

MÓDULO X

MONITOREO Y AFINAMIENTO DE BASE DE DATOS

MONITOREO SOBRE EL COMPORTAMIENTO DE LOS SGBD Y LOS SISTEMAS OPERATIVOS.

GENERALIDADES DE AFINAMIENTO AL SGBD Y LOS SISTEMAS OPERATIVOS

Marcela Russo
Laboratorio IV

MONITOREO Y AFINAMIENTO

- Si bien el componente de optimización de consultas lleva a cabo su función de forma automática, el administrador de bases de datos (DBA) tiene la responsabilidad de monitorear el comportamiento de la base de datos y aplicar las acciones necesarios para optimizar su rendimiento.
- El DBA debe revisar los planes de acceso de consultas y actualizaciones que se desempeñen mal.
- Los DBMS corporativos proporcionan despliegues gráficos de los planes de acceso para facilitar la revisión. Para mejorar las malas decisiones de los planes de accesos, algunos DBMS corporativos permiten incluir pistas que influyan en la selección de los planes de acceso.
- Algunos motores de base de datos proporcionan sugerencias para seleccionar la meta de optimización, las estructuras de archivos para acceder a las tablas individuales, el algoritmo y el orden del JOIN. Las sugerencias deben usarse con precaución ya que pueden invalidar la decisión del optimizador.
- Las propuestas acerca de los algoritmos de JOIN y el orden de los JOIN son especialmente problemáticos debido a lo sutil de estas decisiones. Anular las decisiones del optimizador sólo debe hacerse como último recurso después de determinar la causa del mal desempeño.
- El DBA puede reparar problemas con deficiencias en el perfil de tablas y en el estilo de codificación de consultas para mejorar el desempeño en lugar de anular la decisión del optimizador.

MONITOREO Y AFINAMIENTO

- Deficiencias del perfil de tablas:
 - ✓ El componente de optimización de consultas necesita estadísticas detalladas y actualizadas para evaluar los planes de acceso.
 - ✓ La mayoría de los DBMS proporciona control sobre el nivel de detalle de las estadísticas y su actualización.
 - ✓ Algunos DBMS permiten el muestreo dinámico de la base de datos durante el tiempo de optimización, pero normalmente este nivel de actualización de datos no es necesario.
 - ✓ Si las estadísticas no se recopilan para una columna, la mayoría de los DBMS utilizan el valor uniforme que se asume para estimar el número de filas.
 - ✓ El uso del valor uniforme que se asume generalmente lleva a un acceso secuencial de archivos en lugar de un acceso con un Btree, esto en el caso de que la columna tenga suficientes inclinaciones en sus valores.

MONITOREO Y AFINAMIENTO

- Deficiencias del perfil de tablas:

- ✓ Por ejemplo, en una consulta para listar a los empleados con salarios mayores a \$100 000:
 - Si el rango del salario va de \$10 000 a \$2 000 000, cerca del 95 % de la tabla de empleados deberá satisfacer esta condición utilizando el valor uniforme que se asume.
 - Es probable que para la mayoría de las empresas, pocos empleados tengan un salario mayor a \$100 000.
 - Por lo tanto, utilizando la estimación del valor uniforme que se asume, el optimizador escogerá un archivo secuencial en vez de un Btree para acceder a la tabla de empleados.
 - Como resultado, la estimación no mejorará mucho utilizando un histograma de igual intervalo debido a la inclinación extrema de los valores de los salarios.
 - Un histograma de la misma altura proporcionará mucho mejores estimaciones.
 - Para mejorar el cálculo utilizando un histograma de la misma altura, debe aumentar el número de rangos.
 - Por ejemplo, con 10 rangos el error máximo es de alrededor de 10 % y el error esperado es de cerca de 5%.

MONITOREO Y AFINAMIENTO

- Deficiencias del perfil de tablas:
 - Para disminuir las estimaciones de los errores máximo y esperado en 50%, se debe duplicar el número de rangos.
 - Un DBA debe aumentar el número de rangos si los errores estimados para el número de filas ocasionan selecciones inadecuadas para acceder a las tablas individuales.
 - Una sugerencia puede ser útil para condiciones que incluyan valores de parámetros.
 - Si el DBA sabe que los típicos valores de parámetros generan un conjunto de pocas filas, se puede utilizar una sugerencia para obligar al componente de optimización a que use un índice.

MONITOREO Y AFINAMIENTO

- Un histograma de la misma altura proporcionará mucho mejores estimaciones.
- Para mejorar el cálculo utilizando un histograma de la misma altura, debe aumentar el número de rangos.
- Por ejemplo, con 10 rangos el error máximo es de alrededor de 10 % y el error esperado es de cerca de 5%. Para disminuir las estimaciones de los errores máximo y esperado en 50%, se debe duplicar el número de rangos.
- Un DBA debe aumentar el número de rangos si los errores estimados para el número de filas ocasionan selecciones inadecuadas para acceder a las tablas individuales.
- Una sugerencia puede ser útil para condiciones que incluyan valores de parámetros.
- Si el DBA sabe que los típicos valores de parámetros generan un conjunto de pocas filas, se puede utilizar una sugerencia para obligar al componente de optimización a
- que use un índice.
- un componente de optimización, en ocasiones necesita estadísticas detalladas de las combinaciones de columnas.
- Si una combinación de columnas aparece en la cláusula WHERE de una consulta, las estadísticas de la combinación de columnas son importantes cuando las columnas no son independientes.

MONITOREO Y AFINAMIENTO

- Por ejemplo, los salarios de los empleados y sus puestos están generalmente relacionados.
- Una cláusula WHERE con las dos columnas probablemente tenga pocas filas que satisfagan ambas condiciones. Un componente de optimización que desconozca la relación entre estas columnas probablemente sobreestimaré el número de filas del resultado.
- La mayoría de los componentes de optimización asumen que las combinaciones de columnas son estadísticamente independientes para simplificar la estimación del número de filas.
- Son pocos los DBMS que conservan estadísticas de las combinaciones de columnas.
- Si un DBMS no conserva las estadísticas de las combinaciones de columnas, el diseñador de bases de datos pudiera utilizar las sugerencias para anular la selección del DBMS cuando una condición de JOIN en la cláusula WHERE genere pocas filas.
- El uso de una sugerencia puede obligar al componente de optimización a combinar índices cuando acceda a una tabla en vez de usar un escaneo secuencial de la tabla.

MONITOREO Y AFINAMIENTO

- **Prácticas en la codificación de consultas:**

- ✓ **Las consultas mal escritas pueden ocasionar una ejecución lenta de las mismas.**
- ✓ **El DBA debe revisar las consultas que se desempeñen mal buscando las prácticas de codificación que ocasionen su lento desempeño.**
- ✓ **Consideraciones a tener en cuenta en la codificación de consultas:**
 - **No utilizar funciones en columnas que tengan índices, ya que anulan la utilización del índice, incluso las conversiones de tipo implícito eliminan la posibilidad de usar un índice, estas ocurren cuando no coinciden el tipo de dato de una columna y el valor constante asociado.**
 - **La velocidad de ejecución de una consulta se determina principalmente por el número de operaciones de JOIN, por lo que la eliminación de operaciones de JOIN innecesarias puede disminuir significativamente el tiempo de ejecución.**
 - **Para consultas que involucren relaciones 1-M en las que exista una condición en la columna de JOIN, la condición debe estar en la tabla madre en lugar de la tabla hija, esto reduce el esfuerzo al enlazar las tablas.**
 - **Para consultas que involucren la cláusula HAVING, eliminar las condiciones que no involucren funciones agregadas. Las condiciones que involucren comparaciones sencillas de las columnas de la cláusula GROUP BY pertenecen a la cláusula WHERE, no a la cláusula HAVING, esto eliminará las filas más pronto y la ejecución será más rápida**

MONITOREO Y AFINAMIENTO

- **Prácticas en la codificación de consultas:**

- ✓ **Consideraciones a tener en cuenta en la codificación de consultas:**

- **Evitar las consultas anidadas de tipo II (consulta anidada en la que la consulta interna hace referencia a alguna tabla usada en la consulta anidada externa), en especial cuando la consulta anidada lleve a cabo la agrupación con cálculos agregados. Muchos DBMS son lentos porque los componentes de optimización de consultas generalmente no consideran formas eficientes para implementar las consultas anidadas del tipo II. Se puede mejorar la velocidad de ejecución de una consulta reemplazando la consulta anidada del tipo II con una consulta separada.**
- **Las consultas con vistas complejas pueden conducir a un lento desempeño debido a que se puede ejecutar una consulta adicional.**
- **El proceso de optimización puede consumir tiempo, en especial para las consultas que contienen más de cuatro tablas. Para reducir el tiempo de optimización, la mayoría de los DBMS guardan planes de acceso para evitar las fases que consumen tiempo del proceso de traducción.**
- **La ligadura de consultas es el proceso de asociar una consulta con un plan de acceso.**
- **La mayoría de los DBMS hacen las ligaduras de forma automática si una consulta cambia o si lo hace la base de datos (estructuras de archivos, perfiles de tablas, tipos de datos, etc.).**

MONITOREO Y AFINAMIENTO

Resumen de las prácticas de codificación

Práctica de codificación	Recomendación	Elemento de desempeño
Funciones en las columnas en condiciones	Evite las funciones en columnas	Elimina la posibilidad del uso de un índice
Conversiones de tipo implícito	Use constantes con tipos de datos que coincidan con las columnas correspondientes	Elimina la posibilidad del uso de un índice
Operaciones de enlace adicionales	Elimine operaciones de enlace innecesarias buscando las tablas que no involucren condiciones o columnas	El tiempo de ejecución se determina principalmente por el número de operaciones de enlace
Condiciones en columnas de enlace	Las condiciones sobre las columnas de enlace deben usar la tabla madre y no la tabla hija	La reducción del número de filas de la tabla madre disminuirá el tiempo de ejecución de las operaciones de enlace
Condiciones de las filas en la cláusula HAVING	Mueva las condiciones de las filas de la cláusula HAVING a la cláusula WHERE	Las condiciones sobre las filas en la cláusula WHERE permiten la reducción en el tamaño del resultado intermedio
Consultas anidadas del tipo II con agrupación	Convierta las consultas anidadas del tipo II en consultas separadas	Los componentes de optimización de consultas generalmente no consideran maneras eficientes para implementar las consultas anidadas del tipo II
Consultas usando vistas complejas	Reescriba las consultas que usan vistas complejas para eliminar las referencias a vistas	Se puede ejecutar una consulta adicional
Reutilización de consultas	Asegúrese de que las consultas de un procedimiento almacenado se usen sólo una vez	El uso repetitivo involucra una sobrecarga considerable para las consultas complejas

MONITOREO Y AFINAMIENTO

- Selección de Índices:

- ✓ **Índice:** Estructura secundaria de archivos que proporciona una ruta alternativa hacia los datos. En un índice agrupado, el orden de los registros de datos es cercano al orden del índice. En un índice desagrupado, el orden de los registros de los datos no está relacionado al orden del índice.
- ✓ **La selección de índices es la decisión más importante y muy difícil, para el diseñador de la base de datos física. La selección incluye dos tipos de índices: agrupados y desagrupados.**
 - **Índices agrupados:** El orden de las filas es cercano al orden del índice. Cercano significa que los registros físicos que contienen las filas no serán consultados más de una vez si el índice se consulta de forma secuencial. La forma más sencilla de hacer un índice agrupado es ordenando los datos de las filas mediante la columna índice.
 - **Índices desagrupados:** No tiene la propiedad de cercanía. El orden de las filas no está relacionado con el orden del índice. El mismo registro físico puede ser consultado de forma repetitiva cuando se usa un conjunto de secuencias. Los punteros los nodos del conjunto de secuencias hacia las filas se cruzan muchas veces, indicando que el orden del índice es distinto al orden de las filas.
- ✓ **Generalmente, el problema de selección de índices está restringido a índices Btree y a archivos separados para cada tabla. Los tipos de índices hash y la ubicación de datos de varias tablas en el mismo archivo también pueden influir si no se agregan mejoras al desempeño.**

MONITOREO Y AFINAMIENTO

- **Selección de Índices:**

- ✓ **La mejor selección de índices balancea una recuperación rápida con actualizaciones más lentas.**
- ✓ **Un índice desagrupado puede mejorar las recuperaciones al proporcionar un acceso rápido a los registros seleccionados.**
- ✓ **Generalmente, menos del 5% de las filas deben satisfacer una condición para que el índice desagrupado sea útil.**
- ✓ **Para que los optimizadores soporten accesos a índices múltiples para la misma tabla, los índices desagrupados pueden ser útiles incluso cuando un índice sencillo no proporcione por sí mismo suficiente selección de filas.**
- ✓ **La habilidad de usar múltiples índices sobre la misma tabla aumenta la utilidad de los índices desagrupados.**
- ✓ **Un índice desagrupado también puede ser útil en un JOIN cuando una de las tablas del enlace tenga un número pequeño de filas en el resultado.**

MONITOREO Y AFINAMIENTO

- **Selección de Índices:**

- ✓ **Un índice agrupado puede mejorar las recuperaciones en más situaciones que un índice desagrupado.**
- ✓ **Un índice agrupado es útil en las mismas situaciones que un índice desagrupado a excepción de que el número de filas resultantes puede ser mayor.**
- ✓ **Un índice agrupado puede ser útil:**
 - **Si tal vez 20% de las filas satisfacen la condición asociada de la consulta.**
 - **En los enlaces ya que evita la necesidad de ordenar, se pueden enlazar mezclando las filas de cada tabla.**
 - **La mezcla de filas generalmente es una forma más rápida para enlazar tablas cuando éstas no necesiten estar ordenadas.**
- ✓ **El costo de conservar índices como resultado de las sentencias INSERT, UPDATE y DELETE balancea las mejoras de recuperación. Las sentencias INSERT y DELETE afectan a todos los índices de una tabla.**
- ✓ **La preferencia es que la tabla no tenga muchos índices cuando tiene operaciones frecuentes de inserción y eliminación.**

MONITOREO Y AFINAMIENTO

- **Selección de Índices:**
 - ✓ Las sentencias UPDATE afectan sólo a las columnas enlistadas en la cláusula SET. Si las sentencias UPDATE hechas sobre una columna son frecuentes, se pierde el beneficio del índice.
 - ✓ Las alternativas de los índices agrupados son más sensibles al mantenimiento que las de los índices desagrupados.
 - ✓ Los índices agrupados son más costosos de mantener que los índices desagrupados porque el archivo de datos debe cambiar de forma similar a como lo hace un archivo secuencial ordenado.
 - ✓ Para los índices desagrupados, el archivo de datos se puede mantener como se hace con un archivo secuencial desordenado.

MONITOREO Y AFINAMIENTO

- **Dificultades en la selección de Índices:** La selección de índices por varias razones, por tal motivo, los DBMS corporativos y algunos otros fabricantes proporcionan herramientas asistidas por computadora para ayudar en la selección.
 - ✓ **Es difícil especificar los pesos de las aplicaciones.** Las decisiones que combinan la frecuencia e importancia pueden hacer que el resultado sea subjetivo.
 - ✓ **En ocasiones se necesita la distribución de los valores de los parámetros.** Muchas sentencias SQL usadas en los reportes y formularios usan valores en los parámetros. Si los valores de los parámetros varían desde ser muy selectivos hasta no serlo, la selección de índices es difícil.
 - ✓
 - ✓ **Se debe conocer el comportamiento del componente de optimización de consultas.** Incluso si un índice parece ser útil para alguna consulta, el componente de optimización de consultas debe usarlo. Puede haber razones sutiles para que el componente de optimización de consultas no use un índice, en especial un índice desagrupado.
 - ✓ **El número de alternativas es grande.** Incluso si los índices que están en las combinaciones de las columnas son ignorados, el número teórico de alternativas es exponencial con el número de columnas (2^{NC} en donde NC es el número de columnas). Aunque muchas de estas alternativas se pueden eliminar fácilmente, el número de alternativas prácticas todavía es muy grande.
 - ✓ **Las alternativas de los índices se pueden interrelacionar.** Las interrelaciones pueden ser sutiles, en especial cuando la selección de índices puede mejorar el desempeño de los JOIN.

MONITOREO Y AFINAMIENTO

- **Reglas de selección de índices:** para reducir la mala selección de índices se pueden aplicar las siguientes reglas:
 - ✓ **Regla 1:** Una clave primaria es un buen candidato para un índice agrupado.
 - ✓ **Regla 2:** Para respaldar los JOIN, considerar los índices sobre las claves foráneas. Un índice desagrupado sobre una clave foránea es una buena idea cuando existen consultas importantes con condiciones altamente selectivas hechas sobre una tabla relacionada. Un índice agrupado es una buena opción cuando la mayoría de los enlaces usan una tabla madre con un índice agrupado sobre su clave primaria, y las consultas no tienen condiciones altamente selectivas sobre la tabla madre.
 - ✓ **Regla 3:** Una columna con muchos valores puede ser una buena opción para un índice desagrupado cuando se usan en condiciones de igualdad. El término muchos valores significa que la columna es casi única.
 - ✓ **Regla 4:** Una columna que se usa en un rango de condiciones altamente selectivas es buena candidata para convertirse en un índice desagrupado.
 - ✓ **Regla 5:** Una combinación de columnas usada en forma conjunta con las condiciones de una consulta puede ser candidata para convertirse en índices desagrupados cuando las condiciones del enlace regresen pocas filas, el optimizador del DBMS soporte el acceso a varios índices y las columnas sean estables. Los índices individuales deben crearse para cada columna.

MONITOREO Y AFINAMIENTO

- **Reglas de selección de índices:** para reducir la mala selección de índices se pueden aplicar las siguientes reglas:
 - ✓ **Regla 6:** Una columna que se actualiza frecuentemente no es un buen candidato para un índice.
 - ✓ **Regla 7:** Las tablas volátiles (con muchas inserciones y eliminaciones) no deben tener muchos índices.
 - ✓ **Regla 8:** Las columnas estables con pocos valores son candidatas a convertirse en índices de tipo bitmap cuando las columnas se encuentren dentro de las condiciones WHERE.
 - ✓ **Regla 9:** Evitar los índices con combinaciones de columnas. La mayoría de los componentes de optimización pueden usar varios índices sobre la misma tabla. Un índice que se haga sobre una combinación de columnas no es tan flexible como varios índices que se hagan para las columnas individuales de la tabla.

MONITOREO Y AFINAMIENTO

- Otras opciones que colaboran con la mejora del desempeño de las bases de datos:
 - ✓ Desnormalización.
 - ✓ Formateo de registros.
 - ✓ Procesamiento en paralelo

MONITOREO Y AFINAMIENTO

- **Desnormalización**

- ✓ **Los diseños normalizados:**

- Tienen un mejor desempeño para las actualizaciones.
- Requieren menos código para obligar a que se cumplan las restricciones de integridad.
- Soportan más índices para mejorar el desempeño de las consultas.

- ✓ **La desnormalización:**

- Ignorar una dependencia funcional si no conduce a anomalías significativas de las modificaciones.
- Después de combinar las tablas, la nueva tabla puede violar alguna de las formas normales como el BCNF (forma normal de Boyce-Codd, por sus siglas en inglés).
- Si bien, algunas de las técnicas de desnormalización no conducen a violaciones de alguna forma normal, hacen que el diseño sea más fácil de consultar y más difícil de actualizar.
- La desnormalización siempre debe hacerse con mucho cuidado ya que un diseño normalizado tiene importantes ventajas.

MONITOREO Y AFINAMIENTO

- **Desnormalización**

- ✓ **Grupos de repetición**

- **Un grupo de repetición es un conjunto de valores asociados.**
- **Las reglas de normalización obligan a que los grupos de repetición se almacenen en una tabla hija separada de su tabla madre asociada.**
- **La desnormalización puede ser una alternativa factible si siempre se accede a un grupo repetido mediante su tabla madre asociada.**
- **Aunque el diseño desnormalizado no viola la forma normal BCNF, es menos flexible para las actualizaciones que el diseño normalizado.**
- **El diseño desnormalizado no requiere un JOIN para combinar los datos.**

MONITOREO Y AFINAMIENTO

- **Desnormalización**

- ✓ **Jerarquías de generalización**

- **La regla de conversión de jerarquías de generalización permite obtener muchas tablas.**
- **Si las consultas necesitan combinar regularmente estas tablas separadas, es factible almacenarlas como una sola tabla.**
- **Las tablas tienen relaciones 1-1, ya que representan una jerarquía de generalización.**
- **Aunque el diseño desnormalizado no viola la BCNF, la tabla combinada puede desperdiciar mucho espacio debido a los valores nulos; sin embargo, el diseño desnormalizado**
- **evita el uso del operador de JOIN externos (OUTER JOIN) para combinar las tablas.**

MONITOREO Y AFINAMIENTO

- **Desnormalización**

- ✓ **Códigos y significados**

- Las reglas de normalización requieren que las claves foráneas se almacenen en forma aislada para representar las relaciones 1-M.
- Si una clave foránea representa un código, generalmente el usuario necesita un nombre asociado o descripción, además del valor de la clave foránea.
- El almacenamiento de la columna del nombre o de la descripción junto con el código viola la BCNF, pero elimina algunas operaciones de JOIN.
- La desnormalización puede ser una opción razonable si la columna del nombre o de la descripción no cambia de forma constante.

MONITOREO Y AFINAMIENTO

- **Formateo de registros**
 - ✓ Las decisiones sobre el formateo de registros incluyen la compresión y datos derivados.
 - ✓ La compresión es un elemento importante con mayor énfasis en el almacenamiento de tipos de datos complejos, tales como audio, video e imágenes.
 - ✓ La compresión tiene sus ventajas y desventajas con respecto al esfuerzo del procesamiento de entradas-salidas.
 - ✓ La compresión reduce el número de registros físicos transferidos, pero puede requerir de un esfuerzo de procesamiento considerable para comprimir y descomprimir los datos.
 - ✓ Las decisiones sobre datos derivados involucran ventajas y desventajas entre las operaciones de consultas y actualizaciones.
 - ✓ Para efectos de consultas, el almacenamiento de datos derivados reduce la necesidad de recuperar datos requeridos para calcular los datos derivados.
 - ✓ Las actualizaciones para los datos utilizados en el cálculo requieren de actualizaciones adicionales de los datos derivados.
 - ✓ Puede ser razonable almacenar datos derivados para reducir las operaciones de JOIN.

MONITOREO Y AFINAMIENTO

- **Procesamiento paralelo**

- ✓ El desempeño se puede mejorar de forma significativa al efectuar operaciones de recuperación y modificación a través del procesamiento paralelo.
- ✓ Las recuperaciones que involucren muchos registros se pueden mejorar al leer los registros físicos en paralelo.
- ✓ El desempeño suele mejorarse de forma significativa para las aplicaciones que trabajan por lotes con muchas operaciones de escritura y de lectura/escritura de enormes registros físicos, tales como las imágenes.
- ✓ En los arreglos redundantes de discos independientes (RAID), el controlador RAID permite que un arreglo de discos se muestre al DBMS como un disco único muy grande. Para obtener un alto desempeño, el controlador RAID puede controlar hasta 90 discos.
- ✓ Debido a este controlador, el almacenamiento RAID no requiere cambios que deban tomarse en cuenta para el procesamiento en paralelo al hacer la evaluación de un plan de acceso.
- ✓ La distribución es un concepto importante del almacenamiento RAID.
- ✓ La distribución incluye la colocación de los registros físicos en distintos discos.
- ✓ Una distribución es un conjunto de registros físicos que pueden leerse o escribirse en paralelo,

MONITOREO Y AFINAMIENTO

- **Procesamiento paralelo**
 - ✓ **Para utilizar el almacenamiento RAID han surgido varias arquitecturas. Las arquitecturas RAID-X soportan el procesamiento en paralelo con diferentes elementos de desempeño y confiabilidad.**
 - ✓ **La confiabilidad es un elemento importante ya que el tiempo entre las fallas (una medida de la confiabilidad de los discos) disminuye conforme aumenta el número de discos.**
 - ✓ **Para combatir las preocupaciones de la confiabilidad, las arquitecturas RAID incorporan redundancia con el uso de discos en espejo, códigos de corrección de errores y discos de repuesto.**
 - ✓ **Para la mayoría de estos propósitos dominan las arquitecturas RAID-1 y RAID-5.**

MONITOREO Y AFINAMIENTO

- **Procesamiento paralelo**

- ✓ **RAID-1:**

- **Incluye un espejo completo o arreglo redundante de discos para mejorar la confiabilidad.**
- **Cada registro físico se escribe en los dos arreglos de discos en paralelo.**
- **Las operaciones de lectura de consultas separadas pueden acceder al arreglo de discos en paralelo para mejorar el desempeño entre las consultas.**
- **incluye la mayor sobrecarga de almacenamiento comparada con otras arquitecturas RAID.**

MONITOREO Y AFINAMIENTO

- **Procesamiento paralelo**

- ✓ **RAID-5:**

- **Utiliza tanto las páginas de datos como de corrección de errores (conocidas como páginas de paridad) para mejorar la confiabilidad.**
- **Las operaciones de lectura se pueden llevar a cabo en paralelo en las distribuciones.**
- **Las operaciones de escritura involucran una página de datos y una página de corrección de errores en otro disco.**
- **Para reducir la contención de los discos, las páginas de corrección de errores se localizan de manera aleatoria entre los discos.**
- **Usa el espacio de almacenamiento de forma más eficiente que RAID-1, pero puede llevar a tiempos de escritura más lentos debido a las páginas de corrección de errores.**
- **Por lo general, se prefiere RAID-1 para las partes altamente volátiles de una base de datos.**

MONITOREO Y AFINAMIENTO

- **Procesamiento paralelo**

- ✓ **RAID-5:**

- **Utiliza tanto las páginas de datos como de corrección de errores (conocidas como páginas de paridad) para mejorar la confiabilidad.**
- **Las operaciones de lectura se pueden llevar a cabo en paralelo en las distribuciones.**
- **Las operaciones de escritura involucran una página de datos y una página de corrección de errores en otro disco.**
- **Para reducir la contención de los discos, las páginas de corrección de errores se localizan de manera aleatoria entre los discos.**
- **Usa el espacio de almacenamiento de forma más eficiente que RAID-1, pero puede llevar a tiempos de escritura más lentos debido a las páginas de corrección de errores.**
- **Por lo general, se prefiere RAID-1 para las partes altamente volátiles de una base de datos.**

MONITOREO Y AFINAMIENTO

- **Procesamiento paralelo**
 - ✓ **Para aumentar la capacidad más allá de RAID y eliminar la dependencia de los dispositivos de almacenamiento propios del servidor, se han desarrollado las redes de áreas de almacenamiento (Storage Area Networks o SANs).**
 - ✓ **Una SAN es una red especializada de alta velocidad que conecta los dispositivos de almacenamiento y los servidores.**
 - ✓ **El objetivo de la tecnología SAN es integrar de forma sencilla distintos tipos de subsistemas de almacenamiento en un solo sistema y eliminar el potencial cuello de botella de un único servidor que controle los dispositivos de almacenamiento.**
 - ✓ **Muchas organizaciones grandes usan las SAN para integrar sistemas de almacenamiento de bases de datos operacionales, data warehouses, almacenamiento histórico de documentos y sistemas de archivos tradicionales.**

MONITOREO Y AFINAMIENTO

- **Otras consideraciones para mejorar el desempeño de las bases de datos**
 - ✓ **Para mejorar el desempeño bajo el tipo de procesamiento transaccional se puede agregar capacidad de cómputo (más procesadores y más rápidos, memoria y disco duro) y evaluar las ventajas y desventajas en el diseño transaccional.**
 - ✓ **Para mejorar el desempeño de los data warehouses se puede agregar capacidad de cómputo y diseñar tablas nuevas con datos derivados.**
 - ✓ **Para mejorar el desempeño bajo el tipo de procesamiento de bases de datos distribuidas se puede colocar el procesamiento y los datos en varias computadoras. Los datos se pueden colocar dividiendo una tabla de forma vertical (subconjunto de columnas) y horizontal (subconjunto de filas) para ubicar los datos cerca de donde se usen. La mayoría de los DBMS proporcionan guías y herramientas para monitorear y controlar la fragmentación.**