



BigData yNoSQL

Base de datos relacio

- Antes de adentrarnos y exponer qué es una BD NoSQL recordemos algunos conceptos básicos
- **Una base de datos** es un **repositorio** donde se almacenan datos relacionados y desde donde se pueden también recuperar. Cada una **se compone de una o más tablas** que guardan un conjunto de datos en **columnas** (atributo o característica del dato) **y filas** (que conforman un registro).

Base de datos relacional

- Una base de datos **es relacional** cuando se cumple con el modelo relacional que **garantiza la unicidad de registros, la integridad referencial y la normalización**.
- En ellas se usa el **lenguaje SQL** como estándar para la definición, manipulación y control. Este lenguaje es declarativo y es muy parecido al lenguaje natural, por estas razones es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales.

¿qué es una base de datos NoSQL?

- **Cada vez más datos:** empresas y personas se conectan a Internet, aumentando con ello tanto los usuarios que acceden a la red como los servicios que se ofrecen a través de ella.
- Esto ha generado un aumento considerable en el volumen de datos que se deben procesar, por lo que la versión clásica de **guardar los datos de forma centralizada**, es decir en un único espacio físico como en una base de datos “tradicional”, **no es ni óptima ni eficiente** por lo que son necesarios otros modelos que soporten la distribución masiva de los datos.

¿qué es una base de datos NoSQL?

- ◉ podemos definir una base de datos NoSQL como aquella que **no requiere de estructuras de datos fijas** como tablas; **no garantizan completamente las características ACID** y **escalan muy bien horizontalmente**.
- ◉ Se utilizan en entornos distribuidos que han de estar siempre disponibles y operativos y que gestionan un importante volumen de datos.

ACID

- ◉ **ACID** es un acrónimo de **A**tomicity, **C**onsistency, **I**solation and **D**urability:
 - ◉ **Atomicidad**, Si una operación consiste en una serie de pasos, bien todos ellos se ejecutan o bien ninguno, es decir, **las transacciones son completas**.
 - ◉ Consistencia(integridad), se **ejecutan** aquellas **operaciones que no van a romper las reglas y directrices de Integridad de la base de datos**. La propiedad de consistencia sostiene que cualquier transacción llevará a la base de datos desde un estado válido a otro también válido.
 - ◉ **Aislamiento**: Esta propiedad asegura que una operación no puede afectar a otras. Esto asegura que **la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error**
 - ◉ Durabilidad: Esta propiedad asegura que **una vez realizada la operación, ésta persistirá** y no se podrá deshacer aunque falle el sistema y que de esta forma los datos sobrevivan de alguna manera.

BASE

- Las BBDD NoSQL son repositorios de almacenamiento más optimistas , siguen el **modelo BASE**:
 - **BAasic availability** – el almacén funciona la mayoría del tiempo incluso ante fallos gracias al almacenamiento distribuido y replicado
 - **Soft-state** – los almacenes no tienen porque ser consistentes ni sus réplicas en todo momento. El programador puede verificar esa consistencia.
 - **Eventual consistency** – la consistencia se da eventualmente
- **BASE es una alternativa flexible a ACID** para aquellos almacenes de datos que no requieren un adherencia estricta al modelo relacional

- Las aplicaciones web modernas presentan desafíos muy distintos a las que presentan los sistemas empresariales tradicionales (e.j. sistemas bancarios):
 - Datos a escala web
 - Alta frecuencia de lecturas y escrituras
 - Cambios de esquema de datos frecuentes
 - Las aplicaciones sociales (no bancarias) no necesitan el mismo nivel de ACID

Características NoSql

- El lenguaje estándar no tiene porqué ser **SQL(Not Only SQL)**
- El **esquema de datos es flexible** o no tiene un esquema predefinido, lo que permite el tratamiento de **datos heterogéneos**
- Las propiedades ACID no siempre están garantizadas
- **Mayor coherencia entre los datos de programas y bases de datos**
- Diseñadas para ser **escalables** generalmente de forma horizontal
- Suelen ser **distribuidas**

Características de programas que usan NoSql

- Trabajan con **datos cuyo origen y formato es variable**
- Trabajan con **datos muy relacionados**
- **Necesidad de una mayor capacidad analítica**
- **Mayor volumen de datos** (espacio > barato que procesamiento y tiempo)
- **Mayor disponibilidad y flexibilidad**
- **Trabajan en tiempo real**

¿Diferencias entre SQL y NoSQL?

- **No hay un modelo único de datos.** El modelo relacional ofrece una visión uniforme de los datos, la relación, mientras que las bases de datos NoSQL engloban a muchos modelos de datos
- En el **modelo relacional** es necesario definir **a priori un esquema conceptual** que indique qué datos hay, cómo se estructuran, que atributos poseen y cómo se relacionan.
- Las bases de datos NoSQL **no** suelen permitir **operaciones JOIN** (búsqueda de datos entre varias tablas) ya que al manejar volúmenes de datos tan grandes el coste en tiempo de respuesta puede ser muy alto
- **En NoSQL generalmente no se usa el lenguaje SQL**, por ejemplo Cassandra utiliza CQL, MongoDB (de la que hablaremos a continuación) utiliza JSON y Big Table utiliza GQL.
- Las bases de datos relacionales tienen un gran recorrido, lo que ha permitido crear estándares, como el lenguaje SQL, lo que no ha sido posible aún para las bases de datos NoSQL

Tipos de bases de datos NoSQL

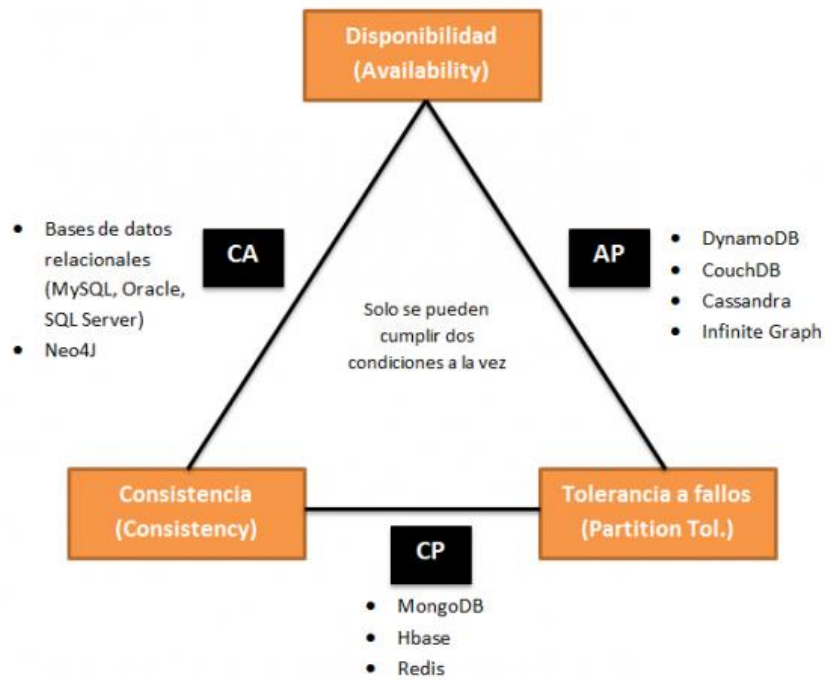
- **Clave valor:** Las más sencilla en cuanto a funcionalidad. [Cassandra](#), [BigTable](#) o [HBase](#) son ejemplos de este tipo. Son bastante eficientes tanto en lectura como en escritura.
- **Columnares:** las bases de datos, en lugar de estar estructuradas por filas, están estructuradas por columnas. Al tratarse de una sola dimensión, hace más eficiente la recuperación de la información. Por ejemplo [Vertica](#).
- **Documentos:** almacena la información como un documento, permitiendo realizar consultas bastante avanzadas sobre el mismo. Por ello, suele considerarse como el más versátil. [MongoDB](#) o [CouchDB](#) son ejemplos de ello.
- **Grafos:** los datos son representados como nodos y aristas que modelizan la relación entre esos nodos. De esta manera, podemos emplear la teoría de grafos para recorrer y navegar por su contenido. Su principal ventaja es que permite una navegación más eficiente entre relaciones que en un modelo relacional. [Neo4J](#) o [Virtuoso](#) son ejemplos de ello

Teorema de CAP de Brewer

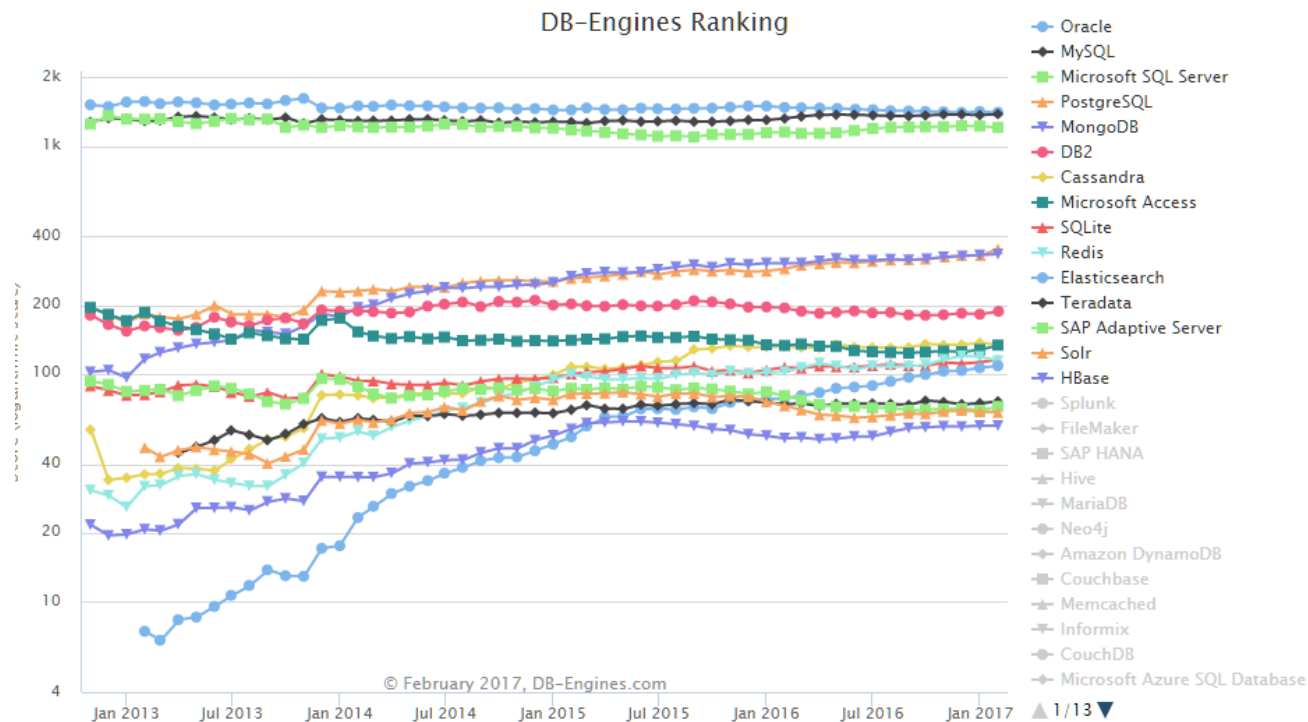
- Consistence, Availability and Partition tolerance
 - Consistencia: al realizar una consulta o inserción siempre se tiene que recibir la misma información, con independencia del nodo o servidor que procese la petición.
 - Disponibilidad: que todos los clientes puedan leer y escribir, aunque se haya caído uno de los nodos.
 - Tolerancia a particiones: Los sistemas distribuidos pueden estar divididos en particiones (generalmente de forma geográfica). Así que esta condición implica, que el sistema tiene que seguir funcionando aunque existan fallos o caídas parciales que dividan el sistema
- Desafortunadamente, en un sistema distribuido sólo dos de estas propiedades pueden ser garantizadas de forma simultánea

Ejemplos de CAP

- Según satisfagan unos criterios u otros, podemos encontrar:
- AP: Cassandra (clave-valor)
- CP: MongoDB (documental) .
- CA: RDBMS. (relacional)



Ranking de uso de db



Mismo Ranking indicando tipo

318 systems in ranking, February 2017

Rank			DBMS	Database Model	Score		
Feb 2017	Jan 2017	Feb 2016			Feb 2017	Jan 2017	Feb 2016
1.	1.	1.	Oracle +	Relational DBMS	1403.83	-12.89	-72.31
2.	2.	2.	MySQL +	Relational DBMS	1380.30	+14.02	+59.18
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1203.45	-17.50	+53.23
4.	↑ 5.	↑ 5.	PostgreSQL	Relational DBMS	353.68	+23.31	+65.02
5.	↓ 4.	↓ 4.	MongoDB +	Document store	335.50	+3.60	+29.90
6.	6.	6.	DB2	Relational DBMS	187.90	+5.41	-6.58
7.	7.	↑ 8.	Cassandra +	Wide column store	134.38	-2.06	+2.62
8.	8.	↓ 7.	Microsoft Access	Relational DBMS	133.39	+5.94	+0.31
9.	↑ 10.	9.	SQLite	Relational DBMS	115.31	+2.93	+8.53
10.	↓ 9.	10.	Redis +	Key-value store	114.03	-4.66	+11.96
11.	11.	↑ 12.	Elasticsearch +	Search engine	108.31	+2.14	+30.47
12.	12.	↑ 13.	Teradata	Relational DBMS	75.60	+1.43	+2.22
13.	13.	↓ 11.	SAP Adaptive Server	Relational DBMS	71.74	+2.63	-8.30
14.	14.	14.	Solr	Search engine	67.69	-0.39	-4.59
15.	15.	↑ 16.	HBase	Wide column store	59.24	+0.10	+7.22
16.	16.	↑ 18.	Splunk	Search engine	56.03	+0.54	+13.20
17.	17.	17.	FileMaker	Relational DBMS	55.19	+1.71	+8.16
18.	18.	↑ 19.	SAP HANA +	Relational DBMS	52.45	+0.52	+14.37
19.	19.	↓ 15.	Hive	Relational DBMS	47.95	-3.19	-4.83
20.	20.	↑ 23.	MariaDB	Relational DBMS	45.35	+0.31	+16.57
21.	21.	21.	Neo4j +	Graph DBMS	36.27	+0.00	+3.98