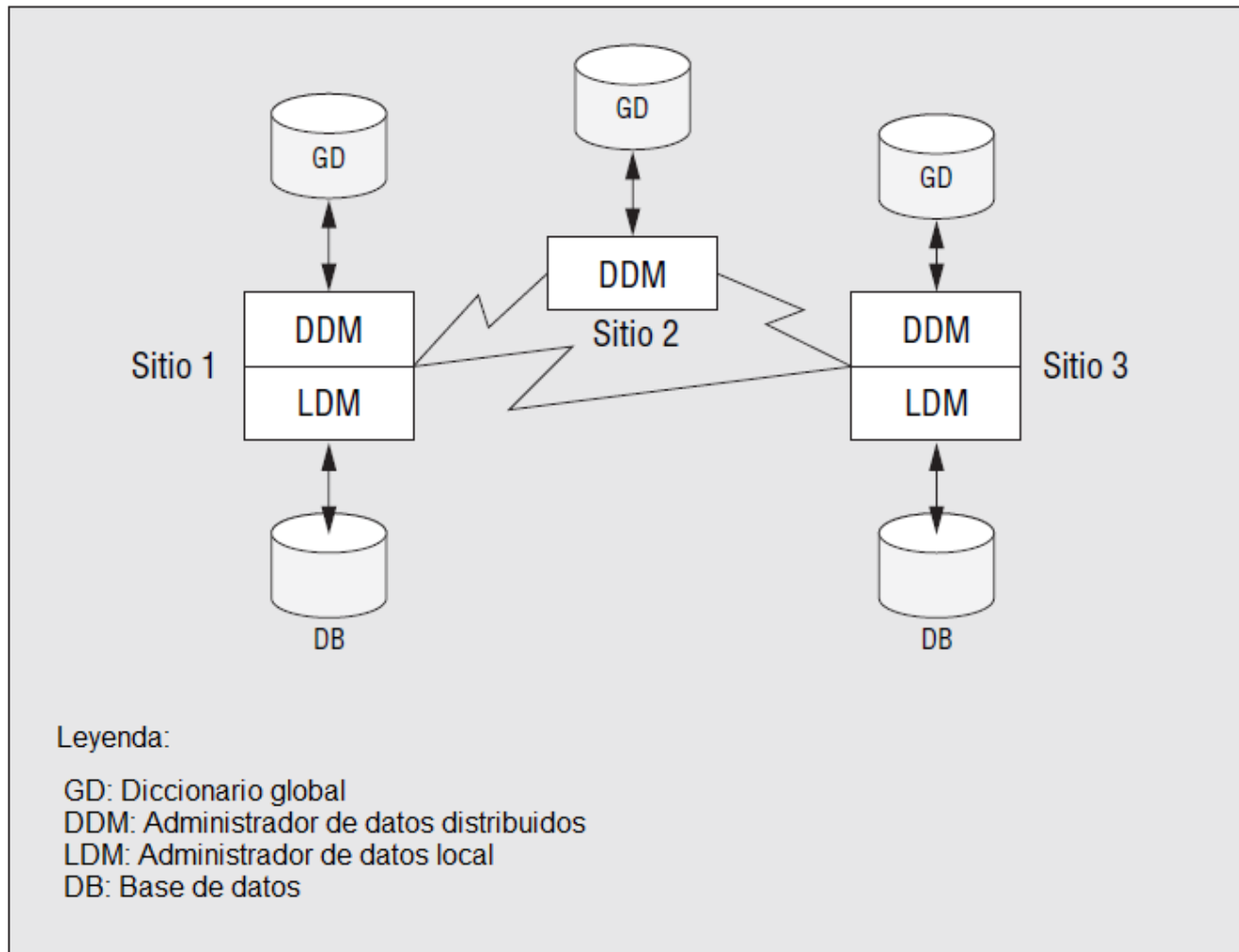
### ➤ *DISTRIBUCION*

- Los DBMS distribuidos soportan solicitudes globales que usan datos almacenados en más de un sitio autónomo, un sitio es cualquier computadora controlada localmente con una dirección de red única y con frecuencia se encuentran distribuidos geográficamente, aunque la definición soporta sitios ubicados en proximidad cercana. Por lo tanto, cada servidor con acceso a la base de datos distribuida se conoce como sitio
- Las solicitudes globales son consultas que combinan datos desde más de un sitio y transacciones que actualizan datos en más de un sitio, pueden implicar una colección de enunciados que acceden a datos locales en algunos enunciados y a datos remotos en otros.
- Los datos locales están controlados por el sitio en el que un usuario normalmente se conecta, los datos remotos implican un sitio diferente, en el que un usuario puede acceder sin tener cuenta de acceso. Si todas las solicitudes requieren datos de un solo sitio, no se requiere capacidades de procesamiento de bases de datos distribuidas
- Las bases de datos distribuidas son potencialmente útiles para organizaciones que operan en múltiples ubicaciones con control local de recursos computacionales



## ➤ *DISTRIBUCION*

- Posible ordenamiento de los componentes de un DBMS distribuido



### ➤ **DISTRIBUCION**

- En el ejemplo:
  - ✓ Cada servidor con acceso a la base de datos distribuida se conoce como sitio.
  - ✓ Si un sitio contiene una base de datos, ésta se controla con un administrador de datos local (LDM). Por lo tanto, los administradores de datos locales proporcionan características completas de un DBMS
  - ✓ El administrador de datos distribuidos (DDM) optimiza la ejecución de consultas a través de sitios, coordina el control de concurrencia y la recuperación a través de sitios, también controla el acceso a datos remotos. Al realizar estas tareas, el DDM utiliza el diccionario global (GD) para localizar partes de la base de datos.
  - ✓ El GD puede distribuirse en varios sitios de manera similar a como se distribuyen los datos

### ➤ DISTRIBUCION

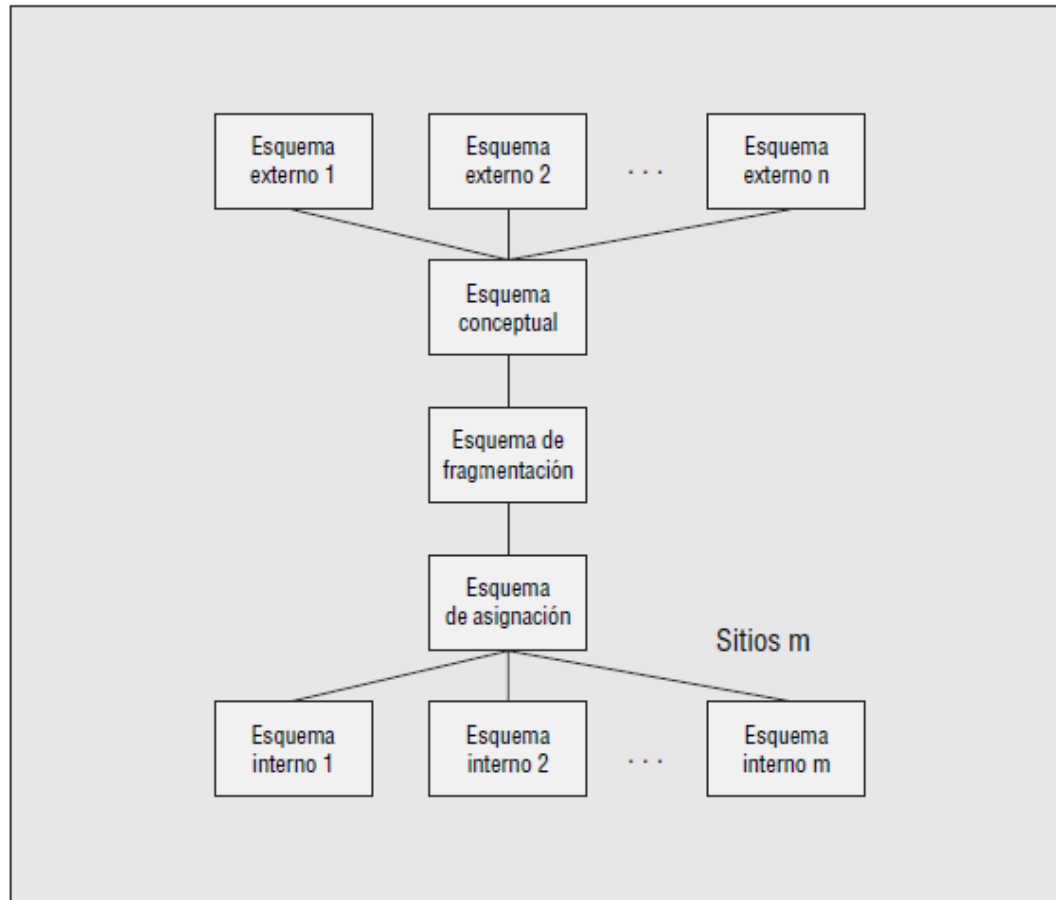
- En la arquitectura de componente, los administradores de base de datos locales (LDM) pueden ser homogéneos o heterogéneos.
  - ✓ **LDM Homogéneos:** Un DBMS distribuido con DBMS local homogéneo está *firmemente integrado*. El administrador de base de datos distribuido puede solicitar componentes internos y acceso al estado interno de los administradores de datos locales. La firme integración permite que el DBMS distribuido soporte eficientemente las consultas y transacciones distribuidas. Sin embargo, el requisito de homogeneidad prohíbe la integración de bases de datos existentes.
  - ✓ **LDM Heterogéneos:** Un DBMS distribuido con administradores de datos locales heterogéneos tiene *integración floja*. El administrador de base de datos distribuido actúa como middleware para coordinar administradores de datos locales. Con frecuencia, SQL proporciona la interfaz entre el administrador de datos distribuido y los administradores de datos locales. La integración floja soporta compartir datos entre sistemas heredados y organizaciones independientes. Sin embargo, el enfoque de integración floja puede no ser capaz de soportar procesamiento de transacciones de forma confiable y eficiente.
- Para acomodar la distribución de datos son necesarias capas adicionales de descripción de datos. Las arquitecturas proporcionan una referencia acerca de los tipos de descripción de datos necesarios y la forma de compartir la descripción de datos.

### ➤ **DISTRIBUCION**

- **DBMS distribuido firmemente integrado:**
  - ✓ La arquitectura de esquema contiene capas adicionales para fragmentación y asignación
  - ✓ El esquema de fragmentación contiene la definición de cada fragmento, mientras que el esquema de asignación contiene la ubicación de cada fragmento.
  - ✓ Un fragmento puede definirse como un subconjunto vertical (operación de proyecto), un subconjunto horizontal (operación de restricción) o un fragmento mezclado (combinación de operaciones de proyecto y de restricción).
  - ✓ Un fragmento se asigna a un sitio, aunque en ocasiones a múltiples sitios.
  - ✓ Si el DBMS distribuido soporta copiado, puede asignarse un fragmento a múltiples sitios, una copia de un fragmento se considera la copia primaria y las otras copias son secundarias. Sólo se garantiza que la copia primaria sea la actual.

### ➤ *DISTRIBUCION*

- Ejemplo gráfico de un DBMS distribuido firmemente integrado



### ➤ *DISTRIBUCION*

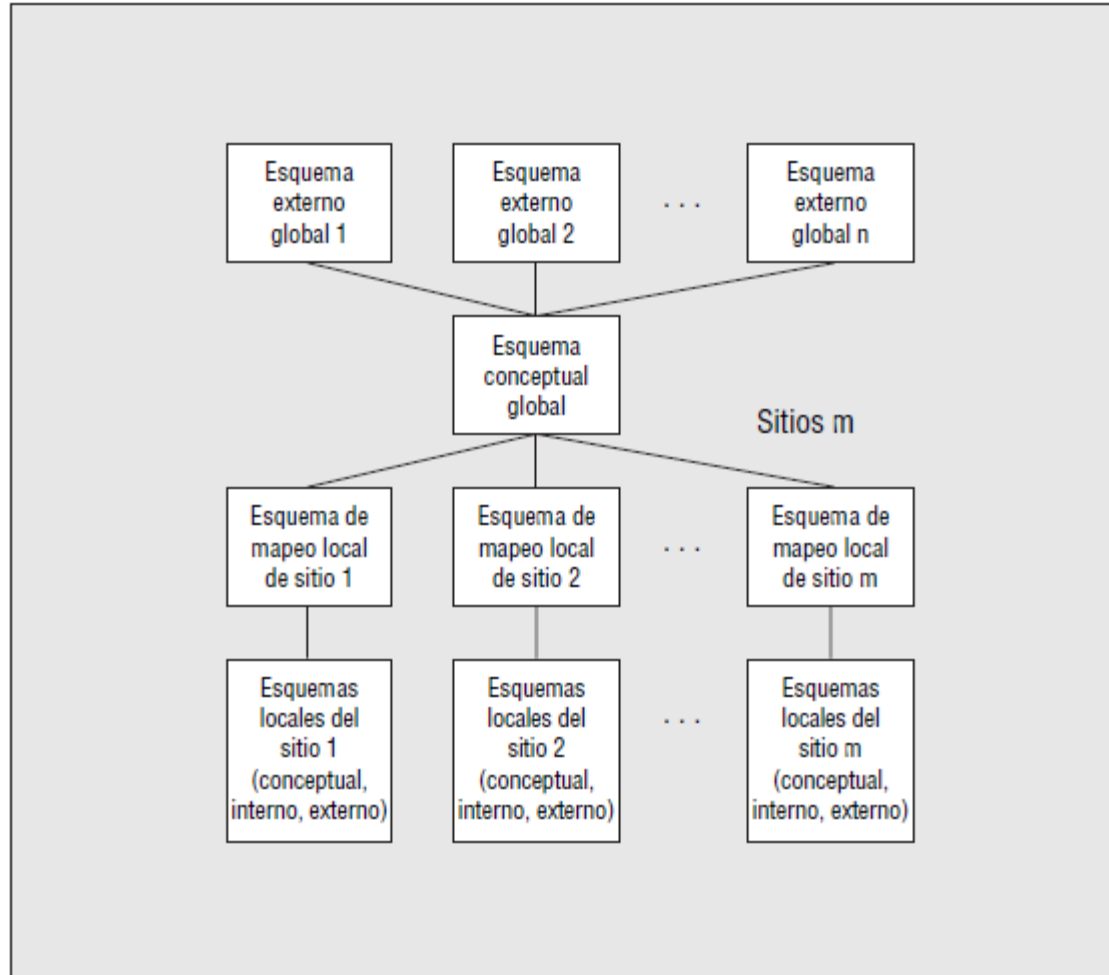
- **DBMS distribuido con integración floja:**
  - ✓ **La arquitectura de esquema soporta más autonomía de sitios de base de datos locales, además de compartición de datos.**
  - ✓ **Cada sitio contiene los tres niveles de esquema tradicionales.**
  - ✓ **Para soportar la compartición de datos, el DBMS distribuido proporciona un esquema de mapeo local para cada sitio.**
  - ✓ **Los esquemas de mapeo local describen los datos exportables en un sitio y proporcionan reglas de conversión para traducir los datos de un formato local a uno global.**
  - ✓ **El esquema conceptual global muestra todos los tipos de datos y relaciones que se pueden usar en solicitudes globales.**
  - ✓ **Los DBMSs distribuidos que no tienen un esquema conceptual global utilizan esquemas globales externos y proporcionan visiones de datos compartidos en un formato común.**
  - ✓ **Los formatos de datos locales pueden ser muy diferentes, los sitios locales pueden usar diferentes DBMSs, cada uno con un conjunto diferente de tipos de datos.**

### ➤ *DISTRIBUCION*

- **DBMS distribuido con integración floja:**
  - ✓ Los modelos de datos de los DBMS locales pueden ser diferentes, especialmente si se integran sistemas heredados.
  - ✓ Los sistemas heredados pueden usar interfaces de archivo y modelos de datos navegacionales (red y jerarquía) que no soportan SQL, como también si los sitios locales soportan un estándar SQL común, tales como diferentes tipos de datos, escalas, unidades de medida y códigos.
  - ✓ Los esquemas de mapeo local resuelven estas diferencias al proporcionar reglas de conversión que transforman los datos de un formato local a un formato global.

## ➤ DISTRIBUCION

- Ejemplo gráfico de un DBMS distribuido con integración floja





### ➤ *DISTRIBUCION*

- Las arquitecturas firmemente integradas y con integración floja representan dos posibilidades extremas, por ello entre estas dos arquitecturas se ha propuesto e implementado muchas variaciones:
  - ✓ Para proporcionar autonomía local adicional con mayor eficiencia para solicitudes globales, un sistema con integración floja puede requerir que todos los sitios locales soporten una interfaz SQL común.
  - ✓ Combinar los enfoques firmemente integrado y con integración floja.
  - ✓ Las redes de bases de datos distribuidas firmemente integradas pueden tener integración floja para compartir datos selectivos en solicitudes globales. Así, el DBMS distribuido con integración floja actúa como puerta de enlace entre las bases de datos distribuidas firmemente integradas

### ➤ **DISTRIBUCION**

- La transparencia en el procesamiento de bases de datos distribuidas se relacionan con la independencia de datos, de esta manera los usuarios pueden escribir consultas desconociendo la distribución. Cualquier cambio en la distribución de los datos no impacta en las consultas y tampoco en las transacciones
- La transparencia de fragmentación proporciona el mayor nivel de independencia de datos. Los usuarios formulan consultas y transacciones sin conocimiento de los fragmentos, ubicaciones o formatos locales. Si los fragmentos, ubicaciones o formatos locales cambian, las consultas y transacciones no se afectan
- La transparencia de ubicación proporciona un menor nivel de independencia de datos que la transparencia de fragmentación. Los usuarios necesitan referirse a fragmentos al formular consultas y transacciones. Sin embargo, no es necesario el conocimiento de las ubicaciones y formatos locales.
- La transparencia de mapeo local proporciona un menor nivel de independencia de datos que la transparencia de ubicación. Los usuarios necesitan hacer referencia a fragmentos en sitios al formular consultas y transacciones. Sin embargo, no es necesario el conocimiento de los formatos locales. Si los sitios difieren en formatos, como en las bases de datos distribuidas con integración floja, la transparencia de mapeo local todavía libera al usuario de un trabajo considerable.

### ➤ **DISTRIBUCION**

- **La transparencia en el procesamiento de bases de datos distribuidas se relacionan con la independencia de datos, de esta manera los usuarios pueden escribir consultas desconociendo la distribución. Cualquier cambio en la distribución de los datos no impacta en las consultas y tampoco en las transacciones**
- **Los datos distribuidos pueden agregar complejidad para la formulación de consultas, añaden considerable complejidad al procesamiento de consultas y transacciones.**
- **El procesamiento de bases de datos distribuidas implica movimiento de datos, procesamiento remoto y coordinación del sitio que están ausentes del procesamiento de bases de datos centralizadas.**
- **Aunque los detalles del procesamiento de bases de datos distribuidas pueden ocultarse a los programadores y usuarios, a veces no es posible ocultar las implicaciones de rendimiento.**

## ***Sistema Gestor de Bases de Datos (SGBD)***

INDEPENDIENTEMENTE DE LA ESTRUCTURA

***¿QUE CARACTERITICAS DEBEN TENER LAS BASES DE DATOS?***

- PERFORMANCE
- TRANSPARENCIA
- ESCALABILIDAD
- DISPONIBILIDAD / INTEGRIDAD (CONFIABILIDAD)

### ➤ **PERFORMANCE**

- **Capacidad para ejecutar operaciones en paralelo**
- **Multi-threading**
- **Tener paralelismo de Input/Output**
- **Capacidad para ejecutar operaciones utilitarias en paralelo (Backups/Restores) todo en línea**
- **Paralelismo para el soporte de aplicaciones:**
  - ✓ **Paralelismo de I-O de disco**
  - ✓ **Paralelismo de utilitarios**
  - ✓ **Paralelismo en consultas y procesamiento**
- **El particionamiento de los datos debe ser independiente de la estructura de Hardware.**
- **Paralelismo de I-O de disco:**
  - ✓ **Posibilidad trabajar con tablas particionadas**
  - ✓ **El motor de BD debe permitir la ejecución concurrente de I/O**

### ➤ **PERFORMANCE**

- **Paralelismo de utilitarios:**

- ✓ **Indices**
- ✓ **Backup**
- ✓ **Restore**

- **Paralelismo en consultas y procesamiento:**

- ✓ **Es el más complejo de implementar debido a que una consulta o una modificación de los datos, involucra varias operaciones atómicas y resulta complicado hacerlas en paralelo y coordinarlas.**
- ✓ **El SGBD debe seleccionar operaciones atómicas las cuales pueden ser replicadas y luego ser procesadas en forma concurrente. Luego el resultado se combina como una única operación.**
- ✓ ***Paralelismo Horizontal:* es la forma de crear múltiples copias de una operación que corre en paralelo.**
- ✓ ***Otra forma de paralelismo:* mientras una operación clasifica registros según el valor de un campo clave, en forma concurrente existe una operación que accede a los datos y no espera que estos sean procesados, dejándolos listos para su procesamiento**

### ➤ **TRANSPARENCIA:**

- Para los DBMS, transparencia significa que los detalles internos de los servicios de las transacciones son invisibles, mejorando así la productividad de los programadores y analistas de bases de datos
- Los DBMS ofrecen dos servicios para garantizar que las transacciones tienen las propiedades ACID:
  - ✓ **Transparencia de recuperación:** comprende acciones para manejar las fallas, como errores en la comunicación y colapsos del software. El DBMS restaura automáticamente una base de datos a un estado consistente después de una falla
  - ✓ **Transparencia de concurrencia:** comprende acciones para controlar la interferencia entre varios usuarios simultáneos de la base de datos. Los usuarios perciben la base de datos como un sistema para un solo usuario, aunque haya varios usuarios simultáneos
- En el procesamiento de bases de datos distribuidas, la transparencia se relaciona con la independencia de datos. Si la distribución de la base de datos es transparente, los usuarios pueden escribir consultas sin conocimiento de su distribución, esto se conoce como transparencia de fragmentación.
- El procesamiento de bases de datos paralelas en las arquitecturas nada compartido (SN) involucra transparencia de fragmentación.

### ➤ **ESCALABILIDAD:**

- Propiedad que tiene cualquier sistema, para crecer dinámicamente, en base a sus necesidades, utilizando los recursos disponibles (Hardware), sin operaciones complementarias, al incrementar la cantidad de usuarios, procesos, etc.
- La BD debe adaptarse sin inconvenientes a mayor storage.
- Si se agrega más hardware a los servidores que soportan la BD éste debe poder utilizarlo automáticamente y solapar procesos o utilizar más memoria.
- La escalabilidad se analiza desde transacciones de lectura
- Dos tipos de escalabilidad:
  - ✓ Escalabilidad Vertical: Se escala sobre el mismo server, con una CPU más rápida o agregando CPU's, agregando una controladora más rápida, agregando más memoria.
  - ✓ Escalabilidad Horizontal: Se utilizan múltiples servers que trabajan en forma transparente distribuyendo la carga de trabajo.



### ***ESCALABILIDAD:***

- La escalabilidad permite:
  - ✓ Adaptarse rápidamente a los cambios en las organizaciones, producto de reducir a un mínimo los tiempos de planificación
  - ✓ Adaptarse a los cambios tecnológicos.

# ***INTEGRIDAD / DISPONIBILIDAD***

- Las BD deben dar **CONFIABILIDAD/ROBUSTEZ/INTEGRIDAD**, por intermedio del Software y de Hardware
- Dos factores a analizar:
  - ✓ **Administración ON-Line:** Utilitarios de operación continua que permitan realizar tareas on-line.
  - ✓ **Robustez:**
    - Reducir la posibilidad de fallas y realizar recuperos en forma transparente.
    - **Fault tolerant:** se llaman así a los sistemas con alta disponibilidad, implementando redundancia de hardware (mirroring de disco) y redundancia de datos (soft mirroring / replicación).

# ***INTEGRIDAD / DISPONIBILIDAD***

- **Redundancia datos**
  - **Soft mirroring (Clusters):**
    - ✓ **Duplicar unidades de datos en discos distintos.**
    - ✓ **Las lecturas se pueden distribuir**
    - ✓ **Las escrituras son transparentes**
    - ✓ **Ante una falla las operaciones se realizan sobre el medio no dañado.**
    - ✓ **Para los usuarios es totalmente transparente**
    - ✓ **En el medio dañado una vez reparado se deben sincronizar las copias de los dispositivos**
  - **Replicación:**
    - ✓ **Similar al mirroring, pero se puede localizar en forma remota.**
    - ✓ **Se trata de copiar los datos en diferentes lugares, puede ser en un solo equipo o en varios discos,**
    - ✓ **Esta forma de resguardo no asegura alta disponibilidad, intenta ganar en performance**

# **CONFIABILIDAD Y DISPONIBILIDAD**

- **CONFIABILIDAD:** Un SGBD es confiable si bajo las condiciones de uso determinadas por sus especificaciones cumple con su misión específica en un período determinado. Se utiliza sobre sistemas críticos. Se utiliza un parámetro crítico para evaluar una BD sobre la base de las probabilidades de ocurrencias de fallas. (MTBF Mean-Time-Between-Failures).
- **DISPONIBILIDAD:** Porción de tiempo que la BD se encuentra apta para cumplir su misión, respecto al tiempo que debió haber cumplido su misión y no lo hizo. Para medir la disponibilidad se utiliza el parámetro MTTR (Tiempo Medio de Reparaciones) y el MTBF de la siguiente forma:

$$\text{DISPONIBILIDAD} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

# CONFIABILIDAD Y DISPONIBILIDAD

- Para diseñar un SGBD distribuido confiable es necesario definir los tipos de falla que pueden ocurrir:
- ✓ **FALLA DE TRANSACCIONES:** No se cumple con el modelo **ACID** (**A**tomicidad – **C**oncurrencia – **I**slamamiento - **D**urabilidad).
- ✓ **FALLAS DE SISTEMA:** Son las fallas propias del hardware (procesador, memoria, etc.) donde opera el SGBD.
- ✓ **FALLAS DEL MEDIO DE ALMACENAMIENTO:** Es el lugar físico donde se almacenan los datos.
- ✓ **FALLAS DE COMUNICACIÓN:** Son fallas en el medio de comunicación propiamente dicho que producen pérdida de mensajes

## ***MODOS EN QUE TRABAJA UNA BASE DE DATOS:***

- ***OLTP: Procesamiento de Transacciones en Línea***
- ***DSS: Sistemas de Soporte de Decisión ( o Sistemas de Asistencia de Decisiones), también llamados OLAP***
- ***OLCP: Procesamiento Complejo en Línea***

# ***MODOS EN QUE TRABAJA UNA BASE DE DATOS:***

## **➤ *OLTP: Procesamiento de Transacciones en Línea***

- El procesamiento de transacciones comprende el lado operativo de las bases de datos.
- El motor de la base de datos está optimizado para trabajar con un conjunto de transacciones en paralelo y cortas, en otras palabras se envían al servidor transacciones pequeñas de corta duración.
- Normalmente realizan lectura de porciones de información pequeñas.
- En este modo de trabajo se generan muchas operaciones de inserción y/o modificación
- Los tiempos de ejecución de las transacciones son cortos
- La ejecución de reportes, toma estadísticas degradan la performance de la BD, volviendo lentas todas las operaciones.
- La ejecución de grandes reportes y copias de respaldo suelen realizarse en momentos de poco uso de la BD.

### ***MODOS EN QUE TRABAJA UNA BASE DE DATOS:***

- ***DSS: Sistemas de Soporte de Decisión ( o Sistemas de Asistencia de Decisiones), también llamados OLAP***
- El motor de la base de datos está optimizado para realizar lecturas de una gran cantidad de datos, sumariazación de datos, es decir soporta procesos largos que van a necesitar leer muchos datos para la generación de reportes, análisis estadísticos, etc.
- Generalmente realizan lectura de porciones de datos muy grandes.
- En este modo de trabajo es casi nula la generación de operaciones de inserción y/o modificación.
- Los tiempos de ejecución de las transacciones suelen ser importantes en cuanto al tiempo.
- Suelen utilizarse para la ejecución de reportes, toma estadísticas.
- Para este modo de trabajo suelen utilizarse imágenes relativamente estáticas de la base de datos OLTP para asegurar la integridad de los datos mientras se ejecutan los procesos largos.



# ***MODOS EN QUE TRABAJA UNA BASE DE DATOS:***

## ***➤ OLCP: Procesamiento Complejo en Línea***

- Es un intermedio “o un mix” entre el modo OLTP y DSS, intentando alcanzar un balance entre los dos modos, reflejando la necesidad de obtener:
  - ✓ Alto rendimiento de la base (muy buena performance)
  - ✓ Completa disponibilidad de los datos
  - ✓ Capacidad de realizar copias de respaldo en línea (backup en caliente)
  - ✓ Ejecución de grandes consultas e informes mientras los usuarios actualizan la información diaria de la organización
  - ✓ Información disponible en todo momento, sin límite de acceso a los usuarios OLTP y DSS.
  - ✓ Destinada a aquellas organizaciones que necesitan obtener informes críticos en tiempo real