

¿QUÉ CARACTERÍSTICAS TIENEN LOS ESQUEMAS NOSQL?

WHAT KINDS OF FEATURES HAVE THE SCHEMES NOSQL?

Deivi Robles*, Maury Sánchez*, Roy Serrano*, Bryan Adárraga* & Diana Heredia Vizcaíno**

drobles, msuarez38, rserrano5, badarraga {@unisimon.edu.co} {dianahv@unisimonbolivar.edu.co}

Universidad Simón Bolívar, Barranquilla-Colombia

Resumen | Actualmente existen muchos tipos de diferentes bases de datos, entre ellas las bases de datos relacionales, también se han diseñado numerosas arquitecturas para el manejo de distintas tipologías de datos. Desde 1970 el modelo relacional se ha implantado en casi todas las bases de datos; con el inicio de una nueva era los desarrolladores notaron que sus datos diferían de la estructura del común modelo relacional e implementaron nuevas ideas o arquitecturas, para que esos datos no se vieran afectados por el modelo relacional. Este trabajo pretende dar a conocer las principales características de los sistemas de bases de datos NoSQL, discutir modelos de representación de datos así como establecer sus ventajas y desventajas frente a otros modelos.

Palabras clave: | *NoSQL, Bases de datos, Modelo relacional*

Abstract | Currently there are many different types of databases, including relational databases give, but also numerous architectures are designed to handle different data types. Since 1970 the relational model has been implemented in almost all databases; with the start of a new era developers they noticed that their data differed from the structure of the common relational model and implemented new ideas and architectures for these data were not affected by the relational model. This paper seeks to highlight the main features of the systems of NoSQL databases, discuss models of data representation and establish their advantages and disadvantages compared to other models.

Key-words: | *NoSQL Databases, Relational Model*



Para referenciar este artículo (IEEE):

[N] D. Robles, M. Sánchez, R. Serrano, B. Adárraga & D. Heredia, “¿Qué características tienen los esquemas NoSQL?”, *Investigación y Desarrollo en TIC*, vol. 6, no. 1, pp. 40-44, 2015.

Artículo resultado de formación para la investigación

*Estudiante del programa de Ingeniería de Sistemas.

** *Tutora*, Profesora e investigadora del grupo INGEBIOCARIBE.

I. INTRODUCCIÓN

Este documento proporciona contenido sobre NoSQL, el cual se refiere a una nueva arquitectura de sistemas de bases de datos, generalmente distribuidos, que no requieren tablas con esquemas fijos, evitan el uso de la operación JOIN, típicamente escalan de forma horizontal y no siempre exponen sus datos mediante interfaces SQL. Es de notar que existen proveedores NoSQL tanto propietarios como basados en software libre. En algunos casos el término NoSQL incluso se utiliza con el significado de un sistema totalmente no relacional, pero en la práctica no es tan sencillo como a decir que los sistemas RDBMS y NoSQL son totalmente opuestos, ya que existen diferentes soluciones y bibliotecas intermedias (middleware) que adicionan y complementan los sistemas NoSQL con las características ACID de los sistemas RDBMS. Por lo general, en el ámbito académico a este tipo de bases de datos a se les refiere como almacenamiento estructurado, término que también incluye a las bases de datos.

Relacionales clásicas. Entre las características que más demandan las aplicaciones actuales y han incidido en la creación e incremento en el uso de las bases de datos NoSQL se o pueden enumerar las siguientes:

Gran cantidad de peticiones de lectura y escritura de forma concurrente. La propia complejidad de la lógica detrás del funcionamiento de las bases o a de datos relacionales, tiende a perder eficiencia en relación al crecimiento o de los datos. Esto hace difícil responder con poca latencia en el caso de aplicaciones que atienden un gran número de pedidos a la misma vez. La necesidad de almacenar y gestionar eficientemente cantidades masivas de datos. Aplicaciones multimedia, redes sociales o los buscadores, por solo citar algunos ejemplos, requieren almacenar y gestionar cantidades de datos continuamente crecientes sin que esto implique una pérdida de eficiencia. Estas son aplicaciones para los cuales los RDBMS en general no están a diseñados. Brindar un servicio alta escalabilidad y alta disponibilidad. Para la gestión de altos volúmenes de datos y para garantizar su disponibilidad, se hacen necesarios sistemas redundantes y fáciles de escalar.

Los sistemas NoSQL están diseñados para lograr fácilmente un escalado horizontal a diferencia de los clásicos sistemas relacionales que escalan a verticalmente. Disminuir costos de operación y de mantenimiento. La existencia de buenos proveedores NoSQL basados en Software Libre y de los bajos costes del hardware necesario para un escalado horizontal, convierten los sistemas NoSQL en una opción económicamente ventajosa.

Es por todo lo anterior, que el concepto NoSQL no se entiende solamente como una forma de llamar a los sistemas diferentes a los tradicionales, sino que se refieren a las ventajas que estos traen sobre los clásicos sistemas RDBMS a como una interpretación en abreviatura de “No solo SQL”.

A. Nomenclatura

- **SQL:** *Structured Query Language* - Lenguaje Estructurado de consultas.
- **NOSQL:** *Not Only Structured Query Language* - No solo SQL.
- **ACID:** *Atomicity, Consistency, Isolation and Durability* - Atomicidad, Consistencia, Aislamiento y Durabilidad
- **RDBMS:** *Relational Database Management System* - Sistema de Gestión de Bases de Datos Relacionales
- **API:** *Application Programming Interface* - Interfaz de programación de aplicaciones.
- **HTTP:** *HyperText Transfer Protocol* - Protocolo de transferencia de hipertexto.

II. DESARROLLO DE LOS OVA

Para realizar un estudio de los sistemas de bases de datos NoSQL es conveniente partir de los conocimientos y/o conceptos de los modelos de datos que con más frecuencia son utilizados [1].

A. Modelo de datos tradicionales

Los modelos de datos relacionales están basados en el álgebra relacional, que además tiene sus fundamentos en la lógica de predicados y la teoría de conjuntos [2].

La principal razón de esto es que se requieren implementar las reglas del algebra relacional para poder satisfacer a las operaciones necesarias para soportar las transacciones ACID. Por otro lado, los requerimientos que se plantean las bases de datos NoSQL no requieren satisfacer las operaciones ACID, por lo que no es necesario soportar directamente el álgebra relacional, los modelos de datos NoSQL más comunes son [1]:

1. **Modelos llave-valor**, este modelo de datos es muy sencillo, a cada llave se corresponde un valor. Aunque su estructura es muy simple, permite velocidades de consulta mayores que las bases de datos relacionales, los que lo hace muy útil para ser utilizados en bases de datos masivas y que requieran alta concurrencia, etc. Este modelo soporta bien las operaciones de consulta y medicación basadas en la llave primaria

2. **Modelos orientados a columnas**, este modelo se basa en la utilización del concepto de tabla, pero sin o las relaciones de asociación. Los datos son separados y almacenados por o columnas, cada columna es un índice, todos los datos en una columna tienen el mismo tipo y por lo general para el acceso concurrente se dividen las consultas de manera que las consultas para una misma tabla son realizados por un mismo proceso. En general, las ventajas de este modelo lo hacen más adecuado para aplicaciones que explotan la agregación y para a o crear almacenes de datos.

3. **Modelos de documentos**, el modelo orientado a documentos es muy parecido en estructura al modelo llave-valor, pero la principal diferencia está en el tipo de datos que no son simples sino semánticos, expresados por lo general en formato JSON o a XML. Otra diferencia es que los modelos orientados a documentos suelen utilizar índices secundarios y en el modelo llave-valor esto no es posible.

4. **Modelos basados en grafos**, en el modelo basado en grafos, la información se almacena partiéndola en o en fragmentos más básicos y estableciendo relaciones entre ellas. Este tipo de base de datos es capaz de obtener rendimientos altísimos en consultas sobre información relacionada, como los contactos de un usuario en una red social. En un modelo relacional estas relaciones implicarían una enorme cantidad de operaciones *Join* que dependiendo del volumen y la infraestructura, simplemente son muy ineficientes en términos por el enorme consumo de tiempo.

B. Clasificaciones según los modelos

Teniendo en cuenta los anteriores modelos de datos, una clasificación básica o a de los principales sistemas NoSQL a los que se refieren son los siguientes [1]:

- **Modelo llave-valor:** Azure Table Storage, MEMBASE, Riak, Redis, Chordless, GenieDB, Scalaris, Tokyo Cabinet / Tyrant, GT.M, Keyspace, Berkeley DB, MemcacheDB, HamsterDB, Faircom C-Tree, Mnesia, LightCloud, Pincaster, Hibari, Scalify
- **Similares al modelo llave-valor:** Amazon Dynamo, Voldemort, Dynamite, KAI, SubRecord, Mo8onDb, Dovetaildb
- **Modelo orientados a columnas:** Hadoop, HBase, Cassandra, Hypertable, Cloudata, Amazon, SimpleDB, SciDB
- **Modelo de documentos:** CouchDB, MongoDB, Terrastore, ThruDB, OrientDB, RavenDB, Citrusleaf, SisoDB, CloudKit, Perservere, Jackrabbit

- **Modelo basado en grafos:** Neo4J, Infinite Graph, Sones, InfoGrid, HyperGraphDB, Trinity, AllegroGraph, Bigdata, DEX, OpenLink Virtuoso, VertexDB, FlockDB, Java Universal Network / Graph Framework, Sesame, Filament, OWLim, NetworkX, iGraph.

III. ANTECEDENTES Y ESTADO ACTUAL

Para incursionar en el concepto de NoSQL, a continuación se presentan algunos referentes donde se describe que el término NoSQL fue acuñado en 1998 por Carlo Strozzi y retomado en 2009 por Eric Evans, quien sugiere mejor referirse a esta familia de Bases de Datos de nueva generación como "Big Data" mientras que considera ahora que NoREL es un mejor nombre para dicho tipo de bases de datos. NoSQL – "*not only SQL*" es una categoría general de sistemas de gestión de bases de datos que difiere de los Sistemas Manejadores de Bases de Datos – RDBMS, en aspectos como [3]:

- No tienen esquemas, no permiten JOIN, no intentan garantizar ACID y escalan horizontalmente.
- Tanto las bases de datos NoSQL como las relacionales son tipos de Almacenamiento Estructurado.

La principal diferencia radica en cómo guardan los datos (por ejemplo, almacenamiento de un recibo):

- En una RDBMS, la información a persistir se tendría que partir en diferentes tablas y luego, en la capa de aplicación, usar un lenguaje de programación que realice el respectivo mapeo para transformar estos datos en objetos de la vida real [4].

En NoSQL, simplemente se guarda el recibo. Dado que NoSQL es libre de esquemas, no se deben diseñar tablas y su estructura por adelantado. Un primer intento para conceptualizar el término NoSQL se evidenció en la publicación [5], donde se definió como una nueva generación de sistemas de bases de datos que tenía al menos una de las siguientes propiedades:

- El modelo de datos subyacente no es relacional.
- Los sistemas están diseñados desde el principio como escalable horizontalmente y verticalmente.
- El sistema es de código abierto.
- El sistema es esquema-libre o sólo tiene restricciones de esquema suaves.
- Debido a la arquitectura distribuida, el sistema admite un método de replicación de datos simple.
- El sistema proporciona una API sencilla.

- El sistema generalmente utiliza un modelo de consistencia diferente.

El término NoSQL se debe enfocar como un conjunto de características para diferentes almacenes de datos, sin un dominio homogéneo [3].

IV. DESARROLLO TEÓRICO

A. Manejo de datos geoespaciales

Algunas soluciones NoSQL que soportan datos geográficos pueden ser nativas o con extensiones; otros, aunque inicialmente no están diseñados para soportar datos geográficos, pueden ser mejorados para soportarlos. Dentro de los proyectos que soportan datos espaciales se encuentran:

- **Neo4j:** es un almacén de datos NoSQL de entorno gráficos utilizado en la consulta de datos espaciales vectoriales con integración gráfica que interactúa con aplicaciones Java o cualquier otra biblioteca, y que muestra sus datos como un gráfico. Al ser gráfico, los beneficios sobre consultas son mayores sobre recorridos o regiones locales, análisis trayectorias cortas y consultas de conectividad entre ubicaciones [7].
- **CouchDB:** es una base de datos NoSQL que emplea JSON para almacenar los datos, como se muestra en la figura; JavaScript como lenguaje de consulta por medio de Map/Reduce y HTTP como API. [8] Una de sus características más peculiares es la facilidad con la que permite hacer replications. Es un almacén de datos orientado a documentos licenciado por Apache y escrito en Erlang, su objetivo es la consistencia de datos y facilidad de uso. CouchDB maneja una extensión espacial llamada GeoCouch que soporta datos espaciales, con B-tree para búsquedas de clave/valor y R-tree para consultas espaciales [9].
- **MongoDB:** es también un almacén de datos con orientación por documento, con licencia AGPL. Es reconocida como la base de datos NoSQL líder, con una mayor productividad y una mejor experiencia de desarrollo.

B. Aqua Data Studio 14 soporta NoSQL, MongoDB y Cassandra

AquaFold, Inc. ha lanzado Aqua Data Studio 14, una actualización importante de su software universal de gestión y consulta de bases de datos, que le permite a los usuarios poder diseñar y gestionar la clase más reciente de bases de datos, incluidas las bases de datos relacionales, incrustadas, NoSQL, e inclusive las alojadas en la nube. Las

bases de datos NoSQL y en la nube han crecido en importancia ya que las nuevas y las grandes empresas enfrentan el desafío de trabajar con conjuntos masivos de datos.

“Aqua Data Studio es el primero que se comercializa con una interfaz gráfica de usuario con múltiples bases de datos que también es compatible con NoSQL, con las bases de datos en la nube y las basadas en Hadoop”, señala el Director Ejecutivo de AquaFold, Niels Gron. “Los clientes reconocen el enorme valor de estas bases de datos especializadas que proporcionan soluciones altamente escalables y de bajo costo para sus análisis de negocio” [16].

La incorporación del soporte para NoSQL obligó al equipo de desarrollo de AquaFold a desarrollar formas creativas de ofrecer una interfaz de usuario relacional “SQL” para las bases de datos que explícitamente no lo soportan. El equipo diseñó su propio idioma MongoDB específico para MongoDB, además un shell de línea de comandos que incorpora un shell JavaScript MongoDB interactivo directamente en el Aqua Data Studio. El equipo también trabajó para ofrecer el conjunto completo de herramientas de Aqua Data Studio a los usuarios de Apache Hive y de la base de datos distribuida y de fuente abierta Apache Cassandra.

La conexión de la base de datos relacional como un servicio de Microsoft, la base de datos SQL (Azure), ahora también es posible en la versión 14 de Aqua Data Studio utilizando las mismas herramientas que los usuarios esperan para cuando se conectan al Servidor SQL de Microsoft.

V. CONCLUSIONES

Luego de haber concluido este trabajo de investigación sobre bases de datos NoSQL fueron muchos los esfuerzos y conocimientos adquiridos durante dicha elaboración. Algunos de los aspectos aprendidos y de gran peso es la definición de Base de datos NoSQL, requerimientos, ventajas y características de estas, donde se puede afirmar que cuando se habla de bases de datos NoSQL no se refiere únicamente a un tipo de bases de datos sino a diferentes soluciones dadas para almacenar datos cuando las bases de datos relacionales generan problemas. Las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación, no imponen una estructura de datos en forma de tablas y relaciones entre ellas, en ese sentido son más flexibles, ya que suelen permitir almacenar información en otros formatos como clave-valor, Mapeo de Columnas, Documentos o Grafos.

Otro aspecto importante de las bases de datos NoSQL, es que es visto por muchos, como el futuro de los sistemas de bases de datos, pero hay movimientos importantes. Por ejemplo el notable afán de Facebook por crear motores de bases de datos tales como Cassandra y RocksDB, y el uso de NoSQL en otras redes sociales como Twitter y LinkedIn, que indican que todo apunta hacia servicios que usen repositorios datos con NoSQL únicamente. Pero hay dos problemas que retrasan este hecho. El primero es la resistencia al cambio pues se piensa que los motores de bases de datos no pueden resolver muchos problemas de la vida real (cabe señalar que la mayoría de los proyectos no usan ni el 50% de las bondades que ofrece SQL), un ejemplo concreto radica en los nuevos desarrollos, muchos piensan que con NoSQL no se garantiza la integridad de los datos, o la consistencia, pero hay métodos para poder hacer esto con bases de datos documentales por nombrar alguna.

En la investigación se pudo concretar que las bases de datos NoSQL, no son una amenaza para las bases de datos relacionales, sino un complemento a ellas, ya que muchos sistemas utilizan ambos tipos de bases de datos y pueden co-relacionarse de manera efectiva, aunque algunos crean más en un tipo que en el otro, lo cierto es que ambas pueden usarse para fines diversos, y así cubrir el terreno del mundo del manejo de las bases de datos.

VI. REFERENCIAS

- [1] Wikipedia, "Wikipedia," Wikipedia Inc. , 09 05 2014. [Online]. Available: http://es.wikipedia.org/wiki/Google_Acad%C3%A9mico. [Accessed 03 11 2014].
- [2] Wikipedia, "Wikipedia," Wikipedia Inc., 09 03 2013. [Online]. Available: <http://es.wikipedia.org/wiki/ProQuest>. [Accessed 03 11 2014].
- [3] Wikipedia, "Wikipedia," 26 08 2014. [Online]. Available: es.wikipedia.org/wiki/Modelo_relacional
- [4] M. Stonebraker, "In search of database consistency," ACM, p. 8–9, 2010.
- [5] J. Scholz, "Coping With dynamic, Unstructured Data Sets - NoSQL a Buzzword or a Savior?," 2011.
- [6] A. H. S. Steiniger, "Free and open source GIS Software for Building a Spatial Data Infrastructure," 2012.
- [7] J. L. N. S. J. Anderson, "The Definitive Guide," 2010.
- [8] D. L. d. Ipiña, "NoSQL: Introducción a las bases de datos no estructuradas," 2013.
- [9] HispaBigData, "Base de Datos NoSQL," 2013.
- [10] S. Edlich, "Nosql, your ultimate guide to the non - relational universe!," nosql-database.org, pp. 2-8, 2011.
- [11] O. G. F. G. T. D. S. C. De Souza, "Using GC services to interoperate spatial data Stored in SQL and NoSQL Databases," 2011.
- [12] E. Brewer, "Pushing the cap: Strategies for consistency and availability," Computer, p. 23–29, 2012.
- [13] E. Brewer, "A certain freedom: thoughts on the cap theorem," in Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing," ACM, p. 335–335, 2010.
- [14] B. Baas, "NoSQL spatial: Neo4j versus PostGIS. GIMA - Geographical Information Management And Applications," 2012.
- [15] S. Villegas-Giraldo, "Análisis de las bases de datos NOSQL como alternativa a las bases de datos SQL," Vitrina, 17 Abril 2014. [Online]. Available: <http://repository.eia.edu.co/handle/11190/411>. [Accessed 25 10 2014].
- [16] G. Gron (2013). Aqua Data Studio 14 agrega soporte para las bases de datos de NoSQL, MongoDB y Cassandra," BusinessWire, p. 14, 2013.