

Introducción a las bases de datos noSQL



El funcionamiento de casi todas las aplicaciones web se basa en el uso de una base de datos.

Hasta hace poco las bases de datos que se utilizaban eran bases de datos relacionales y se trabajaba con SQL (MySQL, Oracle, SQL Server, ...)

Hace poco han aparecido las otro tipo de bases de datos que reciben el nombre de bases de datos NoSQL(Not Only SQL).

El término NoSQL (aunque ya se había utilizado antes) toma impulso con la aparición de la web 2.0, con la llegada de Facebook, Twitter o YouTube, donde cualquier usuario podía subir contenido, provocando así un crecimiento exponencial de los datos.



90%

de los datos almacenados en el mundo se han creado en los últimos

2 años

Por un lado, el volumen de datos ha crecido tremendamente y, además, esta información es información no estructurada, información que no encaja en una tabla.

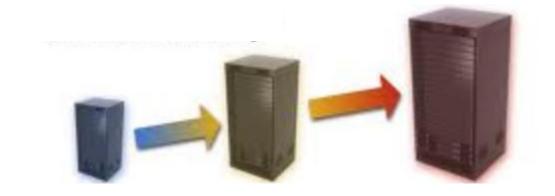
Por otro lado, los usuarios quieren respuestas inmediatas.

Las bases de datos relacionales no sirven para manejar este volumen de datos.

Los sistemas tienen que escalar para poder ser más grandes y más rápidos.

Existen 2 maneras de escalar:

Escalado vertical



Escalado horizontal





- Es caro
- tiene un límite,
- y las regulaciones de ciertos países no te permiten guardar los datos fuera del país, por lo que a veces no puedes hacerlo.

Escalado horizontal



Hay que distribuir los datos

Replicación

(todos los nodos deben guardar la misma información)
Distribuir datos en BD relacionales es
complicado, hay que mantener las propiedades
ACID:

hay que hacer un commit de 2 fases, hay que esperar a que todos los nodos de la red contesten. Unas transacciones se quedan bloqueadas esperando a que otras terminen

Sharding

(los nodos tienen datos distintos)

- También es difícil llevarlo a la práctica en las bases de datos relacionales

Escalado horizontal



Hay que distribuir los datos

Replicación

(todos los nodos deben guardar la misma información)
Distribuir datos en BD relacionales es
complicado, hay que mantener las propiedades

ACID:

hay que hacer un commit de Mases, hay que esperar a que todos los rodos de la red contesten. Unas transacciones se quedan bloqueadas esperando a que otras terminen

Retomamos el tema

Relacionales

(los nodos tienen datos distintos)

 También es difícil llevarlo a la práctica en las bases de datos relacionales En este contexto Google y Amazon se dan cuenta que las bases de datos relacionales no responden a sus necesidades ante el volumen de información que mueven, y deciden crear sus propias bases de datos (BigTable y DynamoDB), cada uno la suya para dar respuesta a sus necesidades.

El éxito de estos sistemas hizo que se iniciara el desarrollo de otros sistemas de bases de datos de código abierto y características similares (Cassandra, MongoDB, DynamoDB, HBase, Redis, ...)

Estas bases de datos tienen varias características en común, que las diferencian de las bases de datos relacionales:

- **arquitectura distribuida**: pensadas para trabajar en múltiples servidores que se comunican entre sí.
- Pueden manejar gran cantidad de datos.
- No genera cuellos de botella: los sistemas relacionales necesitan transcribir cada sentencia para poder ser ejecutada, y la ejecución de muchas sentencias complejas puede ralentizar el sistema.
- Flexibilidad de esquema: a diferencia de las relacionales donde el esquema de la base de datos es fijo, estático, en estas bases de datos existe el esquema no es fijo, es un esquema dinámico.
- No ACID → eventualmente consistentes (BASE)

BASE (Basically Available Soft-state Eventual Consistency)

NoSQL no garantiza las propiedades ACID, permite la Consistencia Eventual.

En un sistema relacional el commit de una transacción EXIGE que este cambio se replique de forma inmediata en todos los nodos (sistemas transaccionales). Esto no es así en los sistemas NoSQL, esto se conoce también como BASE (coherencia eventual flexible básicamente disponible).

El principio BASE facilita enormemente el escalado horizontal.

El principio BASE permite que las operaciones sean mucho más rápidas que los sistemas que garantizan las propiedades ACID

¿Voy a usar una base de datos NoSQL si mis datos pueden no ser consistentes?

Depende ...

Si es un sistema bancario

¿le gustaría a un cliente no ver reflejada (aunque sea momentáneamente) en su saldo un ingreso que le han realizado?

Si es una red social

¿tan importante es para un usuario ver un like más o menos en este momento?

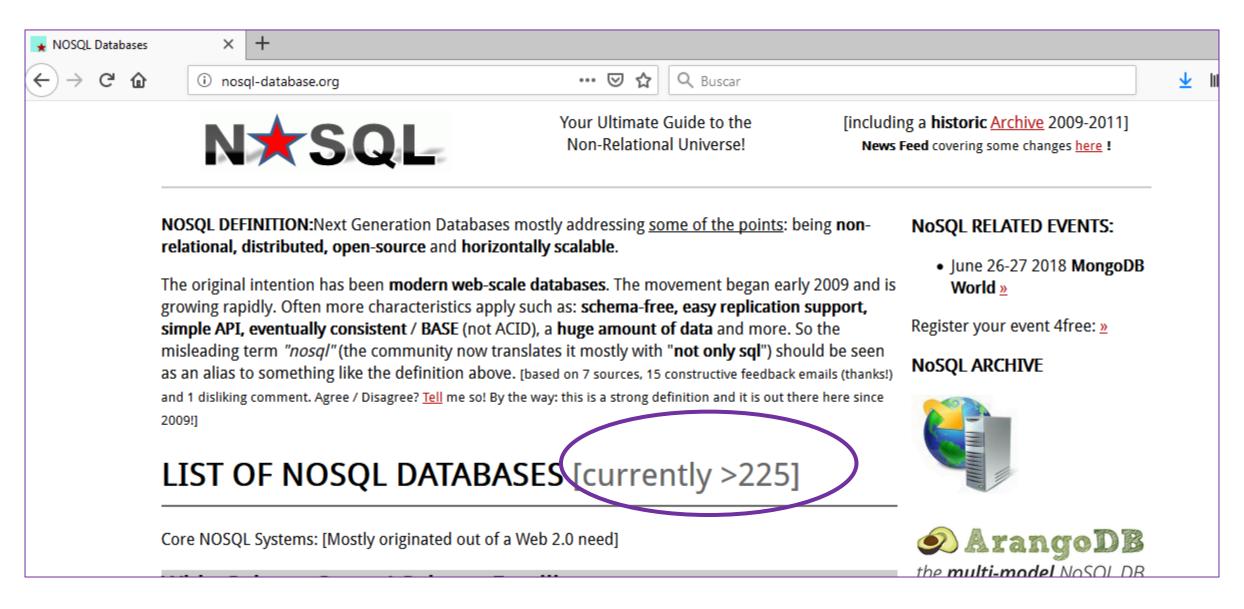
En 2009 se celebraba en San Francisco una conferencia sobre: Sistemas distribuidos, de código abierto y no relacionales (académicamente no existía un término para referirse a ese conjunto de características).

En twitter se refirieron al evento con el hashtag #nosql (el término nosql ya se había utilizado antes en 1998)

El término fue ampliamente aceptado y se comenzó a usar para referirse a bases de datos con esas características.

Se decidió que podía referirse a "not only" SQL como una idea de que no es una batalla, pueden coexistir ambos tipos de bases de datos.

¿Cuántos sistemas de bases de datos NoSQL existen actualmente?



Tipos de Bases de Datos noSQL

- Clave-Valor
- Columnas
- Documentales
- Grafos

Bases de Datos NoSQL clave-valor

Almacena cualquier tipo de objeto (número, cadena de caracteres, array, json,...)Valor y accede a estos objetos por un identificador. Clave

Clave	Valor					
23567	346					
23568	{depart1:"Lenguajes y Sistemas, depart2:"Física Aplicada",depart3:"Ciencias"}					
23590	45.22					
23591	"El Quijote"					
23596	"Vaya usted a saber"					

- Muy rápida la búsqueda por clave ya que se crea un índice de tipo hash para la clave.
- La búsqueda por valores de la base de datos no la puede realizar. No hay una patrón en este contenido. Hay que ir recuperando todos los contenidos y entonces ir analizándolos.

Ejemplo: DynamoDB (Amazon)

Bases de Datos NoSQL de columna

Es como si pusiéramos lo que serían las filas en columnas

ID	nombre	fecha_nac	num_hijos
1	Juan Robles	22/05/1985	2
2	María Alarcón	23/10/1980	3
3	Pedro Vigo	12/08/2000	0
4	Rosa Blas	04/06/1977	2

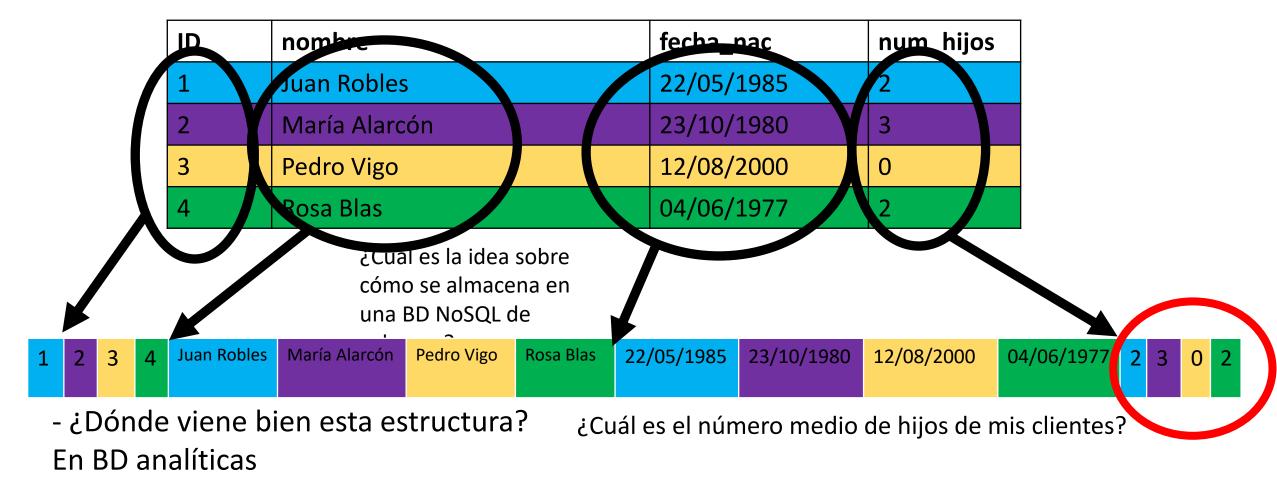
Tabla CLIENTE en M. Relacional

- Se almacena por filas.
- Cada lectura recupera una fila completa

¿Cuál es la idea sobre cómo se almacena en una BD Relacional?

1 Ju	an Robles	22/05/1985	2	2	María Alarcón	23/10/1980	3	3	Pedro Vigo	12/08/2000	0	4	Rosa Blas	04/06/1977	2
------	-----------	------------	---	---	---------------	------------	---	---	------------	------------	---	---	-----------	------------	---

Bases de Datos NoSQL de columna



- Desventajas: se ralentizan las operaciones de borrado y modificación *Ejemplo de BD columna: BigTable (Google)*

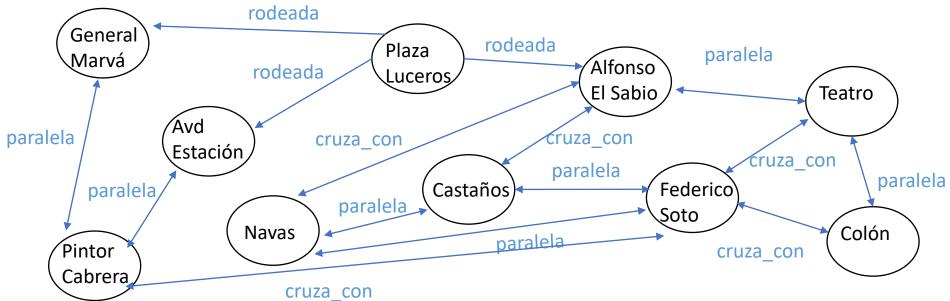
Bases de Datos NoSQL de grafo

La información se representa como nodos de un grado y sus relaciones son las aristas del mismo.

Este tipo de bases de datos es muy útil para información muy interconectada entre sí y donde esta conexión es tan importante como la información en sí.

Son muy muy eficientes para ir pasando de un nodo a otro

Se usan en casos muy específicos: redes sociales, tráfico aéreo, distribución eléctrica...



Ejemplo de BD grafo: Neo4j

NO se refiere a una BD que guarda pdf's o documentos Word, ...

Se refiere a una BD que guarda estructuras que a su vez pueden tener muchos niveles de información.

Para cada entrada no tenemos las típicas columnas de estructura fija, cada entrada tendrá su propia estructura.

Cada una de estas estructuras que contiene información desestructurada se llaman **documento.**

Muchas bases de datos documentales representan estas estructuras como objetos JSON.

```
"Departamento de Lenguajes y Sistemas Informáticos",
Nombre:
Empleados:
                70,
Asignaturas:
                [ "Diseño de Bases de Datos", "Fundamentos de Bases de
                Datos", "Programación", "Gestión de la Información" ],
Dirección:
                                        "Carretera de San VIcente",
                        calle:
                        número:
                                        2,
                        codigopostal:
                                        03690
                },
Teléfono:
                        centralUA:
                                                número:
                                                                965903400,
                                                extensión:
                                                                3972
                        directo:
                                        [965903466, 965943211]
                                       DBD(34014) - T7
```

```
"Departamento de Lenguajes y Sistemas Informáticos",
Nombre:
                                                                                   campos
Empleados:
                70.

Valor numérico

                                                                               Cadena caracteres
Asignaturas:
                 [ "Diseño de Bases de Datos", "Fundamentos de Bases de
                Datos", "Programación", "Gestión de la Información" ],
                                                                                Array
Dirección:
                                        "Carretera de San VIcente",
                        calle:
                        número:
                                        2,
                        codigopostal:
                                        03690
                },
                                                                                  Subdocumentos
                                                                 965903400,
Teléfono:
                        centralUA:
                                                 número:
                                                 extensión:
                                                                 3972
                        directo:
                                         [965903466, 965943211]
                                        DBD(34014) - T7
```

```
Nombre: "Departamento de Lenguajes y Sistemas
Informáticos",
Empleados: 70,
Asignaturas: ["Diseño de Bases de Datos",
"Fundamentos de Bases de Datos", "Programación",
"Gestión de la Información" ],
Dirección:
             {calle: "Carretera de San VIcente",
             número:
             codigopostal: 03690
             { centralUA: { número: 965903400,
Teléfono:
                           extensión:
                                        3972
                      [965903466, 965943211]
```

```
{
Nombre: "Departamento Física Aplicada"

Empleados: 25,

Asignaturas: ["Física1", Física 2"],

Dirección: {calle: "Carretera de San VIcente", número: 2, codigopostal: 03690 },

Teléfono: { centralUA: { número: 965903400, extensión: 3977 } directo: [965903477, 965943220] }
}
```

```
Nombre: "Departamento Biología"
Empleados: 40,
Asignaturas: ["Biología 1", Biología 2"],
             {calle: "Carretera de San VIcente",
Dirección:
             número:
             codigopostal: 03690
Teléfono:
             { centralUA: { número: 965903400,
                           extensión:
                                         3988
              directo
                           965907766
```

Los documentos se agrupan formando colecciones.

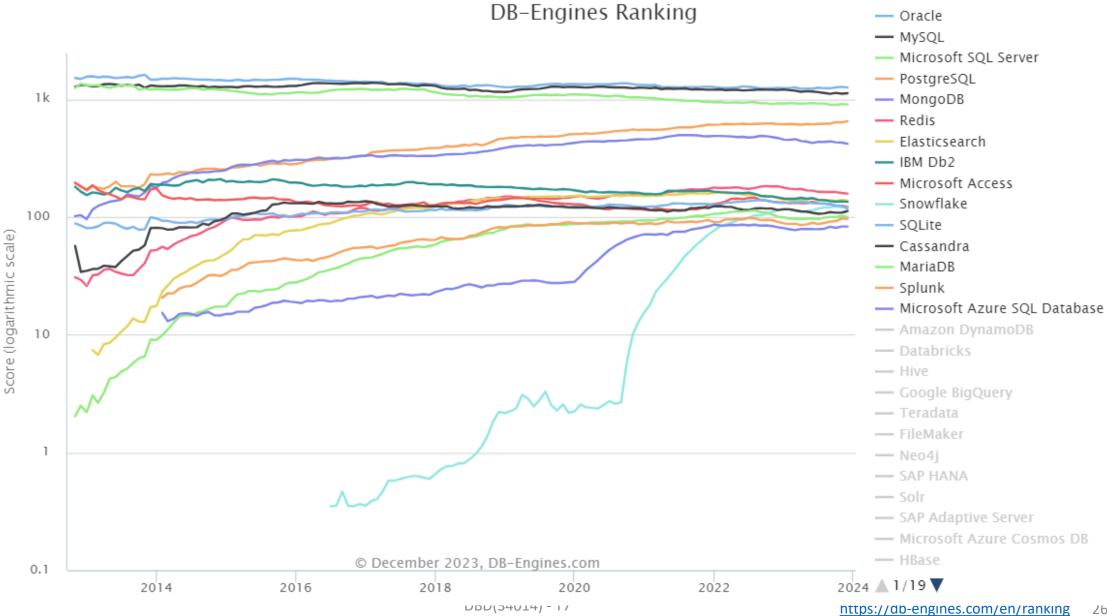
Permiten hacer indexación

Permiten hacer búsquedas por valor a cualquier nivel.

Permiten agrupamientos

Ejemplo: MongoDB, CoachDB

Conociendo ya algunas características de las bases de datos NoSQL, ¿se usan tanto como las relacionales?



363 systems in ranking, December 2020

			diffilig, beceimber 2020					
	Rank				S	core		
Dec 2020			DBMS	Database Model	Dec 2020	Nov 2020	Dec 2019	
1.	1.	1.	Oracle 🚹	Relational, Multi-model 👔	1325.60	-19.40	-20.80	
2.	2.	2.	MySQL 🚹	Relational, Multi-model 👔	1255.45	+13.81	-20.21	
3.	3.	3.	Microsoft SQL Server 🚹	Relational, Multi-model 👔	1038.09	+0.45	-58.11	
4.	4.	4.	PostgreSQL 🞛	Relational, Multi-model 👔	547.57	-7.49	+44.20	
5.	5.	5.	MongoDB 🚹	Document, Multi-model 🛐	457.73	+3.90	+36.61	
6.	6.	6.	IBM Db2 🚹	Relational, Multi-model 👔	160.43	-1.19	-10.91	
7.	7.	1 8.	Redis 🚹	Key-value, Multi-model 👔	153.63	-1.79	+7.39	
8.	8.	4 7.	Elasticsearch 🖽	Search engine, Multi-model 👔	152.49	+0.94	+2.24	
9.	9.	1 11.	SQLite 🚹	Relational	121.68	-1.63	+1.32	
10.	10.	10.	Cassandra 🚹	Wide column	118.84	+0.09	-1.87	
11.	11.	4 9.	Microsoft Access	Relational	116.74	-0.50	-12.73	
12.	12.	1 3.	MariaDB 🖽	Relational, Multi-model 👔	93.61	+1.31	+6.82	
13.	13.	↓ 12.	Splunk	Search engine	87.00	-2.71	-3.53	
14.	14.	1 5.	Teradata 🚹	Relational, Multi-model 👔	73.83	-1.77	-4.66	

417 systems in ranking, December 2023

	Rank Dec Nov Dec 2023 2023 2022					Score				
						Dec 2023	Nov 2023	Dec 2022		
\leq	1.	1.	1.	Oracle 🖽	Relational, Multi-model 🔞	1257.41	-19.62	+7.10		
	2.	2.	2.	MySQL #	Relational, Multi-model 👔	1126.64	+11.40	-72.76		
	3.	3.	3.	Microsoft SQL Server 🖽	Relational, Multi-model 🛐	903.83	-7.59	-20.52		
	4.	4.	4.	PostgreSQL 🚹	Relational, Multi-model 👔	650.90	+14.05	+32.93		
	5.	5.	5.	MongoDB 🔠	Document, Multi-model 🛐	419.15	-9.40	-50.18		
	6.	6.	6.	Redis 😷	Key-value, Multi-model 👔	158.35	-1.66	-24.22		
	7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model 🛐	137.75	-1.87	-7.18		
	8.	8.	4 7.	IBM Db2	Relational, Multi-model 👔	134.60	-1.40	-12.02		
	9.	1 0.	9.	Microsoft Access	Relational	121.75	-2.74	-12.08		
	10.	↑ 11.	1 11.	Snowflake 🔠	Relational	119.88	-1.12	+5.11		
	11.	4 9.	↓ 10.	SQLite 🚹	Relational	117.95	-6.63	-14.49		
	12.	12.	12.	Cassandra 🚹	Wide column, Multi-model 🛐	112.20	+3.03	-2.46		
	13.	13.	13.	MariaDB 🚹	Relational, Multi-model 👔	100.43	-1.66	-0.50		



Introducción a las bases de datos noSQL

