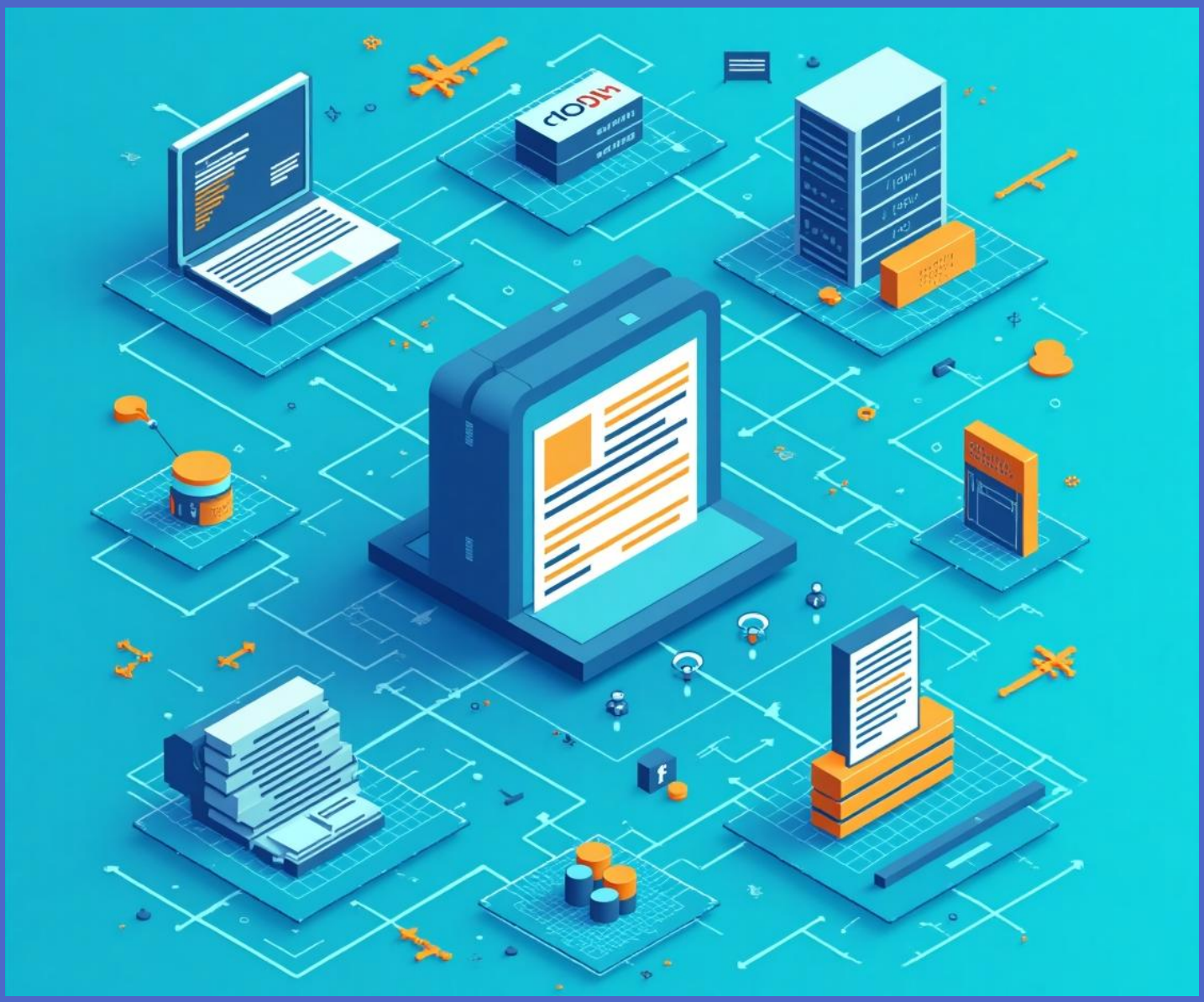
BASES DE DATOS

FORO EVALUABLE 2



ALUMNO CESUR

24/25

Alejandro Muñoz de la Sierra

PROFESOR

Inmaculada Morales Quesada

Introducción

En los últimos años, la capacidad de las bases de datos tradicionales se ha visto sobrepasada por el volumen, la diversidad y la rapidez con que se generan los datos. Como respuesta, las bases de datos NoSQL (Not Only SQL) han emergido, ofreciendo arquitecturas diseñadas para el manejo de grandes cantidades de información distribuida, heterogénea y sin una estructura rígida, pero manteniendo un rendimiento alto. En este foro, examinaremos en detalle sus características más importantes, su clasificación y cómo han evolucionado, así como su función complementaria a las bases de datos relacionales.

Importancia de las bases de datos NoSQL

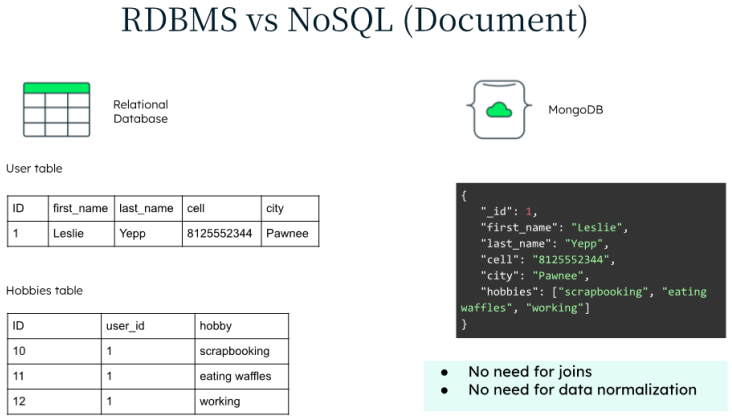
Las bases de datos NoSQL no buscan reemplazar a las tradicionales; más bien, amplían el rango de opciones en situaciones donde el modelo relacional empieza a encontrar limitaciones. Son cruciales en entornos donde el rendimiento, la escalabilidad o la disponibilidad son más importantes que la integridad relacional, como en aplicaciones de Big Data, redes sociales, el Internet de las Cosas (IoT) o sistemas que operan en tiempo real.

La popularidad de NoSQL se debe, en esencia, a tres factores principales:

- El crecimiento exponencial de los datos que se generan diariamente.

- La demanda de respuestas casi instantáneas por parte de los usuarios.

- Las dificultades de escalabilidad y la falta de adaptabilidad de las bases SQL tradicionales frente a estos desafíos.



Características principales de NoSQL

Las bases de datos NoSQL se distinguen por:

- Escalabilidad horizontal: la capacidad de agregar nodos con facilidad para distribuir la carga de trabajo.

- Distribución de datos: la replicación de datos entre servidores para mejorar la disponibilidad y la resistencia a fallos.

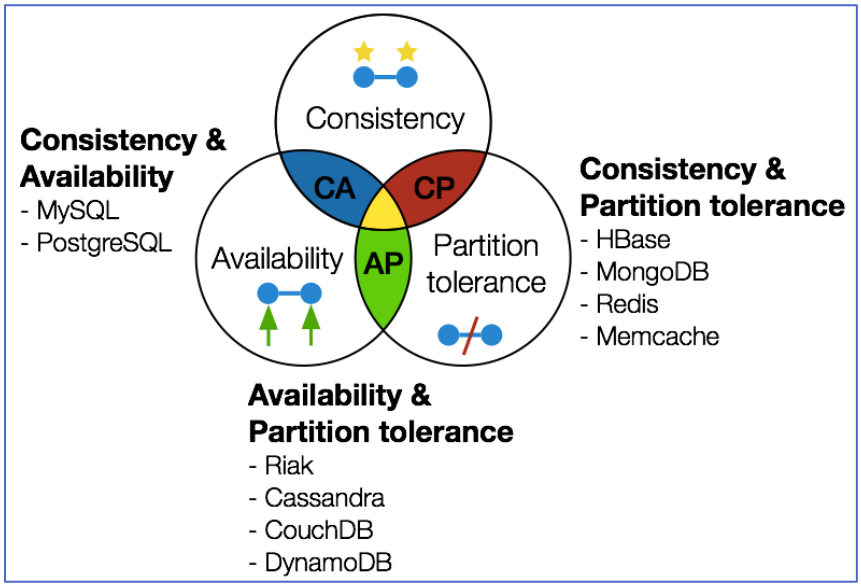
- Modelo flexible (sin esquema fijo): ideal para trabajar con datos heterogéneos o que cambian constantemente.

- Consistencia eventual: se prioriza la disponibilidad y la rapidez de respuesta, en lugar de adherirse estrictamente al modelo ACID.

- Consultas simples y eficientes: diseñadas para operaciones frecuentes y en grandes cantidades.

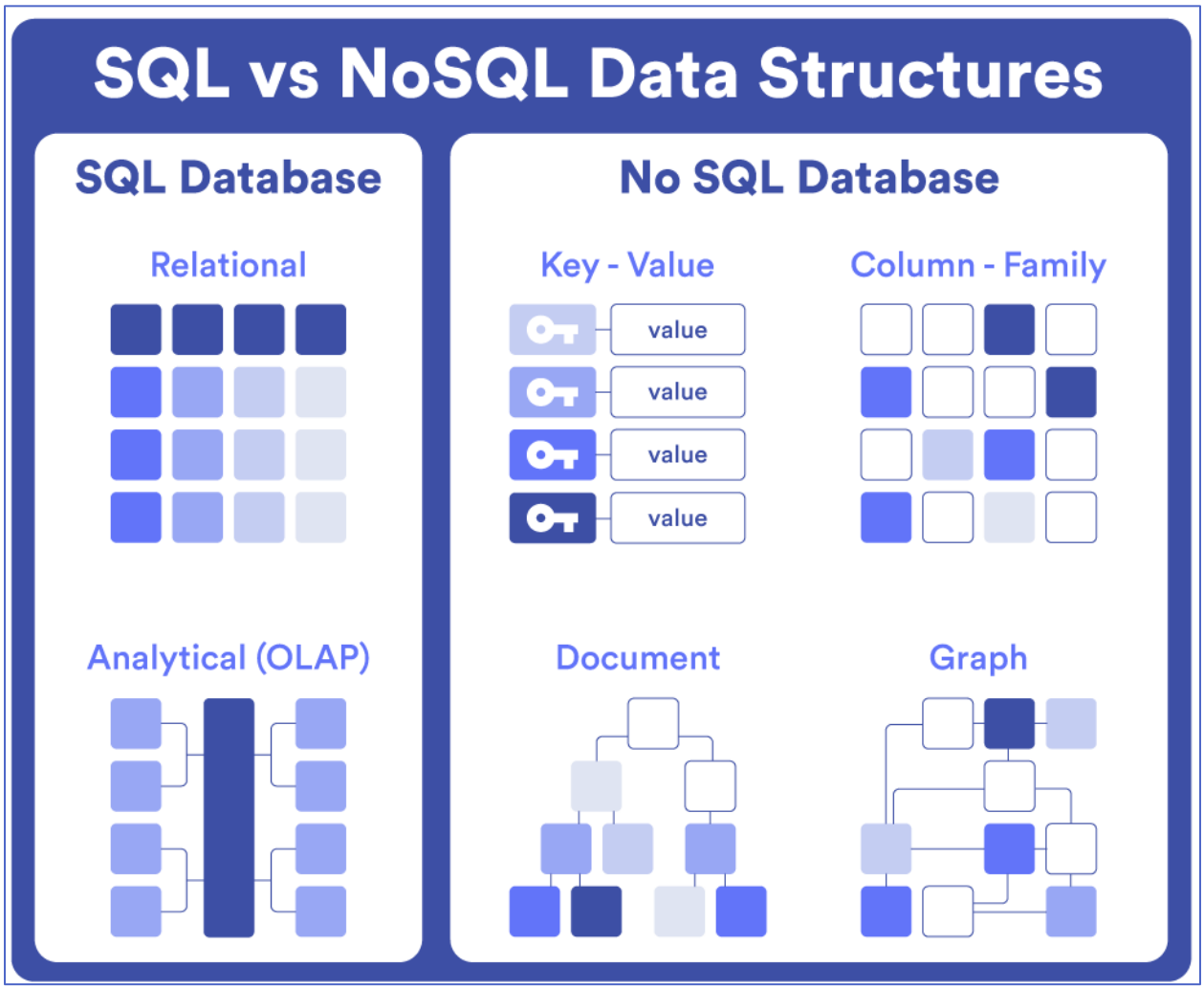
- Integración sencilla con lenguajes orientados a objetos.

En este contexto, el teorema CAP es relevante. Establece que ningún sistema distribuido puede garantizar simultáneamente consistencia, disponibilidad y tolerancia a particiones. En general, NoSQL tiende a sacrificar la consistencia en aras de las otras dos.



Taxonomía de NoSQL: Cuatro grandes familias

Tipo Descripción Ejemplos Casos de uso
- Clave-valor Almacenan pares únicos de clave y valor Redis, DynamoDB Gestión de sesiones, carritos de compra
- Documentales Guardan datos semiestructurados, como JSON MongoDB, CouchDB CMS, catálogos, perfiles de usuario
- Columnas Organizan la información por columnas, no por filas Cassandra, HBase Análisis de Big Data, series temporales
- Grafos Modelan relaciones como nodos y aristas Neo4j, OrientDB Redes sociales, sistemas de recomendación



Uso combinado de NoSQL y bases relacionales

En la práctica, muchas organizaciones prefieren un enfoque híbrido conocido como persistencia políglota, donde se combinan diferentes tecnologías de almacenamiento según las necesidades del sistema:

SQL se utiliza para datos estructurados, operaciones transaccionales y relaciones complejas.

NoSQL se utiliza cuando se trabaja con datos no estructurados, consultas masivas o cuando se requiere agilidad y escalabilidad.

Ejemplo real: EMC utiliza Hadoop (una tecnología NoSQL) para recopilar datos sobre la percepción de la marca en redes sociales y sitios web, y luego analiza estadísticamente estos datos con herramientas SQL tradicionales.

Evolución y desafíos de NoSQL

Desde su origen como una alternativa técnica a las bases relacionales, NoSQL se ha convertido en una herramienta madura y ampliamente utilizada, especialmente en entornos tecnológicos de vanguardia como Google, Amazon o Facebook. Sin embargo, aún enfrenta ciertos retos:

Falta de estandarización: cada motor NoSQL tiene su propia lógica y sintaxis, lo que dificulta el aprendizaje y la migración. El atractivo de la novedad: A veces, la adopción de estas tecnologías responde más a una moda pasajera que a una evaluación concienzuda de lo que realmente se necesita.

Integración formativa incompleta: Es preciso seguir trabajando en la creación de material didáctico riguroso, recursos para la enseñanza y cierta homogeneización en la manera en que se abordan estos conceptos.

Conclusión

Las bases de datos NoSQL han transformado radicalmente la gestión de datos. Su habilidad para procesar grandes volúmenes de información, datos cambiantes y en tiempo real las vuelve esenciales en muchos sistemas contemporáneos. No se trata de una simple tendencia, sino de una herramienta tangible que da solución a problemas concretos. Sin embargo, no deberían ser consideradas como un reemplazo absoluto de SQL, sino más bien como un complemento.

Desde mi punto de vista, el enfoque más sensato y eficaz en el desarrollo de sistemas pasa por combinar ambas tecnologías de acuerdo con las exigencias del proyecto. Esta perspectiva híbrida permite sacar el máximo partido de cada modelo: la solidez de SQL y la adaptabilidad de NoSQL.

Referencias

https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview

https://www.mongodb.com/es/resources/basics/databases/nosql-explained

https://aws.amazon.com/es/nosql/

https://www.oracle.com/database/nosql/what-is-nosql/

https://www.confluent.io/blog/

https://neo4j.com/docs/getting-started/graph-database/

https://martinfowler.com/bliki/PolyglotPersistence.html

https://www.sciencedirect.com/science/article/pii/S0164121225000597