

# SQL vs NoSQL

Nilson Felix, Laura Atencio  
Andree Ludwerd, Velasco Sucauca

*Universidad Privada de Tacna \Facultad de Ingenieria \Escuela Profesional de Ingenieria de Sistemas*

## Resumen

Los lenguajes de programación se mantienen en constante cambio, desde lenguaje máquina, assembler, lenguajes de segunda y tercera generación hasta llegar a una cuarta, la cual comúnmente fue utilizada para poder administrar diferentes **DBMS (Database Management System)**, este lenguaje fue llamado; **SQL (Structured Query Language)**. Durante la maduración de este lenguaje surgieron varias soluciones de software que permitían poder realizar operaciones de consultas, grabar o insertar información, actualizar, eliminar, etc. Fueron surgiendo varios programas que con el tiempo fueron denominados **RDBMS (Sistema de Gestor de Base de Datos Relacionales ó Relational DataBase Management System)**.

El SQL fue uno de los lenguajes de programación estructurada más aceptada, pero siempre van a existir limitantes, en las cuales la tecnología tiene que ir en constante evolución para buscar respuestas a esas limitantes, por eso el surgimiento de soluciones de software basados en sistemas **NoSQL**, el cual significa "Not Only SQL" por sus siglas en inglés, lo que busca es mejorar temas de rendimiento sobre las bases de datos relacionales pero también dentro de las ventajas conllevan algunas desventajas las cuales trataremos de discutir durante el desarrollo de este documento.

**Palabras clave:** virtualizacion, contenedores, herramientas, simulacion, procesos, recursos.

## Abstract

The programming languages are in constant change, from machine language, assembler, second generation and third generation languages. SQL (structured query language). During the maturation of this language emerged several software solutions that allowed to perform operations of consultations, record or insert information, update, delete, etc. Several programs were created that over time were also called RDBMS (Relational Database Management System (Relational Database Management System)). SQL was one of the most accepted structured programming languages, but there was always a way to limit itself, in the network. , which means "Not only SQL" for its acronym in English, which seeks to improve performance issues on relational databases but also within the advantages that carry some disadvantages that deal with the subject of this report.

**Keywords:** virtualization, containers, tools, simulation, processes, resources.

## I. INTRODUCCIÓN

Las bases de datos no son para nada ajenas a las innovaciones y nuevas tendencias que se desarrollan a su alrededor. Es por ello que a las tradicionales bases SQL les salió ya hace ya un tiempo un competidor que cada vez tiene más fuerza, las bases NoSQL. Muchos desarrolladores han optado por migrar sus proyectos y trabajos a este modelo, pero para hacerlo es conveniente primero saber las diferencias entre ambas y también sus principales tecnologías, de forma que puedas tener información precisa sobre cuál te conviene más en cada caso.

## II. OBJETIVOS

### A. General:

- Determinar las características diferenciales entre SQL y NoSQL.

### B. Específicos:

- Comparar el concepto de bases de datos SQL y NoSQL, además de presentar sus ventajas y compararlos con otros sistemas de bases de datos, como los relacionales.

## III. MARCO TEÓRICO

SQL, bases de datos relacionales

Las bases de datos relacionales (SQL) son el tipo que más tiempo lleva en el mercado de la tecnología y son por tanto las utilizadas en mayor medida tradicionalmente. Su composición está hecha con bases de datos llenas de tablas con filas que contienen campos estructurados. No se trata precisamente de un tipo de base de datos muy flexible, pero tiene a favor su gran soporte y el enorme desarrollo en herramientas debido a todo su bagaje histórico. De hecho, es una tecnología muy reconocida en comparación con las bases NoSQL que al ser más novedosas no tienen tanto alcance todavía. En cambio, las bases de datos SQL necesitan más recursos como norma general, ya que cuanto más compleja sea la base más procesamiento necesitará.

El mayor inconveniente de las SQL es probablemente el referido a la escalabilidad, que es precisamente el que permitió el desarrollo de alternativas que desembocaron en las redes NoSQL, ya que grandes empresas que manejan datos a gran escala con necesidad de mucha infraestructura sufrían con este problema. A partir de ahí se desarrollaron las primeras NoSQL cerradas creadas por empresas para su uso interno, y finalmente los sistemas de código abierto. Algunas de sus tecnologías más conocidas son Hypertable, Cassandra, MongoDB, DynamoDB o Redis, mientras que Oracle, Microsoft SQL Server, SQLite o MySQL son cuatro de las más utilizadas dentro de las bases de datos SQL.

NoSQL, base de datos no relacional

La gran diferencia entre los dos tipos de bases de datos es la estructuración, que en el caso de las NoSQL se trata de una forma de almacenamiento no estructurado, todo lo contrario a sus predecesoras. En este caso, son de bases de datos sin una tabla fija como las que sí se encuentran en las bases de datos relacionales, lo que permite una alta escalabilidad en ellas. Además, es abierta y por lo tanto flexible a diferentes tipos de datos y no necesita tantos recursos para ejecutarse; de hecho, el

hardware necesario no es de un coste muy elevado. En este sentido, también se deja notar el menor coste a la hora de la expansión, ya que no necesitan la evolución en hardware del equipo, sino que basta con hacer un escalado horizontal, con más máquinas en las que distribuir la carga completa.

Por contra, un par de desventajas de las bases NoSQL, que no son perfectas, son que los desarrollos, al implementar su propio código en pos de la fiabilidad y coherencia, es decir, al no existir tanta estandarización, se limita el número de aplicaciones seguras para realizar transacciones, y por otro lado, tampoco es una virtud el hecho de que sea incompatible con ellas llevar a cabo consultas SQL, lo que hace necesario un lenguaje de consulta manual que ralentiza los procesos de este tipo concreto.

En definitiva, lo más importante que debes saber es que las NoSQL no son un sustituto de las SQL, sino que son una alternativa que ofrece otras posibilidades, lo que las convierte en más interesantes para determinados casos como por ejemplo proyectos que requieren una alta escalabilidad en los recursos son escasos y la integridad de los datos no es lo más importante, como sí ocurre en cambio en aplicaciones especializadas por ejemplo en transacciones bancarias.

## Ventajas y Desventajas de SQL:

### Ventajas:

- **Madurez:** Dado que tiene ya muchos años de madurez y aceptación por la comunidad de desarrolladores existe una gran variedad y cantidad de información para poder realizar cualquier tipo de desarrollo o extracción de información, lo cual esto ayuda increíblemente en la mejora de tiempos de entrega de cualquier proyecto de software.
- **Atomicidad:** En las operaciones e información, esto quiere decir que cualquier operación realizada en la base de datos, garantiza que si a la mitad de cualquier operación de base de datos, surgió algún tipo de problema, la información no se completa, o se realiza al 100% o no se realiza nada.
- **Estándares bien definidos:** Por ejemplo, la creación de tablas, el insertar, eliminar y actualizar información, consultas, se escriben bajo la misma sintaxis, basados en el estándar de SQL.
- **Sencillez en la escritura:** Su principal aceptación, es su sencillez de escritura ya que se asemeja mucho al lenguaje humano, la comprensión de las operaciones que se programen puede ser interpretadas o escritas por personas que no tengan grandes conocimientos de informática.

### Desventajas:

- **Crecimiento:** Cuando estas bases de datos tienden a crecer demasiado en el almacenamiento y el mantenimiento es sumamente difícil y costoso, suelen presentar fallas en tiempo de respuesta.
- **Cambios en la estructura:** En muchas ocasiones, los negocios necesitan realizar cambios, tanto en sus operaciones como en los sistemas de informática, entonces la base de datos puede verse afectada a cambios, si el

diagrama de Entidad Relación no lo soporta, entonces esto implica, tener que realizar una modificación en la estructura de la base de datos y posiblemente, detener el sistema por un tiempo moderado hasta terminar el proceso, esto puede requerir mucho tiempo en el sistema y tal vez conocimientos avanzados en la persona designada para realizar estos cambios.

- **Elección del mas adecuado:** En el mercado de RDBMS, se pueden encontrar servicios gratuitos o comprados, dependiendo de varios factores, como pueden ser, cantidad de información, tiempo de disponibilidad, fuerza de operación, etc., Cada solución de RDBMS que se presente en el mercado ofrece también ventajas y desventajas, no precisamente lo mas caro vaya a ser la solución a todo.
- **Complejidad en la instalación:** Algunos RDBMS dependen del sistema operativo donde se vayan a instalar, no garantizan el buen funcionamiento si no cumplen con los requerimientos mínimos de instalación.

## Ventajas y desventajas de NoSQL

### Ventajas

- **Versatilidad:** La principal ventaja por la cual esta nueva tecnología difiere de las demás soluciones de bases de datos es la versatilidad que ofrece a crecimientos o cambios sobre la forma como almacena la información, si fuera necesario agregar un nuevo campo sobre una “colección” (en base de datos relacional se definen tablas), dado que se basa sobre una notación ligera de intercambio de documentos baso en notación **JSON(Java Script Object Notation)**, simplemente se agregan sobre el documento y el sistema sigue operando sin agregar configuraciones extras.
- **Crecimiento Horizontal:** Soportan una escalabilidad descentralizada, es decir, soportan estructuras distribuidas, si durante la operación se ve que el desempeño de los servidores tienden a bajar, se instalan nuevos nodos operativos para que balanceen la carga de trabajo, a esto le llaman crecimiento horizontal.
- **Disponibilidad de Recursos:** No se requieren servidores con una gran cantidad de recursos disponible para operar, pueden empezar a operar con bajos recursos e ir creciendo dependiendo de las necesidades sin tener que detener los servicios de operación.
- **Optimización:** Los sistemas NoSQL tienen un algoritmo interno para reescribir las consultas escritas por los usuarios o las aplicaciones programadas, esto con el fin de no sobrecargar el rendimiento de los servidores y mantener un nivel optimo en las operaciones.

### Desventajas

- **Atomicidad:** No todas las bases de datos contienen la característica de la atomicidad en la información, esto quiere decir, que la información en ocasiones no es consistente, puede ser diferente en cada uno de los nodos replicas que se puedan configurar en la arquitectura de base de datos.

- **Documentación del Software:** Dado que NoSQL, es relativamente nuevo, las operaciones pueden ser limitadas y se requiera de conocimientos avanzados con el uso de la herramienta y las personas que se encuentran realizando estos desarrollos en el software tengan que invertir mas tiempo en los desarrollos.
- **Estándares en el lenguaje:** No se tiene un estándar definido entre los diferentes motores que ofrecen este servicio, es decir, por ejemplo: DB2 para poder insertar información sobre su base de datos, el manejo de los objetos JSON no es el mismo como se utiliza en MondosDB y con ellos la diversidad de conocimientos que se tiene que tener dependiendo de la solución NoSQL se vaya a utilizar.
- **Herramientas GUI(Graphical User Interface):** Las herramientas que ofrecen para la administración de estas herramientas, suelen tener acceso por consola, no tienen una interfaz gráfica, se requiere conocimiento amplio de las instrucciones a utilizar para su mantenimiento.

## Factores a considerar al seleccionar una base de datos SQL o NoSQL

### Estructura de datos

El primer y principal factor para tomar la decisión de SQL frente a NoSQL es cómo se ven sus datos.

Si sus datos están estructurados principalmente, una base de datos SQL es probablemente la opción correcta.

Una base de datos SQL es ideal para sistemas orientados a transacciones, como herramientas de administración de relaciones con clientes, software de contabilidad y plataformas de comercio electrónico. Cada fila en una base de datos SQL es una entidad distinta (por ejemplo, un cliente), y cada columna es un atributo que describe esa entidad (por ejemplo, dirección, título del trabajo, artículo comprado, etc.).

Debido a estas relaciones distintas y estructuradas entre filas y columnas en una tabla, las bases de datos SQL son mejores cuando necesita el cumplimiento con ACID. ACID significa:

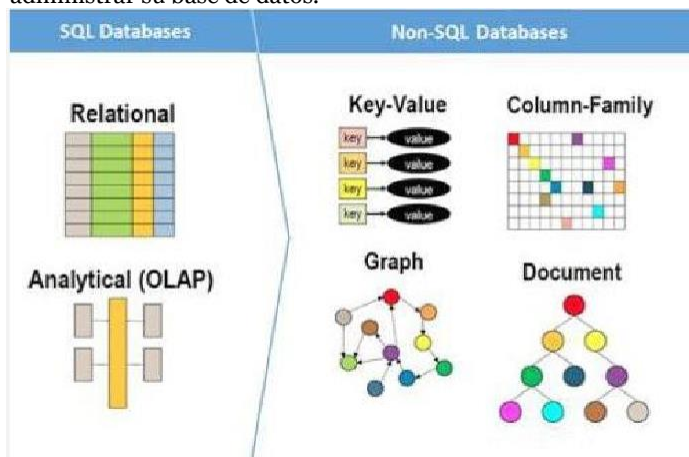
- **Atomicidad:** cada transacción tiene éxito o se retrotrae completamente.
- **Coherencia:** los datos escritos en una base de datos deben ser válidos de acuerdo con todas las reglas definidas.
- **Aislamiento:** cuando las transacciones se ejecutan simultáneamente, no se enfrentan entre sí, y actúan como si se estuvieran ejecutando de forma secuencial.
- **Durabilidad:** una vez que una transacción se ha comprometido con la base de datos, se considera permanente, incluso en el caso de una falla del sistema.

El cumplimiento con ACID protege la integridad de sus datos al definir exactamente qué es una transacción y cómo interactúa con su base de datos. Evita que las tablas de la base de datos se desincronicen, lo cual es muy importante para las transacciones financieras. El cumplimiento con ACID garantiza la validez de las transacciones incluso ante errores, fallas tecnológicas, eventos desastrosos y más.

Si sus datos están muy estructurados y el cumplimiento de ACID es una necesidad, SQL es una excelente opción.

Por otro lado, si sus requisitos de datos no están claros o si sus datos no están estructurados, NoSQL puede ser su mejor apuesta.

Los datos que almacena en una base de datos NoSQL no necesitan un esquema predefinido como lo hace para una base de datos SQL. Más bien, los datos pueden ser almacenes de columnas, orientados a documentos, basados en gráficos o pares de valores clave. Esto proporciona mucha más flexibilidad y menos planificación por adelantado al administrar su base de datos.



Con NoSQL, puedes:

- Crea documentos sin definir cuidadosamente su estructura por adelantado.
- Agregue campos a su base de datos sin cambiar los campos de documentos existentes
- Almacena documentos que tienen su propia estructura única.
- Tener múltiples bases de datos con diferentes estructuras y sintaxis.

Una base de datos NoSQL es mucho mejor para almacenar datos como contenido de artículos, publicaciones en redes sociales, datos de sensores y otros tipos de datos no estructurados que no encajan perfectamente en una tabla. Las bases de datos NoSQL se crearon teniendo en cuenta la flexibilidad y la escalabilidad, y siguen el modelo de consistencia BASE, que significa:

- Disponibilidad básica
- Esto significa que, si bien la base de datos garantiza la disponibilidad de los datos, la base de datos puede no obtener los datos solicitados o los datos pueden estar en un estado cambiante o inconsistente.
- Estado blando
- El estado de la base de datos puede estar cambiando con el tiempo.
- Consistencia eventual
- La base de datos eventualmente se volverá consistente y los datos se propagarán a cualquier lugar en algún momento en el futuro.

El modelo BASE fue construido para la máxima flexibilidad. Pero en realidad hay algunas bases de datos NoSQL que son compatibles con ACID. Puede leer más sobre esto en la sección "[La convergencia de SQL y NoSQL](#)".

La estructura de sus datos es el factor más importante a la hora de decidir si usar una base de datos SQL o NoSQL, así que piense mucho en esto antes de tomar una decisión.

### **Posibilidad de consultar datos**

El siguiente factor a considerar es la frecuencia con la que consultará sus datos, la rapidez con la que necesita ejecutar las consultas y quién será el responsable de ejecutarlas.

Debido a que sus datos están bien estructurados y organizados, es muy eficiente consultarlos con una base de datos SQL.

SQL es un lenguaje de programación popular que ha existido durante más de 45 años, por lo que es extremadamente maduro y conocido. Ejecuta eficientemente las consultas y recupera y edita los datos rápidamente. Es muy ligero y declarativo, y por lo tanto es fácil de aprender. Por lo tanto, las consultas pueden ser realizadas por personal menos técnico como analistas de negocios y comercializadores.

Una base de datos NoSQL proporciona un montón de flexibilidad en los tipos de datos que puede almacenar, pero debido a las diferencias potencialmente grandes en las estructuras de datos, las consultas no son tan eficientes como con una base de datos SQL.

Cuando se estaba construyendo la tecnología de base de datos NoSQL, los desarrolladores se centraron en la escalabilidad y la flexibilidad, no en la eficiencia de las consultas.

Por lo tanto, para ejecutar consultas NoSQL, deberá realizar un procesamiento adicional en los datos. Dependiendo de la base de datos NoSQL que esté utilizando, es posible que deba implementar algún nivel de MapReduce. Muchos desarrolladores crean funcionalidades de consulta en la capa de aplicación, en lugar de preocuparse por ello en la capa de base de datos. Ha habido algunos intentos de estandarizar las consultas NoSQL, como XQuery o JSONiq, pero estas herramientas no han sido ampliamente adoptadas.

La consulta de bases de datos NoSQL generalmente requiere desarrolladores o científicos de datos, que serán más costosos y menos eficientes.

¿Con qué frecuencia consultará sus datos y quién ejecutará estas consultas? Las respuestas a estas preguntas afectarán su decisión de SQL o NoSQL.

### **Escalada**

Las bases de datos SQL y NoSQL se escalan de manera diferente, por lo que tendrá que pensar cómo crecerá su conjunto de datos en el futuro.

Las bases de datos SQL se escalan verticalmente, lo que significa que tendrá que aumentar la capacidad de un solo servidor (aumentando la CPU, RAM o SSD) para escalar su base de datos. Las bases de datos SQL fueron diseñadas para ejecutarse en un solo servidor para mantener la integridad de los datos, por lo que no son fáciles de escalar.

Las bases de datos NoSQL se escalan horizontalmente, lo que significa que puede agregar más servidores para potenciar su base de datos en crecimiento. Esta es una gran ventaja que NoSQL tiene sobre SQL.

La capacidad de las bases de datos NoSQL para escalar horizontalmente tiene que ver con la falta de estructura de los datos. Debido a que NoSQL requiere mucha menos estructura que SQL, cada objeto almacenado es bastante independiente e independiente. De este modo, los objetos se pueden almacenar fácilmente en varios servidores sin tener que estar vinculados. Este no es el caso de SQL, donde cada fila y columna de la tabla debe estar relacionada.

Una analogía para la escala vertical y horizontal es un pastel de bodas. Con SQL, puede alimentar a más personas agregando más capas al pastel de bodas. Con NoSQL, puedes hacer un montón de pastelitos de boda. Yum

## SQL vs NoSQL, estilo pastel de boda

A medida que su negocio crezca, también lo hará su base de datos. Así que asegúrese de tener en cuenta sus necesidades de escalado.

### La convergencia de SQL y NoSQL.

Las bases de datos SQL y NoSQL tienen sus pros y sus contras. Como tal, ha habido un movimiento para tomar las mejores características de ambos tipos de bases de datos e integrarlas para que los usuarios puedan realizar lo mejor de ambos mundos.

Por ejemplo, MySQL, la base de datos relacional de código abierto más popular, ofrece MySQL Document Store. Esto proporciona la estructura de una base de datos MySQL combinada con la flexibilidad y la alta disponibilidad de NoSQL, sin tener que implementar una base de datos NoSQL separada. MongoDB, una de las bases de datos NoSQL más populares, ofrece transacciones ACID de varios documentos.

La base de datos NoSQL administrada de AWS, DynamoDB, también proporciona funcionalidad de transacción compatible con ACID.

Y con la sencilla configuración de la base de datos que ofrecen los proveedores de servicios en la nube, tiene la capacidad de utilizar bases de datos SQL y NoSQL en su arquitectura de datos en la nube para satisfacer sus necesidades de almacenamiento de datos.

Ahora tiene mucha más flexibilidad, independientemente de si elige una base de datos SQL o NoSQL, y seguramente habrá opciones más flexibles en el futuro.

### Opciones de base de datos

Independientemente de si va con una base de datos SQL o NoSQL (¡o con ambas!), Hay muchas opciones para elegir. Las ofertas de bases de datos SQL en las instalaciones incluye

- MySQL - como se mencionó anteriormente, la base de datos relacional de código abierto más popular
- Microsoft SQL Server - la versión empresarial de SQL de Microsoft
- PostgreSQL - y base de datos de fuente abierta de nivel empresarial enfocada en la extensibilidad
- Oracle - Opción SQL de servicio completo (y costoso)
- MariaDB: una versión mejorada de MySQL, creada por los desarrolladores originales de MySQL
- Y muchos más  
Las principales plataformas de servicios en la nube tienen sus propias opciones de SQL:
- AWS tiene:
  - RDS, su base de datos SQL estándar en la nube
  - Aurora, que se centra en aumentar el rendimiento y la escalabilidad
- Microsoft Azure tiene:
  - Base de datos SQL de Azure, su base de datos gestionada como servicio
  - Base de datos de Azure para MySQL, PostgreSQL y MariaDB

- Google Cloud Platform (GCP) tiene:
  - Cloud SQL, que puede usar para MySQL y PostgreSQL
  - Cloud Spanner, que combina elementos de SQL y NoSQL. Las opciones de base de datos NoSQL en las instalaciones incluyen:
    - MongoDB - por mucho la base de datos NoSQL más popular
    - Redis: una base de datos de código-valor de código abierto, distribuida, en memoria, que es súper rápida
    - Cassandra: base de datos NoSQL de código abierto y gratuita creada por Facebook que se centra en la escalabilidad y la alta disponibilidad
  - Muchos otros  
Los proveedores de servicios en la nube también ofrecen muchas opciones NoSQL:
- AWS tiene:
  - DynamoDB, su base de datos NoSQL gestionada.
  - DocumentDB, una base de datos rápida, escalable y altamente disponible compatible con MongoDB
- Microsoft Azure ofrece:
  - CosmosDB, su base de datos multimodelo distribuida globalmente.
- Google Cloud tiene:
  - Bigtable, su servicio de base de datos de columna ancha NoSQL.
  - Cloud Datastore, su servicio de base de datos de documentos NoSQL
  - Cloud Firestore, una base de datos de documentos NoSQL nativa de la nube que ayuda a almacenar y consultar datos de aplicaciones

No hay escasez de opciones de base de datos para elegir!

## Conclusiones:

- Siempre es bueno evaluar las diferentes alternativas que ofrece el mundo del software, para poder dar solución a muchas de los requerimientos que se presentan en el medio, pero siempre hay que considerar que cualquier solución de software que se seleccione va a tener sus ventajas y desventajas al momento de su desarrollo como también en la operación.
- La aparición del nuevo NoSQL es relativamente nueva, todavía le queda mucho tiempo para poder afirmar si es la solución a muchos de los problemas que hoy en día se presentan en el mundo de la base de datos, una de las ventajas muy importantes que ofrece NoSQL es que puede manejar grandes cantidades de información, pudiendo devolver el resultado de una consulta en cuestión de segundo, cuando un manejador tradicional RDBMS puede tardar 1000 veces mas el tiempo que el anterior haya resuelto
- Antes de tomar la decisión de migrar una base de datos relacional a una no relacional hay que considerar todos estos puntos ventajas y desventajas para no caer en un punto del cual ya no se puede regresar y no tener pérdidas de información importantes.

## Referencias:

- [1] Acens. Bases de datos nosql.qué son y tipos que nos podemos encontrar. Accessed: 2019-06-10.
- [2] Ana cyv. Existen varios tipos para almacenar información en base de datos nosql. Accessed: 2019-06-10.
- [3] GeeksforGeeks. Difference between sql and nosql. Accessed: 2019-06-10.
- [4] Marcela Sena. Como pasar de sql a nosql sin sufrir. Accessed: 2019-06-10.
- [5] Shimmy. Nosql use case scenarios or when to use nosql. Accessed: 2019-06-10.
- [6] wikipedia. Nosql. Accessed: 2019-06-10.