

Universidad Mariano Gálvez de Guatemala

Ingeniería en Sistemas

Sede Jalapa

Marco Tulio Valde Albizurez

Introducción a los Sistemas

Investigación

Julio Rodolfo Vásquez Ortiz

0907-24-19595

Introducción

Las bases de datos NoSQL utilizan una variedad de modelos de datos para acceder y manipular datos. Estos tipos de bases de datos están optimizados para aplicaciones de uso intensivo de datos que requieren baja latencia y modelos de datos flexibles. Todo esto se logra suavizando los estrictos requisitos de consistencia de datos que son típicos para otros tipos de bases de datos.

Bases de datos no relacionales

Las **bases de datos no relacionales o NoSQL** son sistemas que se ocupan del almacenamiento de datos de una forma no tabulada y suele ser más flexible que las bases de datos tradicionales. Una de sus características principales es que no utilizan el lenguaje SQL para realizar consultas. No obstante, algunas de ellas sí que pueden usarlo como apoyo. Además, se trata de bases de datos muy **fáciles de desarrollar** y que ofrecen grandes ventajas debido a su funcionalidad y su rendimiento.

¿Qué son las bases de datos no relacionales?

Al comienzo de este artículo hemos podido ver brevemente qué es una base de datos no relacional. A continuación, vamos a profundizar más sobre su funcionamiento.

En primer lugar, cabe señalar que este tipo de base de datos no almacenan la información en filas con campos. Es decir, no se encargan de trabajar con estructuras definidas, sino que se han preparado para **gestionar grandes cantidades de datos semiestructurados o no estructurados**.

Existen cuatro tipos principales de bases de datos no relacionales:

1. Bases de datos orientadas a columna.
2. Bases de datos orientadas a documentos.
3. Bases de datos orientadas a clave-valor.
4. Bases de datos orientadas a grafos.

Ventajas de las Bases de datos no relacionales

Trabajar con bases de datos no relacionales ofrece muchas ventajas, entre las cuales destacan las siguientes:

- Su flexibilidad para elaborar esquemas de información, algo que permite almacenar y gestionar datos que son semiestructurados o no estructurados.
- Son fácilmente escalables horizontalmente y, por tanto, son utilizadas para trabajar con volúmenes grandes de información.

- Su rendimiento es alto debido a que se han preparado para trabajar con patrones y modelos de datos específicos.
- Gracias a sus API exclusivas y la proporción de modelos creados para gestionar los distintos tipos de datos se han convertido en un sistema de almacenamiento muy funcional.

Desventajas de las Bases de datos no relacionales

Aunque utilizar bases de datos no relacionales para almacenar información ofrece múltiples beneficios, también hay que tener en cuenta una serie de desventajas:

- No cumplen con las propiedades ACID como sí lo hacen las relacionales.
- Son incompatibles con las consultas realizadas en lenguaje SQL.
- Todavía no existe un sistema estandarizado.
- Al ser de código abierto, no poseen el soporte que proporciona una empresa comercial, siendo este muchas veces limitado.
- Dado que no comparten un estándar, se pueden presentar problemas de compatibilidad.

¿Qué es MongoDB?

[MongoDB \(enlace externo a IBM\)](#) es un sistema de gestión de bases de datos (DBMS, por sus siglas en inglés) no relacionales y de código abierto, que utiliza documentos flexibles en lugar de tablas y filas para procesar y almacenar varias formas de datos. Al ser una solución de **[base de datos NoSQL](#)**, MongoDB no requiere un sistema de gestión de bases de datos relacionales (RDBMS), por lo que proporciona un modelo de almacenamiento de datos elástico que permite a los usuarios almacenar y consultar fácilmente tipos de datos multivariados. Esto no solo simplifica la gestión de la base de datos para los desarrolladores, sino que también crea un entorno altamente escalable para aplicaciones y servicios multiplataforma.

Los documentos o colecciones de documentos de MongoDB son las unidades básicas de datos. Estos documentos, con formato JSON (Java Script Object Notation) binario, pueden almacenar varios tipos de datos y distribuirse en varios sistemas. El diseño de esquema dinámico de MongoDB brinda a los usuarios una flexibilidad sin igual para crear registros de datos, consultar colecciones de documentos a través de la agregación de MongoDB y analizar grandes cantidades de información.

Ventajas de MongoDB

A lo largo de los años, MongoDB se ha afianzado como solución para muchas empresas que buscan una base de datos NoSQL potente y altamente escalable. Pero MongoDB es mucho más que una base de datos tradicional basada en documentos y cuenta con unas prestaciones excelentes que la hacen destacar entre otras DBMS.

Equilibrio de carga

El auge de las aplicaciones en cloud y el incremento de la demanda de recursos para las empresas complican la disponibilidad y la fiabilidad de los servicios. El proceso de uso compartido del [equilibrio de carga](#) de MongoDB distribuye grandes conjuntos de datos a través de varias [máquinas virtuales](#) a la vez manteniendo rendimientos aceptables de lectura y grabación. Esta escalada horizontal se llama *sharding* (fragmentación) y ayuda a las organizaciones a evitar el coste de la escalada vertical de hardware al mismo tiempo que expande la capacidad de los despliegues basados en cloud.

Raven DB

Raven es un proyecto de código abierto, pero con una opción comercial para base de datos para plataformas .NET/Windows.

Raven ofrece un diseño de modelo de datos flexible para adaptarse a las necesidades de los sistemas del mundo real. Este sistema permite utilizar las consultas Linq que se basan en la baja latencia y el obtienen un alto rendimiento.

Entre las características de este sistema podemos destacar:

Infraestructura escalable

Configuración sencilla en Windows. Es fácil de instalar y se ejecuta en ventanas como un servicio normal o un sitio web de IIS7.

Definir fácilmente los índices con consultas LINQ.

API .NET. Raven viene con un api que implementa la unidad de trabajo.

Raven está publicado bajo licencia de código abierto y una licencia comercial, lo que significa que está libremente disponible, pero si se quiere utilizar con un software propietario, habría que adquirir una licencia comercial.

¿Qué es CouchDB?

[Apache CouchDB](#) (enlace externo a ibm.com) es una base de datos de documentos NoSQL de código abierto que recopila y almacena datos en formatos de documentos basados en JSON. A diferencia de las bases de datos relacionales, CouchDB utiliza un modelo de datos sin esquema, que simplifica la gestión de registros en varios dispositivos informáticos, teléfonos móviles y navegadores web.

CouchDB fue lanzada en 2005 y luego se convirtió en un proyecto de [Apache Software Foundation](#) (enlace externo a ibm.com) en 2008. Como proyecto de código abierto, CouchDB cuenta con el apoyo de una comunidad activa de desarrolladores que mejoran continuamente el software con un enfoque en la facilidad de uso y la adopción de la web.

¿Por qué utilizar CouchDB?

¿Qué es CouchDB?

[Apache CouchDB](#) (enlace externo a [ibm.com](#)) es una base de datos de documentos NoSQL de código abierto que recopila y almacena datos en formatos de documentos basados en JSON. A diferencia de las bases de datos relacionales, CouchDB utiliza un modelo de datos sin esquema, que simplifica la gestión de registros en varios dispositivos informáticos, teléfonos móviles y navegadores web.

CouchDB fue lanzada en 2005 y luego se convirtió en un proyecto de **[Apache Software Foundation](#)** (enlace externo a [ibm.com](#)) en 2008. Como proyecto de código abierto, CouchDB cuenta con el apoyo de una comunidad activa de desarrolladores que mejoran continuamente el software con un enfoque en la facilidad de uso y la adopción de la web.

¿Por qué utilizar CouchDB?

Para la mayoría de las empresas, decidir qué proveedor utilizar al adquirir tecnología de gestión de datos puede ser un desafío. Un software privado no solo impone ciertas restricciones de licencia para usar la tecnología, sino que existen algunas inquietudes acerca de la continuidad del negocio cuando se transfieren todos los datos de su empresa a un sistema de gestión de bases de datos (DBMS) "universal" o "único para todos" sin visibilidad de su estructura interna.

CouchDB es diferente. A diferencia del software privado que puede facilitar la dependencia del proveedor, CouchDB es de código abierto, de uso gratuito y se integra fácilmente en su infraestructura de gestión de datos actual. Debido a que usted tiene más control sobre el software, también tiene más flexibilidad para adaptarlo a las necesidades únicas de su negocio. Ya sea que requiera un almacén de documentos de propósito general, que permita una sincronización de datos eficiente o que adopte una mentalidad **["Sin conexión primero"](#)**, CouchDB brinda a las empresas la flexibilidad que necesitan para crear infraestructuras duraderas, confiables y escalables.

¿Qué es Firebase?

Firestore de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Aunque fue creada en 2011 [pasó a ser parte de Google en 2014](#), comenzando como una base de datos en tiempo real. Sin embargo, se añadieron más y más funciones que, en parte, permitieron agrupar los SDK de productos de Google con distintos fines, facilitando su uso.

¿Para qué sirve Firestore?

Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.

Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de las mismas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al backend, tanto en cuestiones de desarrollo como de mantenimiento.

Tal vez estés buscando en internet: tutorial Firestore, manual firestore o dev firestore, pero la mejor forma de entender en qué consiste Firestore es hacernos una idea de las herramientas que ofrece y a qué están destinadas, algo que veremos a continuación.

Funciones de Firestore

Firestore dispone de diferentes funcionalidades, que se pueden dividir básicamente en 3 grupos: Desarrollo (Develop), Crecimiento

(Grow) y Monetización (Earn), a los que hay que sumar la Analítica (Analytics).

Desarrollo

El primer grupo de funciones es conocido como Desarrollo o Develop en Firebase. Como su nombre indica, incluye los servicios necesarios para el desarrollo de un proyecto de aplicación móvil o web. Estos contribuyen a que el proceso sea más rápido, puesto que se dejan determinadas actividades a mano de Firebase, mientras que otras permiten optimizar diversos aspectos para conseguir la calidad deseada.

REALTIME DATABASE

Una de las herramientas más destacadas y esenciales de Firebase son las bases de datos en tiempo real. Estas se alojan en la nube, son No SQL y almacenan los datos como JSON. Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados, aunque el usuario no realice ninguna acción.

Firestore envía automáticamente eventos a las aplicaciones cuando los datos cambian, almacenando los datos nuevos en el disco. Aunque no hubiera conexión por parte de un usuario, sus datos estarían disponibles para el resto y los cambios realizados se sincronizarían una vez restablecida la conexión.

Bases de datos relacionales vs. no relacionales

Es muy común que los desarrolladores de software estemos en la situación de elegir entre bases de datos relacionales y no

relacionales. No es para verlo como un dilema o pensar que dependiendo de la decisión vamos a fracasar, en realidad con cualquiera de las dos opciones, es posible construir nuestro sistema, dependiendo de lo que queramos.

Entonces surgen diferentes interrogantes: ¿Es importante saber las diferencias entre ambas bases de datos? ¿Cuál deberíamos usar en cada caso? Estas son preguntas bastante interesantes, puesto que un buen diseño de la base de datos es un factor de alta influencia con respecto a la calidad del proyecto. Dependiendo de la naturaleza de éste, se debe elegir la base de datos, ya que las dos tienen diferentes características.

Bases de Datos relacionales

Son una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas, desde donde se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base. La interfaz estándar de programa de usuario y aplicación a una base de datos relacional, es el Lenguaje de Consultas Estructuradas (SQL). Los comandos SQL se utilizan tanto para consultas interactivas como para obtener información de una base de datos relacional y la recopilación de datos para informes.

Las bases de datos relacionales se basan en la organización de la información en partes pequeñas que se integran mediante identificadores; a diferencia de las bases de datos no relacionales que, como su nombre lo indica, no tienen un identificador que sirva para relacionar dos o más conjuntos de datos. Además, son más robustas, es decir, tienen mayor capacidad de almacenamiento, y son menos vulnerables ante fallas, estas son sus principales características.

Bases de datos no relacionales

Están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Son ampliamente reconocidas porque son fáciles de desarrollar, tanto en funcionalidad como en rendimiento a escala. Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en-memoria y búsqueda.

Las bases de datos no relacionales (NoSQL) son las que, a diferencia de las relacionales, no tienen un identificador que sirva de relación entre un conjunto de datos y otros. Como veremos, la información se organiza normalmente mediante documentos y es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar.

Diferencias entre bases de datos SQL y NoSQL

SQL	NoSQL
Permite unir tablas con JOIN	No permite el uso de JOINS o están muy limitados
Dificultad para distribuir los datos	Facilidad para distribuir los datos
Escalabilidad vertical (mejorar potencia del servidor para obtener mejores resultados)	Escalabilidad horizontal (repartir / distribuir base de datos en diferentes servidores)
Los datos se estructuran siguiendo los esquemas de las tablas	Los datos no siguen ningún esquema
Posibilidad de crear restricciones, triggers, claves foráneas... sobre los datos	Estas utilidades no suelen estar disponibles

Conclusión

Las bases de datos NoSQL son ya una opción más en la cartera de alternativas para almacenar los datos de aplicaciones. Existen varios tipos de ellas, pero en general su objetivo principal es resolver los problemas de performance y de escalabilidad de las RDBMS.

E-grafias

Ub1es/blog-ui1/bases-de-datos-no-relacionales

lbn.com/es-es/topic/mongodb

Programación.net/noticia/raven-db-1703

lbn.com/mx-es/topic/couchdb

Digitales55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones

Programa.com.co/academi/lecciones/bases-de-datos-relacionales-vs.-no-relacionales