



Acceso a Datos

Manuel Perera Jiménez

mpererajimenez@safareyes.es

La persistencia de datos es la representación de datos que han sido de alguna manera borrados o eliminados.

Esto puede ser debido a que los datos han sido dejados intactos por un operativo de eliminación nominal, o por las propiedades físicas del medio de almacenaje. La persistencia de datos posibilita en forma inadvertida la exhibición de información sensible si el medio de almacenaje es dejado en un ambiente sobre el que no se tiene control.

Sistema de ficheros

El sistema de archivos ha sido desde los inicios de la programación el modelo preferido por los desarrolladores que hasta la fecha tiene gran relevancia para digitalizar la información. Son útiles para poco volumen de datos sin dependencias entre sí mismos:

Ventajas

- Simplicidad en su almacenamiento, guardar los archivos y descargarlos en el sistema de archivos es mucho más simple que la base de datos, ya que una simple función.
- Fácil migración: el proceso bastante sencillo basta con copiar y pegar (lo que no se recomienda) la carpeta en el destino que desee y asegurarse de que se proporcionan permisos de escritura a su destino.
- Mayor rentabilidad: Debido que en la mayoría de los casos es económico expandir su servidor web en lugar de pagar por ciertas bases de datos.
- Fácil almacenamiento en la nube, ya que existen diferentes soluciones en la actualidad en el mercado como Amazon S3 o CDN, etc, los cuales facilitan el trabajo con ficheros.

Desventajas

Sin apretar: Sin ACID (atomicidad, coherencia, aislamiento, durabilidad) las operaciones de mapeo relacional que significa que no hay garantía. Considere un escenario si sus archivos se eliminan

de la ubicación manualmente o por algunos tipos de hacking, es posible que no sepa si el archivo existe o no. Doloroso ¿verdad?

Baja seguridad: Ya que sus archivos pueden guardarse en una carpeta donde debería haber proporcionado permisos de escritura, es propenso a problemas de seguridad e invita a problemas como la piratería. Por lo tanto, es mejor evitar ahorrar en fs si no puede darse el lujo de comprometerse en términos de seguridad.

Bases de datos relacionales

Ventajas:

Portabilidad: SQL puede ser usado en laptops, computadoras, servidores o dispositivos móviles.

Experiencia y madurez: Este es uno de sus puntos más fuertes. El tiempo y la aceptación generalizada de los desarrolladores ha permitido crear gran cantidad de información y herramientas en torno a ellas.

Atomicidad: Los desarrolladores generalmente se ven dispuestos a inclinarse por los modelos relacionales gracias a la atomicidad. Esto significa que cualquier operación que se quiera ejecutar y no cumpla con los criterios de información preestablecidos, no se realizará.

Estándares bien definidos: Todos los procesos deben estar bajo los estándares que plantea el SQL. Brindando de esta forma criterios de uniformidad a la información.

Escritura simple: Gran parte de la aceptación depende de la sencillez de su método de escritura facilita la comprensión de lectura.

Desventajas:

Dificultades de crecimiento: Cuando estas bases de datos comienzan a crecer en volumen, el almacenamiento y el costo de mantenimiento se convierten en un problema de alto costo.

Cambios en la estructura: el entorno empresarial es altamente dinámico. Esto exige que se realicen cambios de forma eventual en los registros de datos. Si ejecutamos cambios, la Base de Datos debe ser modificada en su estructura para admitir las modificaciones. Si las modificaciones no se realizan esta se verá afectada y sus procesos interrumpidos.

Complejidad en la instalación: Algunas bases de datos SQL se ven condicionadas por el sistema operativo en el cual van a funcionar y los requisitos mínimos de funcionamiento de los servidores u ordenadores.

Dificultad en la interfaz: La interfaz de una base de datos SQL son más complejas que agregar algunas líneas de código.

Bases de datos no SQL

Este modelo de base de datos NoSQL trabaja con un lenguaje diferente a las relacionales. No tienen un identificador que sirva de relación entre los conjuntos de datos. Trabajan con formato NoSQL (Not only SQL) en el que los datos no necesariamente se registran en una tabla.

Ventajas:

Se ejecutan en máquinas con pocos recursos: Estos sistemas, a diferencia de los sistemas basados en SQL, no requieren de apenas computación, por lo que se pueden montar en máquinas de un coste más reducido.

Escalabilidad horizontal: Para mejorar el rendimiento de estos sistemas simplemente se consigue añadiendo más nodos, con la única operación de indicar al sistema cuáles son los nodos que están disponibles.

Pueden manejar gran cantidad de datos: Esto es debido a que utiliza una estructura distribuida.

No genera cuellos de botella: El principal problema de los sistemas SQL es que necesitan transcribir cada sentencia para poder ser ejecutada, y cada sentencia compleja requiere además de un nivel de ejecución aún más complejo, lo que constituye un punto de entrada en común, que ante muchas peticiones puede ralentizar el sistema.

Aplicaciones de big data: grandes volúmenes son manejados fácilmente por las bases de datos NoSQL.

Administración de la base de datos: Requieren menos administración práctica, cuenta con capacidades de distribución de datos y reparación automática, modelos de datos simplificados y menos requisitos de ajuste y administración.

Versatilidad: Las posibilidades de crecimiento en el volumen de datos o la posibilidad de incluir cambios sobre la forma en la que ingresan los datos sin necesidad de alterar la estructura, permite adaptarse de forma rápida a un entorno de alto dinamismo.

Crecimiento Horizontal: Son altamente escalables, si se requiere instalar mayor cantidad de nodos para ampliar la capacidad, se puede hacer sin problemas. Esto no interrumpe la usabilidad o consultas dentro de la base de datos.

Economía: No se necesitan servidores con gran cantidad de recursos para operar. La adaptabilidad y flexibilidad permiten empezar con bajos niveles de inversión en equipos e ir ampliando la capacidad a medida de las necesidades.

Desventajas:

Menos maduro: la mayoría de las alternativas de bases de datos NoSQL apenas han salido de las etapas de preproducción, y hay muchas características importantes que aún no se han implementado

Atomicidad: Algunas de estas bases de datos no incorporan la atomicidad de información. Esto puede derivar en que la información no sea consistente entre nodos.

Software poco documentado: Al ser tan relativamente nuevo, el NoSQL puede adolecer de que algunas operaciones sean limitadas por la falta de información sobre las herramientas y sus características.

Baja estandarización: No se tiene un criterio plenamente definido entre los motores que se utilizan en este tipo de base de datos. El lenguaje tiende a variar según el tipo de base de datos que se vaya a utilizar.

Herramientas GUI: la mayoría de las bases de datos NoSQL no contienen una interfaz gráfica. Requiere conocimiento especial para poder ejecutar algunas de ellas.

Sistema distribuido

Un sistema distribuido es un sistema en el que los componentes hardware o software:

- Se encuentran en computadores unidos mediante una red
- Se comunican únicamente mediante paso de mensajes

Desventajas de los S.O.D:

La desventaja principal de los S.O.D. es que el software no obstante más compacto y disperso, tiene una responsabilidad hasta cierto punto crítica; es complejo en términos de su diseño, implantación y uso; pues presenta numerosos inconvenientes, como posibles cuellos de botella, tiempos de propagación, rendimientos decrecientes, necesidad de sincronización o disponibilidad del servicio.

Uso ineficiente de los recursos distribuidos

Capacidad reducida para administrar apropiadamente grupos de procesadores y memoria localizada en distintos sitios

Enorme dependencia del desempeño de la red y de la confiabilidad de la misma.

Debilitamiento de la seguridad.

Software: No hay mucha experiencia en el diseño, implantación y uso del software distribuido, además existen pocos productos de software para los sistemas distribuidos.

Redes: Una vez que el sistema llega a depender de la red, la pérdida o saturación de ésta puede negar algunas de las ventajas que el S.O.D. debía conseguir.

Mayor complejidad en la administración y mantenimiento.

Mayor complejidad en su construcción.

Ventajas de los S.O.D.

-Economía: Es la razón número uno para elegir los S.O.D. ya que estos sistemas se suponen con una proporción precio/desempeño mucho mejor que la de un equipo de super cómputo.

-Velocidad: Los S.O.D. pueden ser más veloces y tener mayor poder de cómputo que un mainframe.

Distribución inherente: Porque ciertas aplicaciones son distribuidas en forma inherente.

Confiabilidad: Un S.O.D. ofrece mayor confiabilidad al distribuir la carga de trabajo entre muchas máquinas, la falla de un circuito descompondrá a lo más una máquina y el resto seguirá intacto.

Desarrollo incremental de la madurez del sistema: Para añadir poder de cómputo, no se necesita rediseñar ni gastar en componentes más caros, sólo hay que multiplicarlos. Por ejemplo basta con añadir más procesadores al sistema, lo que facilita la escalabilidad y su planificación.

Base de datos XML

Una base de datos en XML es un método de almacenamiento de información que permite albergar datos en formato XML. Suelen consistir en bases de datos de tipo documental y permiten que los datos en XML sean organizados y exportados.

Ventajas

Las principales ventajas de las bases de datos XML son las siguientes:

Son fáciles de leer.

Los documentos XML son sencillos de procesar.

Es un lenguaje que tiene una gran compatibilidad con SGML

El lenguaje XML es sencillo de estructurar con lo que se pueden diferenciar fácilmente las distintas partes de un documento.

Se puede importar y exportar a otras aplicaciones, programas y formatos.

Para los que no dominan del todo el XML, existen analizadores que permiten corregir errores de sintaxis, como XML Copy Editor.

Los documentos se pueden actualizar simplemente añadiendo nuevas etiquetas,

Desventajas

Son más lentas y requieren que los datos estén comprimidos para funcionar más rápidamente.

Las búsquedas son más lentas que en una base de datos relacional, ya que se deben organizar a través de texto y etiquetas.

Existe cierta limitación en cuanto a los gestores de bases de datos que pueden utilizar lenguaje XML.

Las bases de datos creadas con documentos XML no están preparadas para el almacenamiento de información a largo plazo.

Pueden existir problemas para garantizar la seguridad de los datos. Por ejemplo, no se puede configurar para definir quién puede actualizar, añadir o eliminar la información de la base de datos.

Ventajas	Inconvenientes
Los conjuntos de datos complejos pueden guardarse y consultarse de forma rápida y sencilla.	El uso de las bases de datos orientadas a objetos no está muy extendido.
Los códigos de identificación se asignan automáticamente a cada objeto.	En algunas situaciones, la gran complejidad puede acarrear problemas de rendimiento.
Funciona bien con lenguajes de programación orientados a objetos.	

Ventajas y desventajas de un ORM

Ventajas

Facilidad y velocidad de uso

Abstracción de la base de datos usada.

Seguridad de la capa de acceso a datos contra ataques.

Desventajas

En entornos con gran carga poner una capa más en el proceso puede mermar el rendimiento.

Aprender el nuevo lenguaje del ORM.

	Estructura de almacenamiento	Soporte para transacciones	Grado de implantación actual	Ejemplos
Ficheros	Árbol	Si	Medio	...
B.D.R	Tabla Estableciendo relaciones o vínculos entre datos	Si	Alto	Microsoft Access (relacional) Microsoft SQL Server (relacional) MySQL (relacional)

				Oracle Database (relacional)
B.D XML Nativas	Documento XML (.xml)	Algunas	Alto	IBM DB2 (XML puro) Microsoft SQL Server Oracle Database PostgreSQL
B.D Orientadas a objetos	Clases	Si	Bajo	PostgreSQL, Versant, ObjectDB
B.D NOSQL	Clase-valor	No	Alto	MongoDB, Redis RavenDB ObjectDB