

ACTIVIDAD 32

¿QUE ES NO SQL?

NoSQL (abreviatura de "Not Only SQL") se refiere a un tipo de base de datos que no se basa en el modelo relacional tradicional, como las bases de datos SQL.

¿Cuáles son sus limitaciones o desventajas?

Las bases de datos NoSQL, aunque muy versátiles y escalables, presentan algunas limitaciones en comparación con las bases de datos SQL. Algunas de las desventajas más comunes incluyen la falta de soporte para consultas complejas, la posibilidad de inconsistencia en los datos y la dificultad para gestionar transacciones complejas.

¿Por qué surgió NoSQL?

NoSQL surgió principalmente como respuesta a la necesidad de manejar grandes volúmenes de datos no estructurados o semiestructurados, especialmente en entornos web y de nube, donde las bases de datos relacionales (SQL) se demostraron insuficientes.

¿En qué se diferencia de una base de datos relacional (SQL)?

Las bases de datos relacionales (SQL) y no relacionales (NoSQL) se distinguen principalmente por su forma de organizar y almacenar los datos. Las relacionales almacenan los datos en tablas con filas y columnas, y utilizan SQL para interactuar con ellos.

¿Qué es la escalabilidad horizontal y cómo se relaciona con NoSQL?

La escalabilidad horizontal (o "scaling out") en el contexto de las bases de datos, especialmente las NoSQL, se refiere a la capacidad de añadir más servidores o nodos a la infraestructura para distribuir la carga de trabajo y los datos, en lugar de fortalecer un único servidor (escalabilidad vertical).

Diferencia entre datos estructurados y no estructurados

La principal diferencia entre datos estructurados y no estructurados radica en su organización y formato. Los datos estructurados tienen un esquema predefinido, lo que significa que se almacenan y organizan de una manera predeterminada, generalmente en tablas con filas y columnas. Los datos no estructurados, por otro

lado, no tienen un formato predefinido y pueden incluir una variedad de formatos y tipos de datos, como imágenes, videos, audios y documentos de texto.

Breve investigación de que es MongoDB, Cassandra, Redis, Neo4j, Dynamo DB, que son, sus utilidades, diferencia entre ellos

DynamoDB: El Dynamo Escalable

DynamoDB ofrece un rendimiento ultrarrápido y una escalabilidad prácticamente ilimitada . ¿Necesita gestionar millones de lecturas y escrituras por segundo? DynamoDB le respalda. Destaca en aplicaciones distribuidas donde la alta disponibilidad y la escalabilidad son fundamentales. Sin embargo, su naturaleza sin esquemas puede ser un arma de doble filo. Si bien ofrece flexibilidad, también puede provocar la proliferación de esquemas y posibles problemas de consistencia de los datos. Considérelo un guerrero poderoso pero ágil: rápido y adaptable, pero que requiere disciplina.

Cassandra: La prima del dinamo distribuido

Cassandra, una base de datos distribuida No-SQL similar a la robusta prima de DynamoDB, ofrece escalabilidad y tolerancia a fallos similares , y está adaptada a las escrituras, pero con un enfoque más estructurado. Cassandra utiliza un modelo de almacenamiento de columnas anchas , lo que permite estructuras de datos flexibles dentro de un esquema definido. Esto proporciona un buen equilibrio entre rendimiento y organización de datos. Sin embargo, configurar clústeres de Cassandra puede ser más complejo en comparación con el enfoque de servicio gestionado de DynamoDB. Imagine a Cassandra como una veterana experimentada: sólida y fiable, pero que requiere una inversión inicial en estrategia.

MongoDB: El maestro de documentos

MongoDB irrumpió en escena como una auténtica novedad. Su modelo orientado a documentos me resultó intuitivo, lo que me permitió almacenar documentos complejos tipo JSON con facilidad. Esto lo hace perfecto para escenarios donde las estructuras de datos pueden evolucionar rápidamente. Además, MongoDB ofrece un lenguaje de consulta completo que resulta familiar para quienes tienen experiencia en bases de datos relacionales. Sin embargo, su escalabilidad puede ser un problema para aplicaciones de gran volumen. Piense en MongoDB como el transformador de datos: adaptable y versátil, pero con un rendimiento bruto potencialmente limitado para conjuntos de datos masivos.

Redis: El pararrayos en memoria

Finalmente, Redis se robó el protagonismo en cuanto a almacenamiento en caché en memoria . Su velocidad ultrarrápida y la diversidad de sus estructuras de datos lo convirtieron en una herramienta revolucionaria para el almacenamiento en caché de datos de acceso frecuente. ¿Necesita implementar una tabla de clasificación

en tiempo real o un almacén de sesiones de alto rendimiento? Redis es la solución. Sin embargo, su naturaleza en memoria implica que la persistencia de los datos requiere una cuidadosa consideración. Considere a Redis como el explorador ultrarrápido que proporciona información en tiempo real, pero requiere un plan para la durabilidad de los datos.

Neo4j:

Neo4j es una base de datos orientada a grafos nativa, diseñada para almacenar y procesar datos interconectados, lo que permite la identificación de relaciones y la extracción de datos de manera eficiente. En esencia, Neo4j utiliza grafos para representar datos, donde los nodos representan entidades y las relaciones representan las conexiones entre ellas.

Cuando es mejor usar NoSQL? Casos de uso típicos (ej. Big Data, aplicaciones web con muchos usuarios, redes sociales, etc.) Ventajas frente a bases de datos tradicionales

Las bases de datos NoSQL son preferibles cuando se necesita gestionar grandes volúmenes de datos no estructurados o semiestructurados, cuando la escalabilidad y el rendimiento son cruciales, y cuando la flexibilidad del esquema es importante. Según InterSystems, esto las hace ideales para aplicaciones como Big Data, redes sociales, aplicaciones web a gran escala, y sistemas de recomendación.