

# Comparación Bases de datos NoSQL

Andre Reinoso, Samuel Nuñez, Andres De la Barra ,David Damian y Richard Cruz

June 23, 2019

Abstract

*Inglés*

Abstract

*Español.*

## I. INTRODUCCION

Las bases de datos son una de las tecnologías para la organización de la información más eficientes y poderosas de que disponemos en la actualidad. De hecho, las bases de datos están en el núcleo de los sistemas y de los servicios de información que poseen mayor significación tanto desde un punto de vista económico como social. Por ejemplo, la mayor parte de los sistemas de información empresarial, desde el control de existencias de almacén hasta el sistema de ventas, tiene en su núcleo una base de datos. En el seno de muchas actividades sociales vinculadas con la gestión del conocimiento sucede otro tanto: desde la gestión de archivos y museos hasta la automatización de centros de documentación, pasando por los mejores servicios de información en Internet o por la gestión de investigaciones académicas, todos estos sistemas tienen una base de datos en su núcleo. En el presente artículo se desarrollará el tema de bases de datos NoSQL.

## II. OBJETIVOS

- Entender qué es una máquina virtual.
- Entender qué es un contenedor.
- Comparar ambos conceptos.
- Establecer un juicio acerca de las ofertas y el potencial de ambas.

## III. DESARROLLO

### i. ¿Que es una maquina virtual?

Una máquina virtual

	last_name	job_id	Sal
1	King	AD_PRES	24000.00
2	Kochhar	AD_VP	17000.00
3	De Haan	AD_VP	17000.00
4	Hunold	IT_PROG	9000.00
5	Ernst	IT_PROG	6000.00
6	Austin	IT_PROG	4800.00
7	Pataballa	IT_PROG	4800.00
8	Lorentz	IT_PROG	4200.00
9	Greenberg	FI_MGR	12008.00
10	Faviet	FI_ACCOUNT	9000.00
11	Chen	FI_ACCOUNT	8200.00
12	Sciarra	FI_ACCOUNT	7700.00

### ii. Insercion de datos y Consultas de datos.

En este caso haremos la insercion de datos y consultas en MongoDB instalado en Docker.

- Como primer paso, se debe realizar la conexión de MongoDB:  
docker exec -it my-mongodb mongo

```
PS C:\Users\Usuario> docker exec -it my-mongodb mongo
```

- Seguidamente crearemos la colección "Base De Datos2", Después insertaremos los datos a nuestra colección:  
db.createCollection('basededatos')

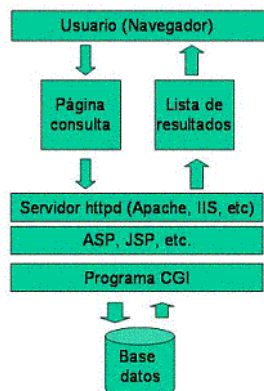
```
Windows PowerShell
PS C:\Users\Usuario> db.createCollection( 'basededatos' )
PS C:\Users\Usuario> db.basededatos.insert({ "_id": "1", "name": "david" })
```

- Por ultimo vamos a consultar u obtener todo los datos con el siguiente codigo:

```
Windows PowerShell
PS C:\Users\Usuario> db.basededatos.find();
```

### iii. Documental

Cuando hablamos de bases de datos documentales, nos referimos a un conjunto de información estructurada en registros y almacenada en un soporte electrónico legible desde un ordenador. Cada uno de los registros compone una unidad autónoma de información que puede estar a su vez estructurada en diferentes campos o tipos de datos que se almacenan en dicha base de datos. Por ejemplo, entre los campos se pueden encontrar: nombre del documento, título, palabras clave que caracterizan el tema tratado en el documento, fecha o autor. Para cada uno de los documentos que componen la base de datos es necesario hacer un registro, de esta manera, se puede recuperar dicho registro y determinado documento.



Una base de datos documental utiliza documentos como la estructura para almace-

namiento y consultas. En este caso, el término "documento" puede referirse a un documento de texto, pero comúnmente también puede un archivo de XML o JSON. En lugar de columnas con nombres y tipos de datos que se utilizan en una base de datos relacional, un documento contiene una descripción del tipo de datos y el valor de esa descripción.

#### Tipos de bases de datos documentales:

- Bases de datos a texto completo
- Bases de datos referenciales
- Multidisciplinarias
- Especializadas
- Bases de datos de acceso local
- Bases de datos en CD ROM
- Bases de datos en línea

#### Ventajas:

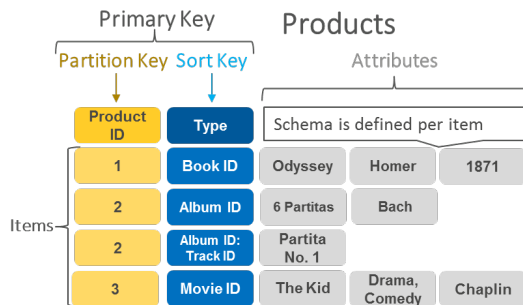
Hoy en día las empresas tienen que trabajar con una gran cantidad de documentación y es muy complicado gestionar todos los documentos diarios que se generan. Por este motivo, ahora más que nunca, es imprescindible contar con un sistema de bases de datos documental en la empresa. Estas son algunas de las ventajas que aporta el uso de las bases de datos documentales:

- Capaces de almacenar cualquier tipo de información en forma de texto.
- Facilidad de poder manejar una gran cantidad de información, de forma rápida y en muy poco tiempo.
- Aumenta el rendimiento general de la empresa al poder recuperar los documentos de forma ágil y fácil.
- Poseen un lenguaje de consulta fácil e intuitivo.
- Capaz de manejar una gran cantidad de datos.
- Escalabilidad y ahorro de espacio.

### iv. Clave - Valor

Una base de datos clave-valor es un tipo de base de datos no relacional que utiliza un método simple de clave-valor para almacenar datos. Una base de datos clave-valor almacena datos como un conjunto de pares clave-valor

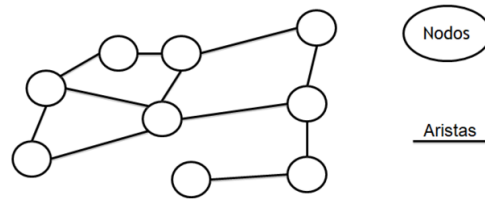
en los que una clave sirve como un identificador único. Tanto las claves como los valores pueden ser cualquier cosa, desde objetos simples hasta objetos compuestos complejos. Las bases de datos clave-valor son altamente divisibles y permiten el escalado horizontal a escalas que otros tipos de bases de datos no pueden alcanzar. Las claves pueden ser organizadas por grupos clave lógicos, requiriendo solamente estas claves para ser únicas dentro de su propio grupo. Esto permite tener claves idénticas en diferentes grupos lógicos. Algunas implementaciones del almacén de valores clave proporcionan mecanismos de almacenamiento en el caché, lo que mejora en gran medida su rendimiento.



Sus principales ventajas son 3: simplicidad, eficiencia y flexibilidad, que permiten unas búsquedas rápidas en lecturas a toda la BBDD, así como funciones de agregación efectivas. Por otro lado, la simplicidad también marcará sus principales inconvenientes: al carecer de estructura no es posible lanzar consultas mediante queries, y solo lo constan de una colección, complicado la implementación modelos complejos.

## v. Grafos

Las bases de datos NoSQL en grafo permiten representar los datos utilizando estructuras de grafos. Un grafo es una representación abstracta de un conjunto de objetos. Los objetos de los grafos se representan mediante vértices (también llamados nodos) y aristas.

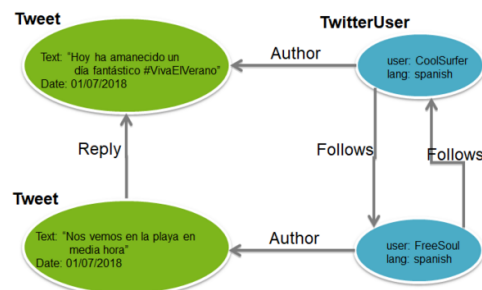


El modelo en grafo es útil cuando los datos a almacenar tienen multitud de interrelaciones entre sí, y cuando la importancia recae más en las interrelaciones que se establecen entre los datos, que en los propios datos en sí.

En consecuencia, este tipo de bases de datos tiende a almacenar pocos datos de los objetos del mundo real que se desean representar pero muchos datos sobre sus interrelaciones, a diferencia de lo que acostumbra a suceder en las bases de datos relacionales, donde hay muchos datos de los objetos (representados mayoritariamente en las propiedades o atributos de las relaciones) y pocas interrelaciones entre los objetos (representadas mediante claves foráneas).

### • Ejemplo

A continuación podemos ver un ejemplo simple en el que hemos creado un grafo con dos tweets (en color verde) y dos usuarios de Twitter (en color azul). Como vemos, los tweets y los usuarios se han etiquetado con las etiquetas "Tweet" y "TwitterUser" para indicar su tipo. Vemos también que uno de los tweets es un reply (se indica mediante una relación con una etiqueta "Reply") y que los dos usuarios de Twitter se siguen mutuamente (se indica mediante la etiqueta "Follows"). También se indican los autores de cada tweet (mediante la etiqueta "Author").



## vi. Column-Store

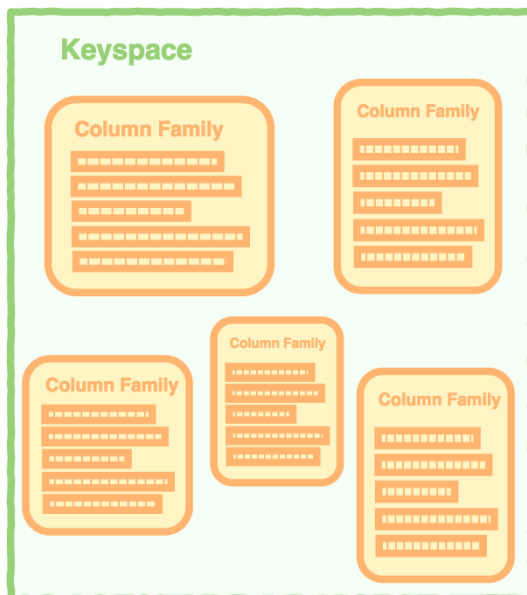
Es un tipo de base de datos que almacena datos utilizando un modelo orientado a columnas.

Tambien denominadas:

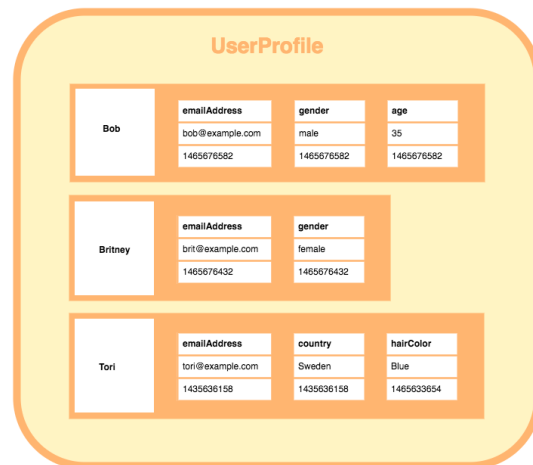
- Column database
- Column family database
- Column oriented database
- Wide column store
- Columnar database
- Columnar store

Utilizan un concepto llamado keyspace . Un keyspace es algo así como un esquema en el modelo relacional. El keyspace contiene todas las Column Family (tipo de tablas similares en el modelo relacional), que contienen filas, que contienen columnas.

### Keyspace

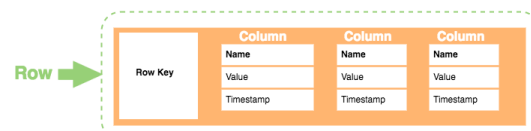


### Column Family



- Cada fila puede contener un número diferente de columnas a las otras filas. Y las columnas no tienen que coincidir con las columnas de las otras filas (es decir, pueden tener diferentes nombres de columnas, tipos de datos, etc.).
- Cada columna está contenida en su fila. No abarca todas las filas como en una base de datos relacional. Cada columna contiene un par de nombre / valor, junto con una marca de tiempo

### Construccion de una fila



- **Row Key:** Cada fila tiene una clave única, que es un identificador único para esa fila.
- **Column:** Cada columna contiene un nombre, un valor y una marca de tiempo.
- **Name:** Este es el nombre del par name/value.
- **Value:** Este es el valor del par name/value.
- **Timestamp:** Esto proporciona la fecha y la hora en que se insertaron los datos. Esto se puede utilizar para determinar la versión más reciente de los datos.

## vii. Comparando tipos de base de datos NoSQL

Más allá de las ventajas y desventajas que podamos encontrar en los tipos de bases de datos NoSQL, creemos que cada tipo de base de datos se puede aprovechar de una forma especial para un fin determinado.

Un grupo de ingenieros se reunieron para estudiar una comparación de calidad entre tres bases de datos populares de distintos tipos (Redis, Neo4j y MongoDB) aplicando un caso de uso real para cada tipo. El objetivo de ellos era el de evaluar las diferencias inherentes entre las bases de datos al definir los datos.

De acuerdo con la comparación que hicieron, pudieron inferir los siguientes hallazgos:

- Las basadas en grafos están diseñadas para ser adecuadas para representar datos fuertemente vinculados y relaciones intensivas como redes sociales, datos geográficos y bioinformática.
- Las basadas en documentos son adecuadas para administrar colecciones con relaciones abstractas.
- Las basadas en clave-valor son adecuadas cuando las relaciones no son un problema, como recuperar información sobre los nombres de productos favoritos de los clientes, los carritos de la compra y la sesión de un usuario.
- Se puede considerar la combinación de más de un tipo de base de datos para cumplir con más de uno de los objetivos anteriores

Ellos encontraron que las bases de datos basadas en grafos eran la mejor opción cuando se trataba de relaciones intensivas. Las bases de datos basadas en documentos eran mejor cuando se trataba de colecciones y relaciones abstractas. Finalmente, las bases de datos clave-valor eran las mejores cuando las relaciones no eran un problema.

Pero siempre, como último objetivo fue el de ayudar a las organizaciones a encontrar bases de datos NoSQL adecuados y que se adapten a sus necesidades.

## IV. CONCLUSIONES

Para hablar de contenedores y

### REFERENCES

- [Codina, 2015] Codina, L (2015). Sistemas gestion de bases de datos documentales.
- [Martin, 2011] Martin, M.M, y J.U (2011). Virtualización, una solución para la eficiencia, seguridad y administración de intranets *El profesional de la informacion*, 350. Contenedor de aplicaciones: Docker (2015)
- [Hernandez, 2017] Primeros pasos en MongoDB. Instalación en Docker. Find y Aggregation