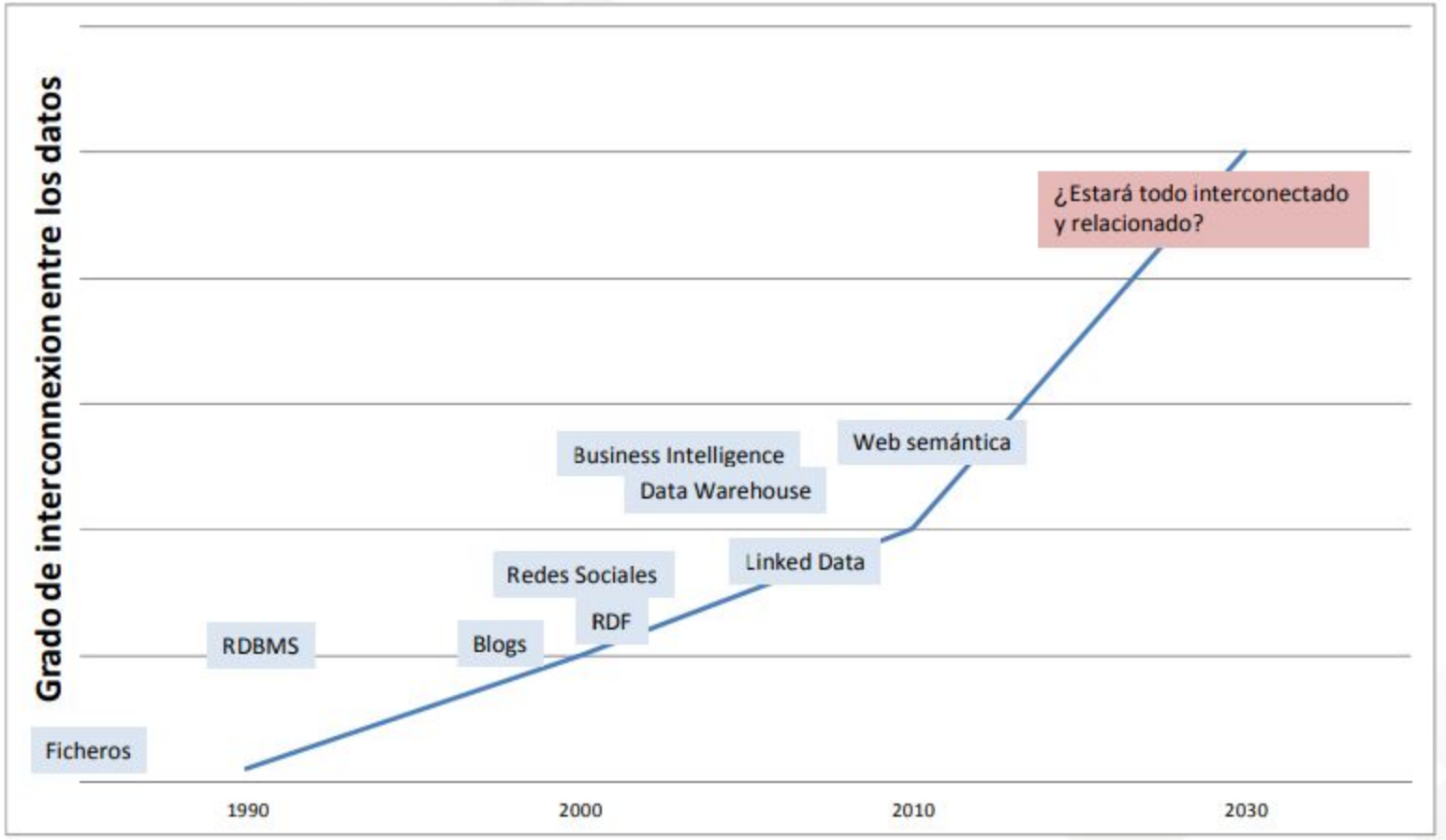
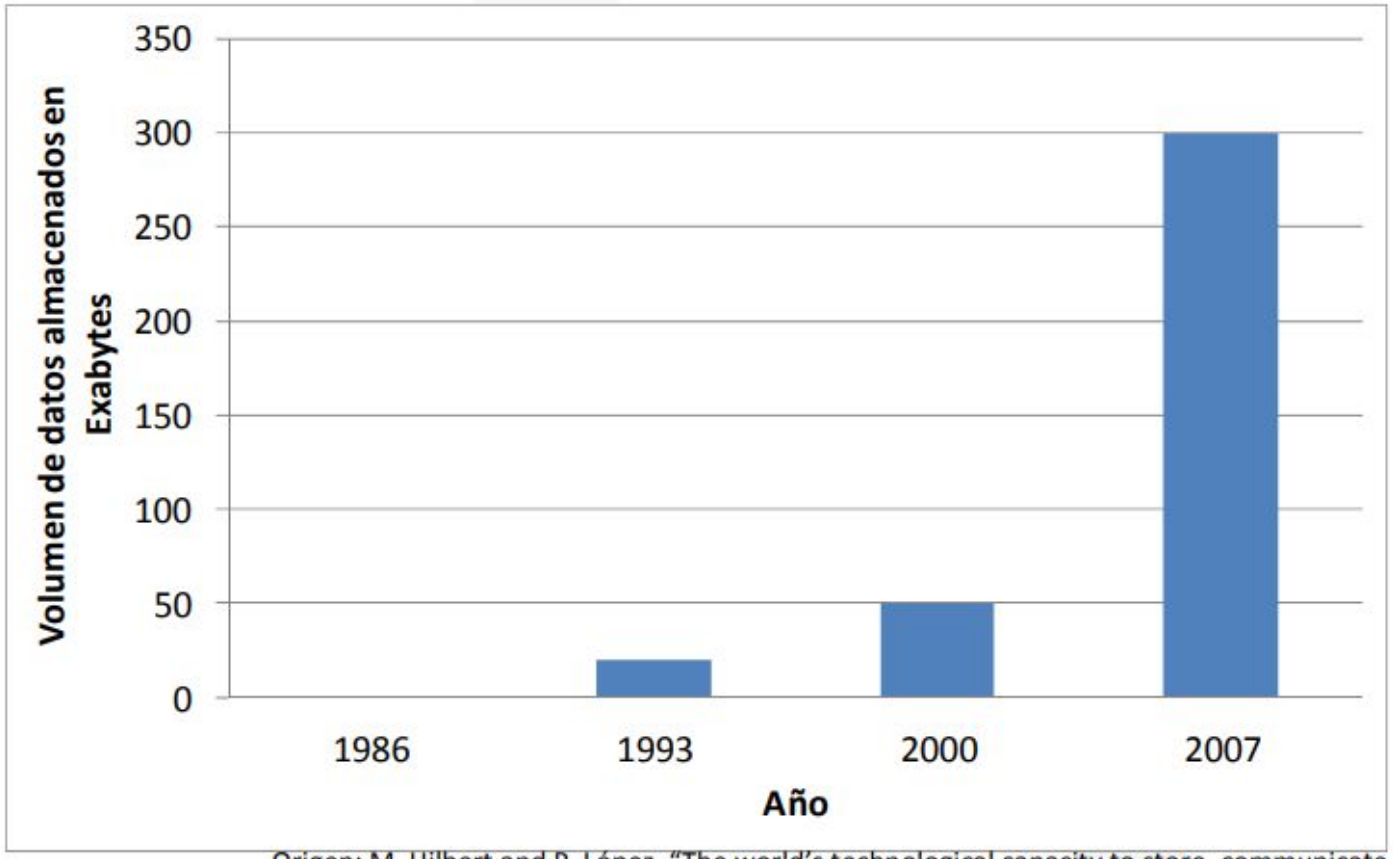
UT7.
Motivación y Características

Origen, uso y representación de los datos



Volumen de datos almacenados



Origen: M. Hilbert and P. López, "The world's technological capacity to store, communicate, and compute information", *Science*, February 10, 2011

Características de NoSQL

- No ofrecen SQL como lenguaje estándar.
- Esquema flexible (*schemaless*)
- No garantizan las propiedades ACID al completo.
- Reducen, en parte, la falta de concordancia (*impedance mismatch*).
- Favorecen la escalabilidad, especialmente horizontal.
- Ofrecen solución a los inconvenientes y rigidez del modelo relacional.
- En general y frecuentemente son:
  - Distribuidas
  - De código abierto

Diferencias entre NoSQL y bases de datos relacionales

- No hay un modelo de datos único:

  NoSQL ofrece diferentes modelos de datos, mientras que las bases de datos relacionales sólo proporcionan el modelo relacional.

- Esquema flexible (*schemaless*):

  A diferencia del modelo relacional en NoSQL no es necesario un esquema predefinido que defina como se estructuran y relacionan los datos.

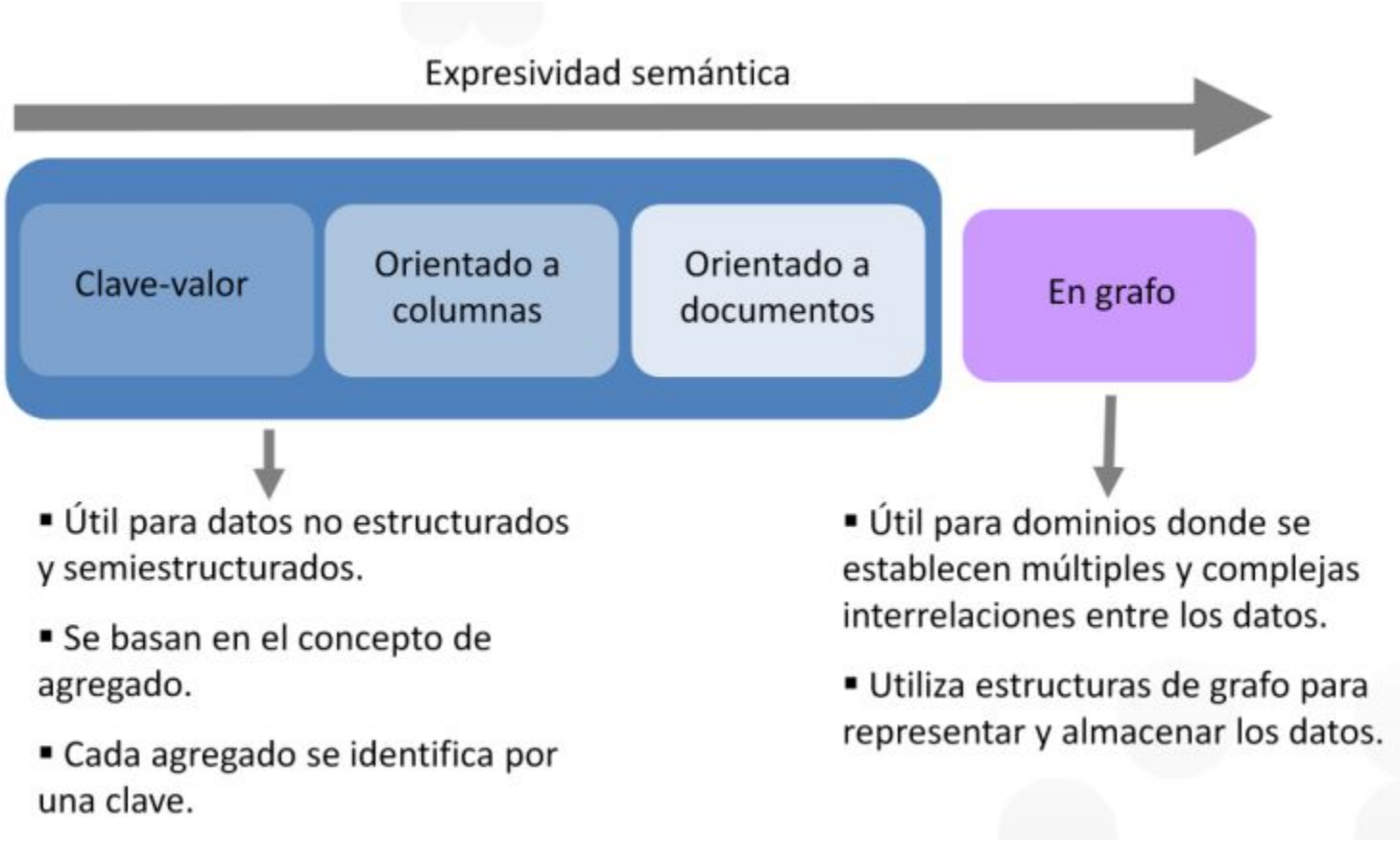
- Falta de estandares:

  Mientras que detrás de las bases de datos relacionales existen estándares, éste no es el caso de las BD NoSQL (de momento).

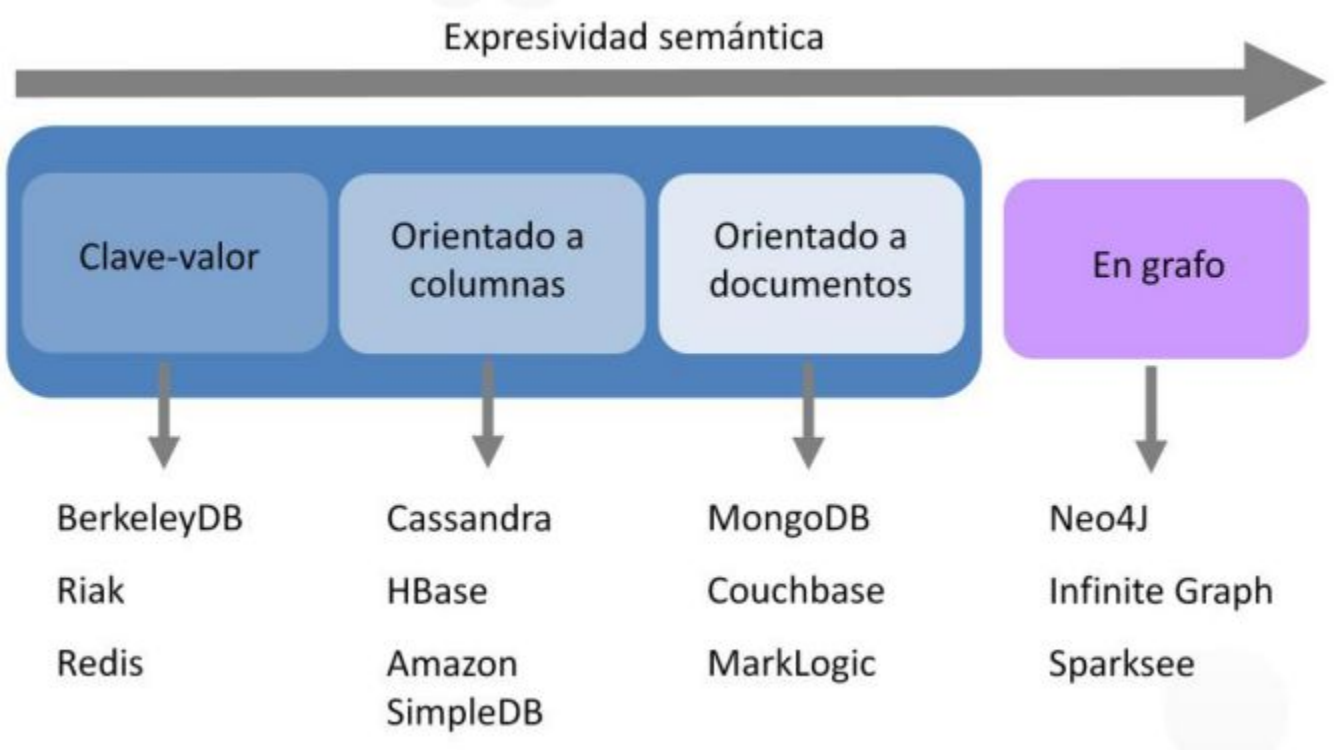
La actualidad

- La tecnología de bases de datos, junto con el del procesamiento de los datos, están en continuo desarrollo. Nuevos productos y estrategias se crean constantemente.
- Existe un debate abierto en la comunidad sobre las ventajas y desventajas de las diferentes bases de datos:
  - Bases de datos relacionales versus NoSQL
  - Entre los diferentes tipos de NoSQL

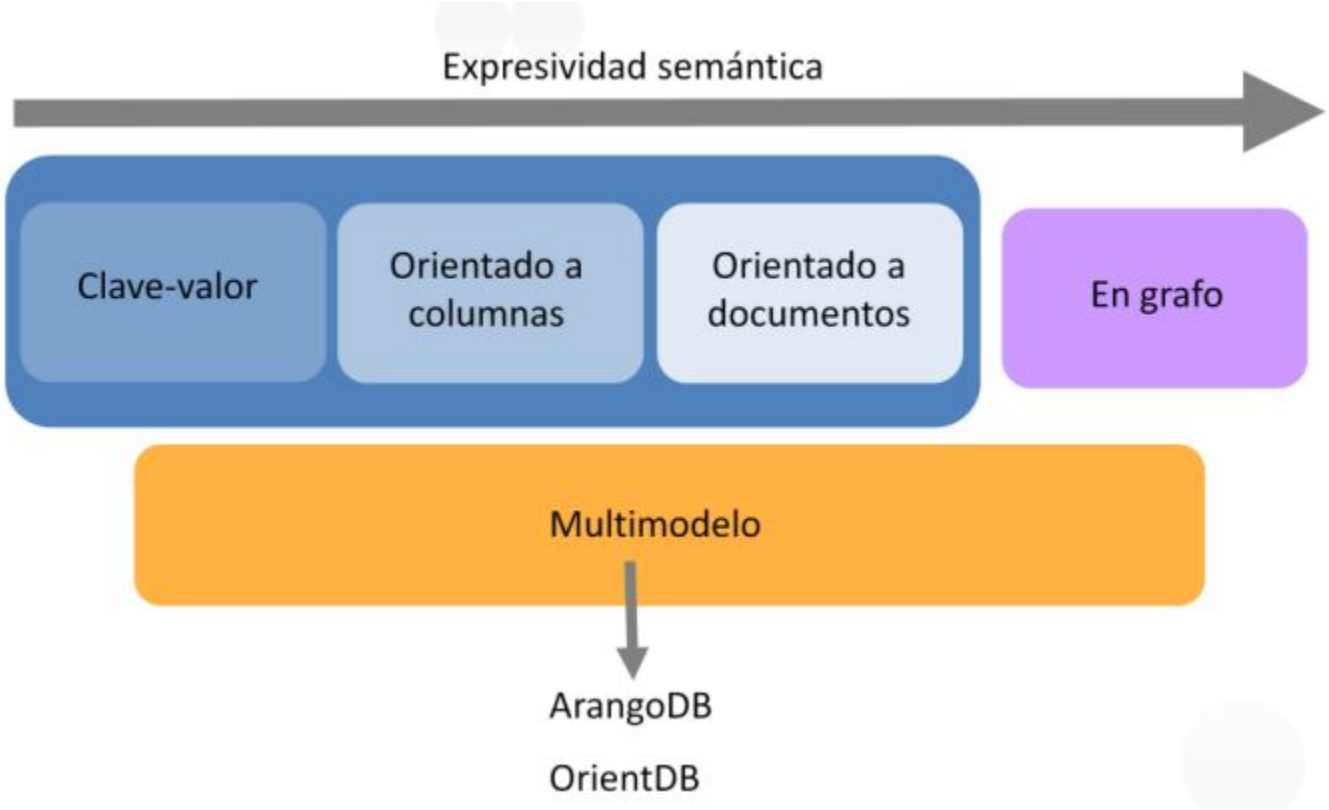
Modelos de datos



Ejemplos de BD No SQL



Ejemplos de BD No SQL

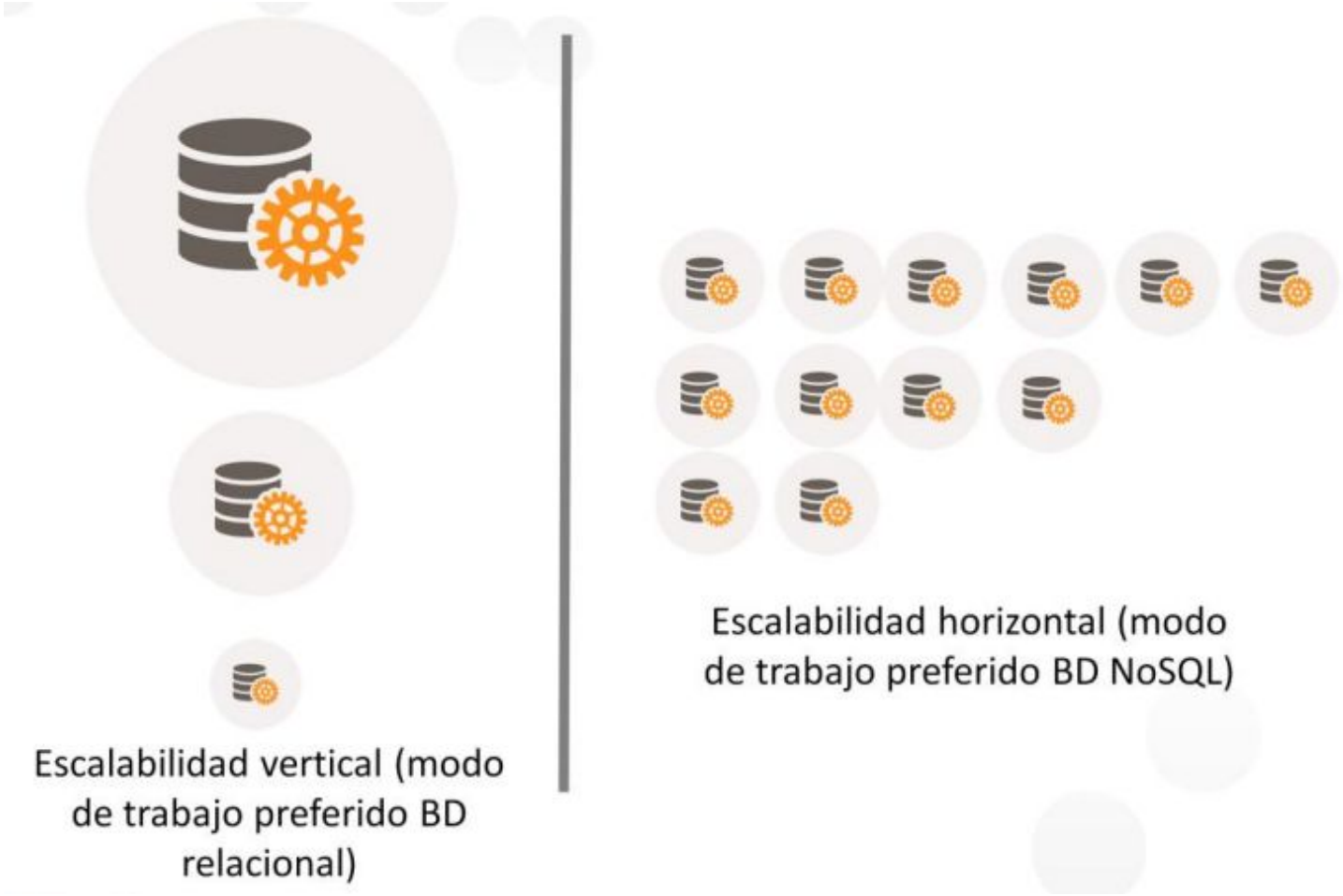


https://hostingdata.co.uk/nosql-database/

No ofrecen SQL como lenguaje

- Algunas BD NoSQL ofrecen su propio lenguaje de manipulación y consulta de los datos.
- Acceso vía API REST: ejecución de peticiones HTTP del tipo POST, GET, PUT, DELETE
- Cada BD NoSQL proporciona *drivers* de acceso a la BD para multitud de lenguajes de programación (C, C++, C#, Java, PHP, Python, Perl, Erlang…).
- Proporcionan operaciones que permiten la integración con sistemas de computación distribuida (p.e. el *framework* MapReduce).

Distribución: escalabilidad



Persistencia políglota

- La persistencia políglota consiste en el uso de diferentes tecnologías de almacenamiento para dar respuesta a las diferentes necesidades de almacenamiento.
- La persistencia políglota no consiste en substituir una tecnología de almacenamiento de datos por otra, sino en utilizar, dentro de un mismo proyecto o una misma empresa, la tecnología de almacenamiento de datos más apropiada para cada necesidad y tarea.

Escenarios donde no funciona un SGBD relacional

- **Data warehouse**: el perfil de los usuarios de los data warehouse es muy diferente al perfil de los usuarios de los sistemas gestores de bases de datos relacionales, así como el tipo de operaciones que realizan.

- **Flujos de datos financieros en tiempo real**: los datos para la toma de decisiones en mercados financieros deben estar disponibles en tiempo real y a menudo no es necesario almacenar todo el flujo de datos para tomar las decisiones porque el tiempo necesario para almacenarlo puede suponer un retraso innecesario, que puede resultar crítico en la toma de decisiones.

- **Redes sociales, motores de búsqueda, sistemas de recomendación de películas y plataformas de juegos online:** Estos entornos de aplicación necesitan almacenar flujos provenientes de diferentes fuentes de forma concurrente en un mismo archivo. Las escrituras acostumbran a ser append-only (es decir, de inserción de datos) y las lecturas secuenciales. Debido a la inmensa cantidad de datos generados, se tienen que utilizar sistemas de almacenamiento económicos que pueden fallar a menudo, por lo que los datos tienen que estar fuertemente replicados. Por estos motivos, una base de datos NoSQL puede ofrecer mejores resultados que una base de datos relacional

Escenarios donde no funciona un SGBD relacional

- **Geolocalización móvil y sensores**: los datos provenientes de sensores y de dispositivos móviles almacenan las coordenadas geográficas (latitud y longitud) en tiempo real de personas, vehículos u objetos. Algunos ejemplos de consultas que se deben realizar serían:
  - Muestra el recorrido realizado por la persona o el vehículo X en las últimas dos horas.
  - Cuántos camiones de una flota se encuentran en una determinada zona. •
  - Cuáles son las entradas a una ciudad que presentan menor congestión en estos momentos.

    Este tipo de consultas no siguen la estructura clásica de SQL, y por lo tanto, un sistema de gestión de bases de datos relacional no es la mejor opción para derivar esta información en el tiempo requerido

Inconvenientes de las BD NoSQL

- **Madurez**: los SGBD relacionales son sistemas estables con múltiples funcionalidades mientras que las BD NoSQL tienen funcionalidades sin implementar.
- **Especialistas**: hay millones de programadores en el mundo familiarizados con los SGBD relacionales, pero no es tan fácil encontrar especialistas en BD NoSQL.
- **Falta de estándares**: no existe un lenguaje de consulta unificado como SQL.
- **Administración**: las BD NoSQL pueden ser más complejas de instalar y mantener que los SGBD relacionales y la curva de aprendizaje puede ser lenta.

Inconvenientes de las BD NoSQL

- **Soporte**: las empresas especializadas en NoSQL no tienen la envergadura de las empresas que trabajan con SGBD relacionales (Oracle, Microsoft o IBM) y deben garantizar una asistencia técnica rápida y eficiente.
- **Falta de un *benchmark*** definido (todos son los mejores)
- El **esquema de una BD NoSQL puede ser complejo** y no siempre está definido.