

# SQL OR NOSQL

## Bases de datos relacionales (SQL)

### Diferencias:

- SQL permite combinar de forma eficiente diferentes tablas para extraer información relacionada, mientras que NoSQL no lo permite o muy limitadamente.
- NoSQL permite distribuir grandes cantidades de información; mientras que SQL facilita distribuir bases de datos relacionales.
- SQL permite gestionar los datos junto con las relaciones existentes entre ellos; en NoSQL no existe este tipo de utilidades .
- NoSQL permite un escalado horizontal sin problemas – por su capacidad de distribución-; mientras que escalar SQL resulta más complicado.
- **Rendimiento.** Las bases de datos NoSQL ofrecen un mayor rendimiento que las SQL (necesitan menos recursos de hardware).
- **Fiabilidad.** Las bases de datos relacionales SQL son más fiables que las NoSQL (si un proceso tiene algún error, no se lleva a cabo).
- **Disponibilidad.** En cuanto a la disponibilidad, ambas alternativas son igual de válidas.
- **Consistencia.** La consistencia de las bases de datos NoSQL es pobre, en cambio las SQL son bases de datos muy consistentes (la consistencia es la capacidad de garantizar la integridad de los datos).
- **Almacenamiento.** Las bases de datos SQL son indicadas cuando la cantidad de datos no son extremadamente grandes, mientras que las NoSQL son ideales para manejar grandes volúmenes de datos.
- **Escalabilidad.** Las bases de datos NoSQL son escalables por lo que se pueden aumentar su capacidad fácilmente, sin embargo, las SQL pueden ser escalables, pero con un costo económico más elevado. Las bases de datos NoSQL utilizan un escalado horizontal (aumentar el número de servidores) mientras que las SQL utilizan un escalado vertical (aumentar los recursos de un servidor).

### Ventajas de las Bases de Datos relacionales

- Tienen un mayor soporte al llevar mucho tiempo en el mercado (mayor comunidad, aplicaciones y complementos).
- Atomicidad de la información. Al realizar cualquier operación en la base de datos, si surge algún problema, la operación no se realiza.
- Dispone de un sistema estándar bien definido (SQL) para las operaciones con la base de datos, como inserción, actualización o consultas. Este sistema es sencillo de comprender ya que se adapta al lenguaje común.
- Dispone de herramientas que permiten evitar la duplicidad de registros, garantizando la integridad referencial.

## Desventajas de las Bases de Datos relacionales

- La atomicidad de las operaciones juega un papel crucial en el rendimiento de las bases de datos, haciendo que este tipo de BDD no siempre sea la que mejor rendimiento ofrezca.
- Escalabilidad. Aunque está probada en muchos entornos productivos, por norma suele ser inferior a las bases de datos NoSQL.

## ¿CUÁNDO USAR SQL?

No existe una respuesta concreta, porque se presume que **cualquier cosa que se deba guardar**, debe ser en una base de datos relacional . Sin embargo los programadores podrían sugerir que en gaming o desarrollos de animación, no sería necesario.

En la mayoría de las opiniones, una base de datos relacional puede ser usada los siguientes ámbitos:

- Educación: para estructurar información, y aportar conocimiento lógico al estudiante.
- Desarrollos web: para mantener jerarquía de datos, siempre y cuando la capacidad de concurrencia, almacenamiento y mantenimiento no sean de considerable dificultad y la información sea consistente .
- Negocios: inteligencia y análisis de negocios, son temas que requieren el uso de SQL para facilitar el consumo de la información y la identificación de patrones en los datos.
- Empresarial: porque tanto el software a la medida y el software empresarial, poseen la característica de mantener información con estructura consistente .
- Cuando en un proyecto el volumen de los datos no tendrá un gran crecimiento, o este se realice de forma lenta, las bases de datos SQL.
- En proyectos donde el pico de usuarios que accedan a la base de datos esté previsto, las bases de datos relacionales funcionan de manera óptima.
- Si las necesidades de procesamiento de la base de datos requieren de un único servidor, se pueden utilizar bases de datos SQL.

## Bases de datos no relacionales (NoSQL)

### Ventajas de una base de datos no relacional o NoSQL database

- La escalabilidad y su carácter descentralizado. Soportan estructuras distribuidas.
- Suelen ser bases de datos mucho más abiertas y flexibles. Permiten adaptarse a necesidades de proyectos mucho más fácilmente que los modelos de Entidad Relación.
- Se pueden hacer cambios de los esquemas sin tener que parar bases de datos.
- Escalabilidad horizontal: son capaces de crecer en número de máquinas, en lugar de tener que residir en grandes máquinas. Esta escalabilidad coincide con uno de los

objetivos DevOps, de adaptarse de forma inmediata a las necesidades de los proyectos.

- Se pueden ejecutar en máquinas con pocos recursos.
- Optimización de consultas en base de datos para grandes cantidades de datos.
- Realizan consultas utilizando JSON.
- Aporta rapidez, escalabilidad, flexibilidad y sencillez a la gestión de la misma.

#### Desventajas de una base de datos NoSQL

- No todas las bases de datos NoSQL contemplan la atomicidad de las instrucciones y la integridad de los datos. Soportan lo que se llama consistencia eventual.
- Problemas de compatibilidad entre instrucciones SQL. Las nuevas bases de datos utilizan sus propias características en el lenguaje de consulta y no son 100% compatibles con el SQL de las bases de datos relacionales. El soporte a problemas con las queries de trabajo en una base de datos NoSQL es más complicado.
- Falta de estandarización. Hay muchas bases de datos NoSQL y aún no hay un estándar como sí lo hay en las bases de datos relacionales. Se presume un futuro incierto en estas bases de datos.
- Soporte multiplataforma. Aún quedan muchas mejoras en algunos sistemas para que soporten sistemas operativos que no sean Linux.
- Poca usabilidad. Suelen tener herramientas de administración no muy usables o se accede por consola.

#### ¿CUÁNDO USAR NOSQL?

Básicamente se utilizan en:

- Redes sociales: casi obligatorio .
- Desarrollo Web: debido a la poca uniformidad de la información que se encuentra en Internet; aun cuando también puede emplearse SQL.
- Desarrollo Móvil: debido a la tendencia – en crecimiento- de Bring Your Own Device.
- Big Data: debido a la administración de grandísimas cantidades de información y su evidente heterogeneidad.
- Cloud (XaaS): “Everything as a service”; NoSQL puede adaptarse casi a cualquier necesidad del cliente, y sus particularidades .
- Si el crecimiento de la base de datos se realiza de forma rápida, con grandes aumentos en poco tiempo, lo ideal es recurrir a bases de datos no relacionales NoSQL.
- Si el acceso a la base de datos puede sufrir picos altos y en múltiples ocasiones, lo mejor es optar por No SQL.
- Si las necesidades de procesamiento no se pueden prevenir, es mejor utilizar bases de datos NoSQL escalables (permiten expandirse).

## CONCLUSIÓN

Viendo los datos que atrás he comentado los que he ido recogiendo de varias páginas y varias fuentes de información, utilizando también la información que nos da Joseja que deberíamos empezar con una Base de Datos vacía. Creo que cualquiera de las 2 opciones son correctas, pero para comenzar el proyecto igual la mejor opción es SQL ya que nos facilita el trabajo inicial en algunos aspectos. Como pueden ser las relaciones entre tablas, la integridad de los datos, consistencia y fiabilidad. Cabe destacar como un punto importante la atomicidad de la información ya que eso nos crea una medida de seguridad extra a nuestro proyecto.