

# Introducción a las bases de datos NoSQL

## Índice de contenido

|   |   |
|---|---|
| Introducción a las bases de datos NoSQL.....        | 2 |
| ¿Qué es una base de datos NoSQL?.....               | 2 |
| Consistencia Eventual.....                          | 2 |
| Flexibilidad en el esquema.....                     | 2 |
| Escalabilidad horizontal.....                       | 3 |
| Estructura distribuida.....                         | 3 |
| Particionado (ó Sharding).....                      | 3 |
| Réplica.....  | 3 |
| Tolerancia a fallos y Redundancia.....              | 4 |
| Base de datos de Documentos.....                    | 4 |
| Almacenamiento Clave-Valor.....                     | 4 |
| Bases de datos de grafos.....                       | 4 |
| Base de datos Columnar (o Columna ancha).....       | 4 |
| ¿Qué es una base de datos de documentos?.....       | 5 |
| Definición de las bases de datos de documentos..... | 5 |
| Casos de uso.....                                   | 6 |
| Administración de contenido.....                    | 6 |
| Catálogos.....                                      | 6 |

# Introducción a las bases de datos NoSQL

## ¿Qué es una base de datos NoSQL?

Pese a la no existencia de una definición formal, cuando hablamos de base datos NoSQL, también conocidas como “No sólo SQL”, nos referimos a una amplia clase de sistemas de gestión de datos (mecanismos para el almacenamiento y recuperación de datos) que difieren, en aspectos importantes, del modelo clásico de relaciones entre entidades (o tablas) existente en los sistemas de gestión bases de datos relacionales, siendo el más destacado el que no usan SQL como lenguaje principal de consulta.

Aunque son conocidas desde la década de los 60 del pasado siglo, su auge actual viene determinado por el uso que, de estos sistemas han hecho las principales compañías de internet como Amazon, Google, Twitter y Facebook. Estas compañías tenían que enfrentarse a nuevos desafíos en el tratamiento de los datos motivados por el enorme crecimiento de la Web donde se requería dar respuesta a la necesidad de proporcionar información procesada a partir de grandes volúmenes de datos con unas estructuras horizontales, más o menos, similares y con aplicaciones web que debían dar respuesta a las peticiones de un número elevado e indeterminado de usuarios en el menor tiempo posible. Estas compañías se dieron cuenta de que el rendimiento y sus necesidades de tiempo real eran más importantes que la consistencia de los datos, aspecto este último al que las bases de datos relacionales tradicionales dedicaban una gran cantidad de tiempo de proceso.

Las características comunes entre todas las implementaciones de bases de datos NoSQL suelen ser las siguientes:

### Consistencia Eventual

A diferencia de las bases de datos relacionales tradicionales, en la mayoría de sistemas NoSQL, no se implementan mecanismos rígidos de consistencia que garanticen que cualquier cambio llevado a cabo en el sistema distribuido sea visto, al mismo tiempo, por todos los nodos y asegurando, también, la no violación de posibles restricciones de integridad de los datos u otras reglas definidas. En su lugar y para obtener un mayor rendimiento, se ofrece el concepto de “consistencia eventual”, en el que los cambios realizados “con el tiempo” serán propagados a todos los nodos por lo que, una consulta podría no devolver los últimos datos disponibles o proporcionar datos inexactos, problema conocido como lecturas sucias u obsoletas.

Asimismo, en algunos sistemas NoSQL se pueden presentar pérdidas de datos en escritura. Esto se conoce también como BASE (Basically Available Soft-state Eventual Consistency), en contraposición a ACID (Atomicity, Consistency, Isolation, Durability), su analogía en las bases de datos relacionales.

### Flexibilidad en el esquema

En la mayoría de base de datos NoSQL, los esquemas de datos son dinámicos; es decir, a diferencia de las bases de datos relacionales en las que, la escritura de los datos debe adaptarse a unas estructuras (o tablas, compuestas a su vez por filas y columnas) y tipos de datos pre-definidos, en los sistemas NoSQL, cada registro (o documento, como se les suele llamar en estos casos) puede contener una información con diferente forma cada vez, pudiendo así almacenar sólo los atributos que interesen en cada uno de ellos, facilitando el polimorfismo de datos bajo una misma colección de información. También se pueden almacenar estructuras complejas de datos en un sólo documento, como por ejemplo almacenar la información sobre una publicación de un blog (título, cuerpo de texto, autor, etc) junto a los comentarios y etiquetas vertidos sobre el mismo, todo en un único registro.

## **Escalabilidad horizontal**

Por escalabilidad horizontal se entiende la posibilidad de incrementar el rendimiento del sistema añadiendo, simplemente, más nodos (servidores) e indicando al sistema cuáles son los nodos disponibles.

## **Estructura distribuida**

Generalmente los datos se distribuyen, entre los diferentes nodos que componen el sistema. Hay dos estilos de distribución de datos:

### ***Particionado (ó Sharding)***

El particionado distribuye los datos entre múltiples servidores de forma que, cada servidor, actúe como única fuente de un subconjunto de datos. Normalmente, a la hora de realizar esta distribución, se utilizan mecanismos de tablas de hash distribuidas (DHT).

### ***Réplica***

La réplica copia los datos entre múltiples servidores, de forma que cada bit de datos pueda ser encontrado en múltiples lugares. Esta réplica puede realizarse de dos maneras

- **Réplica maestro-esclavo** en la que un servidor gestiona la escritura de la copia autorizada mientras que los esclavos se sincronizan con este servidor maestro y sólo gestionan las lecturas.
- **Réplica peer-to-peer** en la que se permiten escrituras a cualquier nodo y ellos se coordinan entre sí para sincronizar sus copias de los datos

## **Tolerancia a fallos y Redundancia**

Pese a lo que cualquiera pueda pensar cuando se habla de NoSQL, no todas las tecnologías existentes bajo este paraguas usan el mismo modelo de datos ya que, al ser sistemas altamente especializados, la idoneidad particular de una base de datos NoSQL dependerá del problema a resolver. Así a todo, podemos agrupar los diferentes modelos de datos usados en sistemas NoSQL en cuatro grandes categorías:

### ***Base de datos de Documentos***

Este tipo de base de datos almacena la información como un documento, usando para habitualmente para ello una estructura simple como JSON, BSON o XML y donde se utiliza una clave única para cada registro. Este tipo de implementación permite, además de realizar búsquedas por clave-valor, realizar consultas más avanzadas sobre el contenido del documento. Son las bases de datos NoSQL más versátiles.

### ***Almacenamiento Clave-Valor***

Son el modelo de base de datos NoSQL más popular, además de ser la más sencilla en cuanto a funcionalidad. En este tipo de sistema, cada elemento está identificado por una clave única, lo que permite la recuperación de la información de forma muy rápida, información que suele almacenarse como un objeto binario. Se caracterizan por ser muy eficientes tanto para las lecturas como para las escrituras.

### ***Bases de datos de grafos***

Usadas para aquellos datos cuyas relaciones se pueden representar adecuadamente mediante un grafo. Los datos se almacenan en estructuras grafo con nodos (entidades), propiedades (información entre entidades) y líneas (conexiones entre las entidades).

### ***Base de datos Columnar (o Columna ancha)***

En vez de “tablas”, en las bases de datos de columna tenemos familias de columnas que, son los contenedores de las filas. A diferencia de los RDBMS, no necesita conocer de antemano todas las columnas, cada fila no tiene por qué tener el mismo número de columnas. Este tipo de bases de datos se adecuan mejor a operaciones analíticas sobre grandes conjuntos de datos.

## ¿Qué es una base de datos de documentos?

### Definición de las bases de datos de documentos

Una base de datos de documentos es un tipo de base de datos no relacional que ha sido diseñada para almacenar y consultar datos como documentos de tipo JSON. Las bases de datos de documentos facilitan a los desarrolladores el almacenamiento y la consulta de datos en una base de datos mediante el mismo formato de modelo de documentos 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 casos de uso como catálogos, perfiles de usuario y sistemas de administración de contenido en los que cada documento es único y evoluciona con el tiempo. Las bases de datos de documentos permiten una indexación fácil, potentes consultas ad hoc y análisis de colecciones de documentos.

En el siguiente ejemplo, un documento de tipo JSON describe un libro.

```
[
  {
    "year" : 2013,
    "title" : "Turn It Down, Or Else!",
    "info" : {
      "directors" : [ "Alice Smith", "Bob Jones"],
      "release_date" : "2013-01-18T00:00:00Z",
      "rating" : 6.2,
      "genres" : ["Comedy", "Drama"],
      "image_url" : "http://ia.media-
imdb.com/images/N/09ERWAU7FS797AJ7LU8HN09AMUP908RLl65JF90EWR7LJKQ7@@._V1_SX400_.jpg",
      "plot" : "A rock band plays their music at high volumes, annoying the neighbors.",
      "actors" : ["David Matthewman", "Jonathan G. Neff"]
    }
  },
  {
    "year": 2015,
    "title": "The Big New Movie",
    "info": {
      "plot": "Nothing happens at all.",
      "rating": 0
    }
  }
]
```

## **Casos de uso**

### ***Administración de contenido***

Una base de datos de documentos es una excelente opción para aplicaciones de administración de contenido, como blogs y plataformas de video.

Con una base de datos documental, cada entidad que rastrea la aplicación se puede almacenar como un único documento. La base de datos de documentos es más intuitiva para que un desarrollador actualice una aplicación a medida que evolucionan los requisitos. Además, si el modelo de datos necesita cambiar, solo se deben actualizar los documentos afectados. No se requiere actualización del esquema y no es necesario tiempo de inactividad de la base de datos para realizar los cambios.

### ***Catálogos***

Las bases de datos de documentos son eficientes y efectivas para almacenar información de catálogo. Por ejemplo, en una aplicación de e-commerce, los diferentes productos generalmente tienen diferentes números de atributos. La administración de miles de atributos en bases de datos relacionales no es eficiente y afecta al rendimiento de lectura. Al utilizar una base de datos de documentos, los atributos de cada producto se pueden describir en un solo documento para que la administración sea fácil y la velocidad de lectura sea más rápida. Cambiar los atributos de un producto no afectará a otros.