

Bases de datos no relacionales con excelente rendimiento y modelos de datos flexibles

Introducción a DynamoDB

¿Qué son las bases de datos NoSQL?

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala. Esta página incluye recursos que lo ayudan a comprender mejor las bases de datos NoSQL y comenzar a usarlas.



Bases de datos en AWS: la herramienta correcta para el trabajo adecuado

¿Cómo funciona una base de datos NoSQL (no relacionales)?

Las bases de datos NoSQL utilizan una variedad de modelos de datos para acceder y administrar datos. Estos tipos de bases de datos están optimizados específicamente para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles, lo que se logra mediante la flexibilización de algunas de las restricciones de coherencia de datos en otras bases de datos.

Considere el ejemplo de modelado del esquema para una base de datos simple de libros:

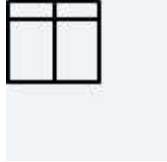
- En una base de datos relacional, un registro de libros a menudo se enmascara (o "normaliza") y se almacena en tablas separadas, y las relaciones se definen mediante restricciones de claves primarias y externas. En este ejemplo, la tabla Libros tiene las columnas ISBN, Título del libro y Número de edición, la tabla Autores tiene las columnas IDAutor y Nombre de autor y, finalmente, la tabla Autor-ISBN tiene las columnas IDAutor e ISBN. El modelo relacional está diseñado para permitir que la base de datos aplique la integridad referencial entre tablas en la base de datos, normalizada para reducir la redundancia y, generalmente, está optimizada para el almacenamiento.
- En una base de datos NoSQL, el registro de un libro generalmente se almacena como un documento [JSON](#). Para cada libro, el elemento, ISBN, Título del libro, Número de edición, Nombre autor y IDAutor se almacenan como atributos en un solo documento. En este modelo, los datos están optimizados para un desarrollo intuitivo y escalabilidad horizontal.

¿Por qué debería usar una base de datos NoSQL?

Las bases de datos NoSQL se adaptan perfectamente a muchas aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos flexibles, escalables, de alto rendimiento y altamente funcionales para proporcionar excelentes experiencias de usuario.

- Flexibilidad: las bases de datos NoSQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido y más iterativo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados.
- Escalabilidad: las bases de datos NoSQL generalmente están diseñadas para escalar usando clústeres distribuidos de hardware en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado.
- Alto rendimiento: la base de datos NoSQL está optimizada para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales.
- Altamente funcional: las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

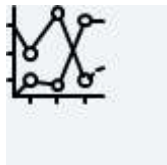
Tipos de bases de datos NoSQL



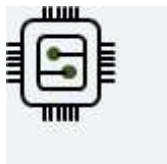
Clave-valor: las bases de datos clave-valor son altamente divisibles y permiten escalado horizontal a escalas que otros tipos de bases de datos no pueden alcanzar. Los casos de uso como juegos, tecnología publicitaria e IoT se prestan particularmente bien con el modelo de datos clave-valor. [Amazon DynamoDB](#) está diseñado para proporcionar una latencia de milisegundos constante de un solo dígito para cualquier escala de cargas de trabajo. Este rendimiento sistemático es uno de los principales elementos que explican por qué la [característica de historias de Snapchat](#), que incluye la carga de trabajo de escritura de almacenamiento más grande de Snapchat, se trasladó a DynamoDB.



Documentos: en el código de aplicación, los datos se representan a menudo como un objeto o un documento de tipo JSON porque es un modelo de datos eficiente e intuitivo para los desarrolladores. Las bases de datos de documentos facilitan a los desarrolladores el almacenamiento y la consulta de datos en una base de datos mediante el uso del mismo formato de modelo de documento que emplean en el código de aplicación. La naturaleza flexible, semiestructurada y jerárquica de los documentos y las bases de datos de documentos permite que evolucionen según las necesidades de las aplicaciones. El modelo de documentos funciona bien con catálogos, perfiles de usuario y sistemas de administración de contenido en los que cada documento es único y evoluciona con el tiempo. [Amazon DocumentDB \(con compatibilidad para MongoDB\)](#) y MongoDB son bases de datos de documentos conocidas que proporcionan API poderosas e intuitivas para un desarrollo flexible e iterativo.



Gráficos: el propósito de una base de datos de gráficos es facilitar la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos para una base de datos de gráficos incluyen redes sociales, motores de recomendaciones, detección de fraude y gráficos de conocimiento. [Amazon Neptune](#) es un servicio de base de datos de gráficos completamente administrado. Neptune admite tanto el modelo de Property Graph como el Resource Description Framework (RDF), que ofrece la opción de dos API de gráficos: TinkerPop y RDF/SPARQL. Las bases de datos de gráficos populares incluyen Neo4j y Giraph.



En memoria: las aplicaciones de juegos y tecnología publicitaria tienen casos de uso como tablas de clasificación, tiendas de sesión y análisis en tiempo real que requieren tiempos de respuesta de microsegundos y pueden tener grandes picos de tráfico en cualquier momento. [Amazon MemoryDB para Redis](#) es un servicio de base de datos en memoria duradero y compatible con Redis que ofrece latencia de lectura de microsegundos, latencia de escritura de milisegundos de un solo dígito y durabilidad Multi-AZ. MemoryDB está especialmente diseñado para ofrecer un rendimiento y una durabilidad de alta velocidad, de modo que pueda utilizarlo como su base de datos primaria para aplicaciones de microservicios modernas. [Amazon ElastiCache](#) es un servicio de almacenamiento en caché en memoria completamente administrado compatible con Redis y Memcached, para atender cargas de trabajo de baja latencia y alto rendimiento. Clientes como [Tinder](#), que requieren una respuesta en tiempo real de sus aplicaciones, confían en almacenes de datos en memoria en lugar de almacenes de datos basados en disco. [Amazon DynamoDB Accelerator](#) (DAX) es otro ejemplo de un almacén de datos especialmente diseñado. DAX hace que DynamoDB lea una orden de magnitud más rápida.

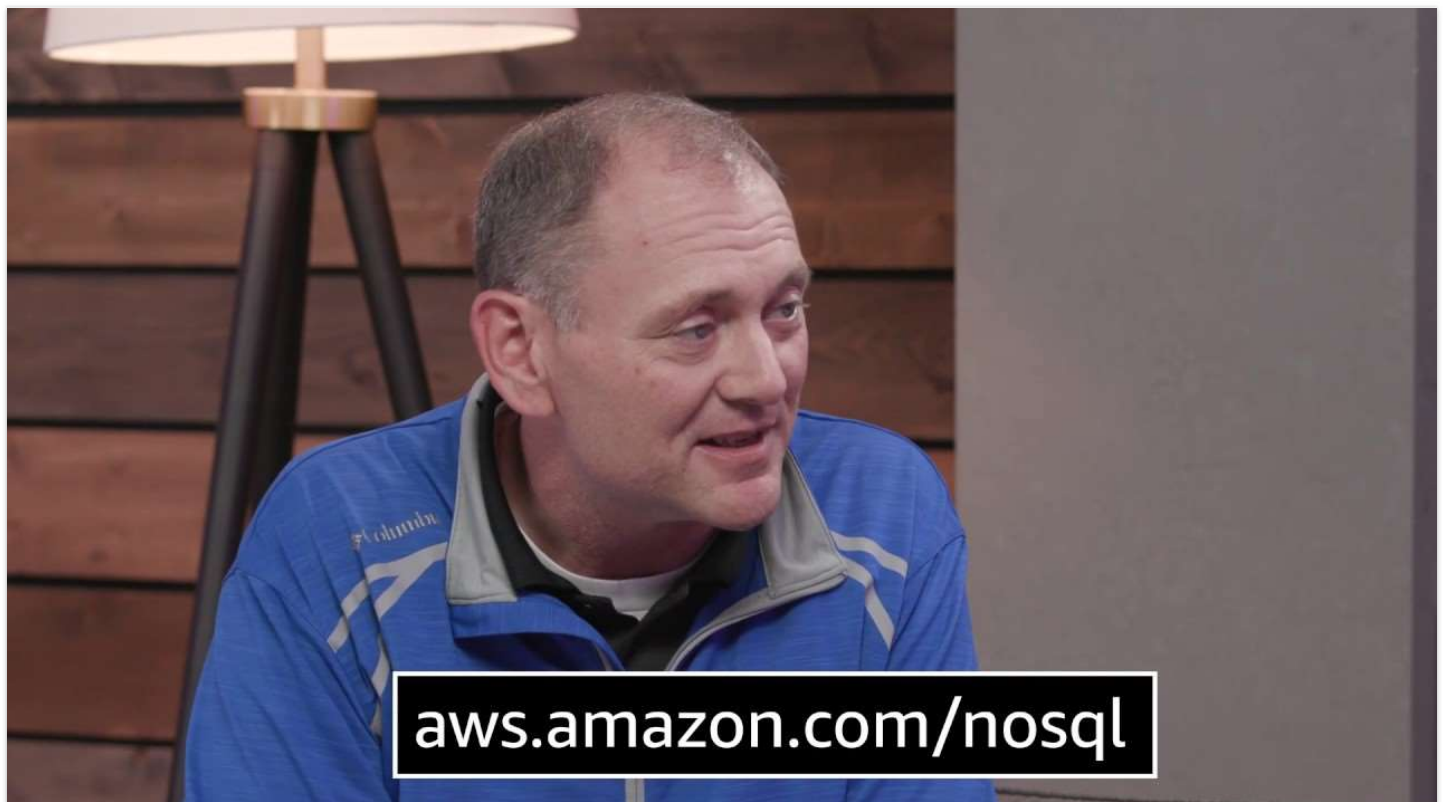


Buscar: muchas aplicaciones generan registros para ayudar a los desarrolladores a solucionar problemas. [Amazon OpenSearch Service](#) está diseñado para proporcionar visualizaciones en tiempo real y análisis de datos generados por máquinas al indexar, agregar y buscar registros y métricas semiestructuradas. Amazon OpenSearch Service también es un motor de búsqueda potente y de alto rendimiento para casos de uso de búsquedas con texto completo. [Expedia](#) está utilizando más de 150 dominios de Amazon OpenSearch Service, 30 TB de datos y 30 mil millones de documentos para una variedad de casos de uso críticos, que van desde el monitoreo operativo y la resolución de problemas, hasta el seguimiento de la pila de aplicaciones distribuidas y la optimización de precios.

SQL (relacional) en comparación con NoSQL (no relacional)

Durante décadas, el modelo de datos predominante utilizado para el desarrollo de aplicaciones era el modelo de datos relacional empleado por bases de datos relacionales como Oracle, DB2, SQL Server, MySQL y PostgreSQL. No fue sino hasta mediados y finales de la década del 2000 que otros modelos de datos comenzaron a adoptarse y aumentó su uso significativamente. Para diferenciar y categorizar estas nuevas clases de bases de datos y modelos de datos, se acuñó el término "NoSQL". Con frecuencia, los términos "NoSQL" y "no relacional" se usan indistintamente.

Aunque hay muchos tipos de bases de datos NoSQL con distintas características, en la tabla siguiente se muestran algunas de las diferencias entre las bases de datos SQL y NoSQL.



aws.amazon.com/nosql

Introducción a NoSQL

SQL en comparación con Terminología NoSQL

La siguiente tabla compara la terminología utilizada por las bases de datos NoSQL seleccionadas con la terminología utilizada por las bases de datos SQL.

SQL	MongoDB	DynamoDB	Cassandra	Couchbase
Tabla	Conjunto	Tabla	Tabla	Bucket de datos
Fila	Documento	Elemento	Fila	Documento
Columna	Campo	Atributo	Columna	Campo
Clave principal	ObjectId	Clave principal	Clave principal	ID del documento

Índice	Índice	Índice secundario	Índice	Índice
Ver	Ver	Índice secundario global	Vista materializada	Ver
Tabla u objeto anidado	Documento incrustado	Mapa	Mapa	Mapa
Matriz	Matriz	Lista	Lista	Lista

Introducción a DynamoDB

Es muy fácil comenzar a usar DynamoDB. Consulte la [página web Introducción a Amazon DynamoDB](#) para crear su primera tabla con unos pocos clics. También puede [descargar un documento técnico de AWS](#), para aprender las prácticas recomendadas para la migración de cargas de trabajo de un sistema de administración de bases de datos relacionales (RDBMS) a DynamoDB.

Introducción a Amazon DynamoDB



¿Qué es Amazon DynamoDB?