|  |
| --- |
|  |
| INVESTIGACIÓN: BASES DE DATOS NOSQL |
|  |



Antony Romero Bonilla

Diego Artavia Chacón

Jimmy Corella Camacho

Lanford Gabriel Murillo

**Objetivo general:**

* Recolectar y analizar la información sobre las Bases de Datos NoSQL, para crear un documento que mejore entendimiento.

**Objetivos específicos:**

* Definir que son bases de datos NoSQL.
* Definir las ventajas y desventajas del uso de bases de datos NoSQL.
* Identificar los diferentes tipos de bases de datos NoSQL.
* Determinar las formas de implementación una base de datos NoSQL.

**Introducción:**

Esta investigación fue hecha con el propósito de recopilar información que mejore el entendimiento de las bases de datos NoSQL. Encontrará información relacionada con las BD NoSQL como:

* 1. ¿Qué es NoSQL?
  2. Origen de NoSQL.
  3. ¿Por qué utilizar NoSQL?
  4. Ventajas y desventajas de las NoSQL.
  5. Métodos para implementar NoSQL.
  6. Soluciones más sobresalientes de NoSQL.
  7. ¿Quiénes utilizan NoSQL?

**¿Qué es NoSQL?**

NoSQL(Not only SQL – No solo SQL). Se puede decir que la aparición del término NoSQL aparece con la llegada de la web 2.0 ya que hasta ese momento sólo subían contenido a la red aquellas empresas que tenían un portal, pero con la llegada de aplicaciones como Facebook, Twitter o YouTube, cualquier usuario podía subir contenido, provocando así un crecimiento exponencial de los datos.

Es en este momento cuando empiezan a aparecer los primeros problemas de la gestión de toda esa información almacenada en bases de datos relacionales. En un principio, para solucionar estos problemas de accesibilidad, las empresas optaron por utilizar un mayor número de máquinas pero pronto se dieron cuenta de que esto no solucionaba el problema, además de ser una solución muy cara. La otra solución era la creación de sistemas pensados para un uso específico que con el paso del tiempo han dado lugar a soluciones robustas, apareciendo así el movimiento NoSQL.

Por lo tanto hablar de bases de datos NoSQL es hablar de estructuras que nos permiten almacenar información en aquellas situaciones en las que las bases de datos relacionales generan ciertos problemas debido principalmente a problemas de escalabilidad y rendimiento de las bases de datos relacionales donde se dan cita miles de usuarios concurrentes y con millones de consultas diarias.

Las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad–relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos sino que para el almacenamiento hacen uso de otros formatos como clave–valor, mapeo de columnas o grafos (ver epígrafe ‘Tipos de bases de datos NoSQL’).

NoSQL difiere de los RDBMS (Relational Database Management System) en diferentes modos:

* No tienen Schemas.
* No permiten joins.
* No garantizan la propiedad ACID (Atomicity, Consistency, Isolation and Durability).
* Escalan horizontalmente.
* Resuelven el problema de los altos volúmenes de información y la inmensa cantidad de consultas y transacciones diarias
* La principal diferencia que tienen con una base de datos relacional es en la manera en la que se guardan los datos:
* En las bases de datos relacionales tienes que dividir la información en tablas y luego en la parte del servidor convertir esos datos a objetos.
* NoSQL por lo tanto, es libre de schemas, tu no diseñas tus tablas.
* Pero si tus datos son relacionales es mejor seguir con una RDBMS.
* Características principales de una base de datos NoSQL.
* Facilitan escalabilidad horizontal.
* Guardan datos persistentes.
* No tienen esquemas fijos y permiten la migración del esquema sin necesidad de reiniciar o detenerla.
* Por lo general tienen un sistema de consultas propio.

**Origen de NoSQL**

Iniciando aproximadamente a inicios de los 70’s, el modelo relacional se volvió tan dominante que el mundo pareció olvidar que habían muchas otras maneras de almacenar datos. Pero, conforme los almacenes de datos se volvieron más y más grandes, éste modelo comenzó a ser cuestionado.

Entre más crecen los almacenes de datos comienzas a ver que cada vez son más las conexiones entre los objetos, los que significa que cada consulta va a tener demasiados “joins”, haciendo casi imposible expresar las consultas por medio de SQL.

Así que las alternativas, empezaron a aparecer...

El creador del término fué Carlos Strozzi en 1998, y posteriormente resucitado por Eric Evans en el año 2009.

No nacieron en 2009 sino que se remontan a la época de las bases de datos de red y jerárquicas, una serie de productos que no eran relacionales y que en la época en que surgieron no se tenía internet. Desde 1965 se han venido desarrollando productos para almacenamiento masivo, datos multivalor, de red, jerárquicos, con estructuras B+, productos de procesamiento de transacciones de alto desempeño llave valor.

Por ejemplo Neo4j empezó en el año 2000, pero si algo contribuyó al desarrollo de los productos noSQL fueron la serie de “papers” publicados por Google en 2003, 2004 y 2006 sobre cómo construir una infraestructura escalable para el procesamiento paralelo de grandes cantidades de datos, que originó Hadoop. Más tarde en 2007 Amazon liberó su historia sobre Dynamo, el almacenamiento llave/Valor de alta disponibilidad.

En el 2012 la cantidad de productos noSQL pasó a ser un poco más de 120.

Algunos ejemplos de bases de datos no relacionales que han surgido en los últimos años:

* JackRabbit - 2006.
* Tokyo Cabinet - 2006.
* Amazon Dynamo - 2007.
* MongoDB - 2007.
* Cassandra - 2008.
* Proyecto Voldemort - 2008.
* Terrastore - 2009.
* Redis - 2009.
* Riak - 2009.
* HBase - 2009.
* Vertexdb - 2009.

**¿Por qué utilizar NoSQL?**

Las Bases de Datos NoSQL permiten manejar grandes volúmenes de información con buen rendimiento y así poder emplear soluciones de Big Data además son muy utilizadas por empresas de gran renombre como Google, razones suficientes para sean muy utilizadas como alternativa al clásico SQL.

Cuándo pensamos en bases de datos relacionales a nuestra mente suelen acudir los mismos nombres. En la parte comercial tenemos Oracle y Microsoft SQL Server. Del lado del software libre, tenemos opciones como Postgre SQL o MySQL. Aunque cada una tiene sus peculiaridades, para un desarrollador no es difícil elegir entre un sistema y otro. Al final todo son tablas, columnas, claves primarias, y sobre todo, consultas SQL. La decisión de cuál elegir, se basará en sus características y precio.

Pero si hablamos de bases de datos NoSQL, la cosa se complica. A día de hoy existen unos 150 sistemas de bases de datos NoSQL. Elegir uno de ellos puede ser muy difícil, ya que ninguno ha obtenido todavía la fama que sí han conseguido las bases de datos relacionales.

Pero el problema principal que encontramos, es que aunque todas se denominan NoSQL, en realidad hay diferentes tipos. Dependiendo de lo que necesitemos, deberemos decidirnos por una u otra.

**Ventajas y desventajas NoSQL.**

**Ventajas:**

* Escalabilidad
* Estás Base de Datos pueden distribuir la carga con diferentes host además, de poderse ampliar sus capacidades, también se evita tener configuraciones complejas con llaves primarias al no ser estas necesarias.
* Flexibilidad
* Como carecen de un esquema ya determinado, se puede moldear a las necesidades que vayan surgiendo.
* Sencillo de Administrar
* Los equipos pueden manejar las Base de Datos sin que el Administrador de la Base de Datos (BDA), tenga que interactuar mucho con ellas, esto quiere decir que, a los DBA se les reduce el trabajo.
* Código Abierto
* Normalmente el código abierto es soportado por una comunidad, por lo tanto lo hace muy seguro, además de ser buena alternativa al usar un software barato o gratuito y muy eficiente si a seguridad se refiere.
* Ajustable a diferentes proyectos
* Las Base de Datos con sistemas NoSQL, al conformar un grupo de diferentes clases de sistemas, pueden ser ajustados a diferentes proyectos según lo amerite la necesidad de este.
* Ajustes con la Nube
* Los servicios que están en la nube permiten que estos sistemas sean ampliables al ser estos manejados con máquinas virtuales.
* No hay cuellos de botella
* Las Base de Datos con Sistemas NoSQL no necesitan tener sentencias que sean transcritas y ejecutadas esto evita niveles de ejecución más complejos y no terminan en un punto de entrada común y único.

**Desventajas:**

* Incompatibilidad
* Cada Sistema tiene lo suyo y no tienen una guía definida por lo que estos sistemas solo pueden ofrecer soporte a nivel de fabricante y no de alguna administración más allá de este.
* Inmadurez
* Estos sistemas son muy jóvenes por lo que no llevan confianza, se requerirá más inversión y solo empresas con buen capital pueden resolver los problemas que se presenten.
* Mancha de Código Abierto
* Los Sistemas NoSQL de Código Abierto, pueden no tener credibilidad en soporte, y un alcance global.

**Métodos para implementar NoSQL**

Los sistemas de bases de datos NoSQL crecieron con las principales compañías de Internet, como Google, Amazon, Twitter y Facebook. Estas tenían que enfrentarse a desafíos con el tratamiento de datos que las tradicionales RDBMS no solucionaban. Con el crecimiento de la web en tiempo real existía una necesidad de proporcionar información procesada a partir de grandes volúmenes de datos que tenían unas estructuras horizontales más o menos similares. Estas compañías se dieron cuenta de que el rendimiento y sus propiedades de tiempo real eran más importantes que la coherencia, en la que las bases de datos relacionales tradicionales dedicaban una gran cantidad de tiempo de proceso.

Existen distintos tipos de bases de datos no relacionales, lo cuales son considerados NoSQL. Para su implementación, debemos tener en cuenta cuáles son nuestras necesidades en cuanto al manejo de nuestra información (datos), además del crecimiento a futuro de nuestra base de datos.

Algunos de los sistemas de bases de datos NoSQL más utilizados son los siguientes:

* Sistema de Bases de Datos Documentales.
* Sistema de Bases de Datos en Grafo.
* Sistema de Bases de Datos Clave/Valor.

**Sistema de Bases de Datos Documentales**

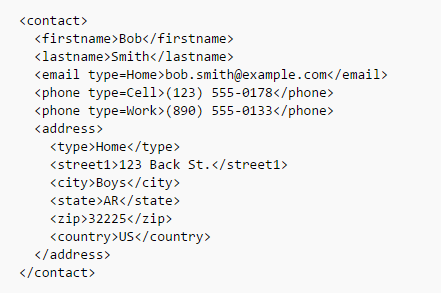
Conformada por un conjunto de programas que almacenan, recuperan y gestionan datos de documentos o datos de algún modo estructurados. Este sistema constituye una de las principales sub categorías dentro de las bases de datos NoSQL.

Este sistema en general, se basa en una definición abstracta de lo que es un documento.

Mientras cada implementación de base de datos orientada a documentos difiere en los detalles, en general todas ellas comparten el principio de que los documentos encapsulan y codifican datos o información siguiendo algún formato estándar. Entre las codificaciones usadas en la actualidad se encuentran XML, YAML, JSON y BSON, así como formatos binarios como PDF y documentos Microsoft Office (MS Word, Excel y demás).

Mientras cada implementación de base de datos orientada a documentos difiere en los detalles, en general todas ellas comparten el principio de que los documentos encapsulan y codifican datos o información siguiendo algún formato estándar. Entre las codificaciones usadas en la actualidad se encuentran XML, YAML, JSON y BSON, así como formatos binarios como PDF y documentos Microsoft Office (MS Word, Excel y demás).

Estos documentos contienen alguna información similar y otra diferente. Al contrario que una base de datos relacional en la que todos los registros deben tener los mismos atributos -que pueden quedar vacíos- , en un documento no quedan 'campos' vacíos. De este modo es posible añadir nueva información sin necesidad de establecer qué información queda excluida.



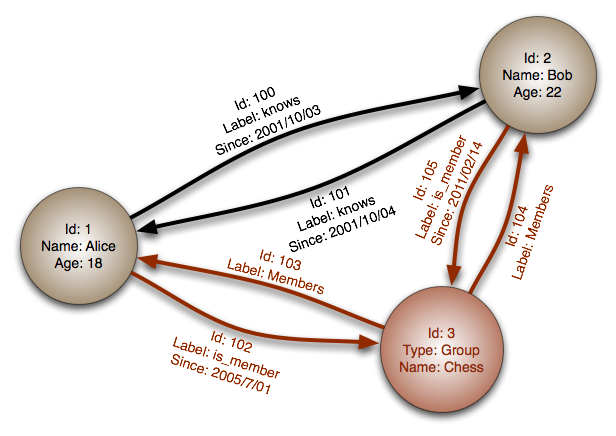
**Sistema de Bases de Datos en Grafo**

Las bases de datos orientadas a grafos (BDOG) representan la información como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar teoría de grafos para recorrer la base de datos ya que esta puede describir atributos de los nodos (entidades) y las aristas (relaciones).

Una base de datos orientada a grafos debe estar absolutamente normalizada, esto quiere decir que cada tabla tendría una sola columna y cada relación tan solo dos, con esto se consigue que cualquier cambio en la estructura de la información tenga un efecto tan solo local.

Las BDOG también ofrecen servicios nuevos o mejorados como:

* Consultas más amplias y no demarcadas por tablas, ejemplo “Muestre todas las tablas que posean un nombre Carlos”.
* No hay que definir un número determinado de atributos, esto quiere decir que una persona puede tener relacionados 4 nombres mientras que otra tan solo 2, esto sin desperdiciar espacio.
* Los registros también son de longitud variable, evitando tener que definir un tamaño y también posibles fallas en la base de datos.
* Se puede recorrer directamente la base de datos de forma jerárquica, obtener el nodo abuelo del nodo y viceversa.

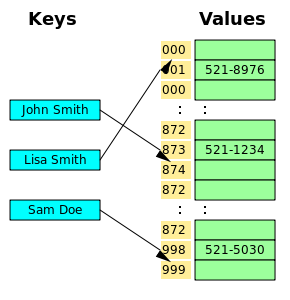


**Sistema de Base de Datos Clave/Valor**

Las bases de datos basadas en el modelo clave-valor son probablemente los sistema más sencillo utilizados para almacenar información dentro del ámbito de los noSQL. Estos sistemas son altamente funcionales y de muy alto rendimiento con una alta curva de escalabilidad dada la simplicidad de las estructuras que lo componen.

En un sistema clave-valor la información o valor es almacenado en relación directa con una clave única que identifica la información. Cada valor sin embargo puede estar estructurado según convenga ya sea en forma de hash o JSON (aunque no es el propósito de las bases de datos clave-valor). Las consultas a realizar sobre estos sistemas solo pueden ser ejecutadas en relación a la clave, es decir, no están permitidas las búsquedas por valor por lo que es mandatorio conocer la clave relacionada para poder extraer la información.

Este tipo de modelo de almacenamiento de información es ampliamente utilizado para almacenar información sobre la sesión de un usuario, opciones de configuración, video juegos y comercio electrónico entre otros ejemplos. Es sumamente útil para el almacenamiento de información temporal que deba persistir en el tiempo o a través procesos concatenados.



**¿Cuáles son las soluciones más sobresalientes de NoSQL?**

**¿Quiénes utilizan NoSQL?**

**Conclusión**

**Referencias bibliográficas**

**Bases de datos NoSQL. Elige la opción que mejor se adapte a tus necesidades:**

<http://www.genbetadev.com/bases-de-datos/bases-de-datos-nosql-elige-la-opcion-que-mejor-se-adapte-a-tus-necesidades>

**Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar:**

<http://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>

**NoSQL llegó para quedarse:**

<http://basesdedatosnosql.blogspot.com>

**Historia breve sobre NoSQL:**

<http://cs.lmu.edu/~ray/notes/nosql>

**¿Cuál es la diferencia entre RDBMS y NoSQL?**

<http://www.researchgate.net/post/What_is_the_difference_between_RDBMS_and_NoSQL2>

**Lista de bases de datos noSQL:**

[**http://nosql-database.org/**](http://nosql-database.org/)

**NoSQL:**

[**https://es.wikipedia.org/wiki/NoSQL**](https://es.wikipedia.org/wiki/NoSQL)

**Implementación de sistemas de bases de datos NoSQL**

[**http://quantummode.com/las-bases-de-datos-nosql-ii-el-modelo-de-datos/**](http://quantummode.com/las-bases-de-datos-nosql-ii-el-modelo-de-datos/)