Introducción a Bases de Datos NoSQL

Bases de Datos 2023

Base de Datos NoSQL

Las Bases de datos NoSQL son **bases de datos no relacionales**.

- Proveen modelos de datos diferentes al modelo relacional (tablas)
  - Modelo de Documentos, Grafos, ...
- Proveen lenguajes de consultas distintos a SQL
  - MQL, Cipher, CQL, ...

Emergen a fines de la década de 2000 como respuesta a nuevas necesidades de las empresas

- Necesidad de almacenar, acceder y procesar grandes cantidades de conjuntos de datos
- Limitaciones de la bases de datos relacionales tradicionales

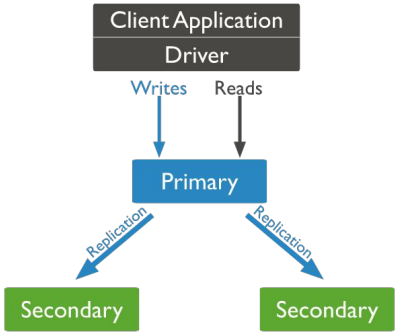
Características de NoSQL

Esquemas Flexibles

- No imponen un esquema predefinido
  - Datos semiestructurados o no estructurados
- Estructuras de datos polimórficas

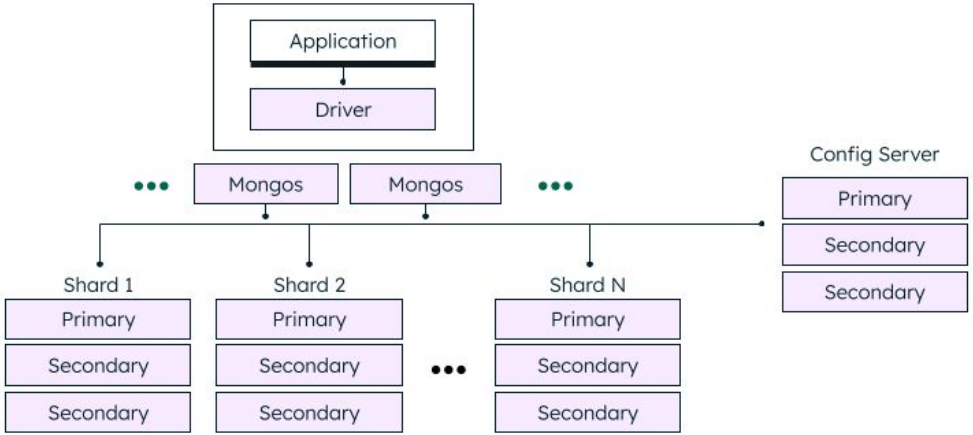
Alta disponibilidad

- Réplica de los datos en varios nodos



Escalabilidad horizontal

- Partición de los datos (sharding)



Características de NoSQL

Integridad Referencial

- Usualmente no proveen FKs
- Delegan la responsabilidad al desarrollador

JOINs

- Usualmente no proveen JOINs o
- Recomendado sólo para consultas de análisis

Desnormalización

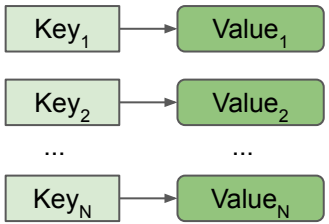
- Es una práctica usual tener duplicados

Modelado de datos Query-driven

- Las consultas son requeridas para diseñar modelos de datos optimizados
- Los requisitos no funcionales (NFRs) de la aplicación son necesarios para el diseño
- Los NFRs son considerados para la selección de la BD más adecuada

Tipos de BD NoSQL: Base de datos Clave-Valor

Modelo de datos



Caracteristicas

- Esquema de datos super flexibles
- In-memory
- Tipos de datos
  - Memcached: Strings
  - Redis: Strings, Hashes, Lists, Sets, Sorted Sets

Lenguaje de consulta: Redis

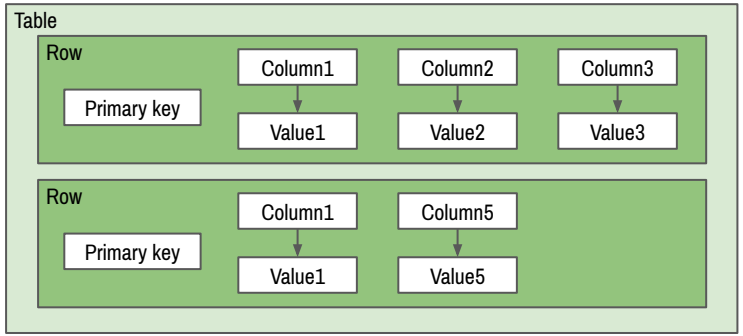
```
> sadd venues "Olympic Stadium" "Nippon Budokan" "Tokyo Stadium"

> sscan venues 0 match *

> sismember venues “Eiffel Tower”
```

Casos de usos

- Almacenar datos en cache (caching)
- Gestión de sesiones
- Deduplicate

Tipos de BD NoSQL: Base de datos Wide-column

Modelo de datos



Caracteristicas

- Agregar columnas con zero downtime
- Tipos de datos
  - Cassandra: incluye native types, collection types, user-defined types, tuple types

Lenguaje de consulta: CQL

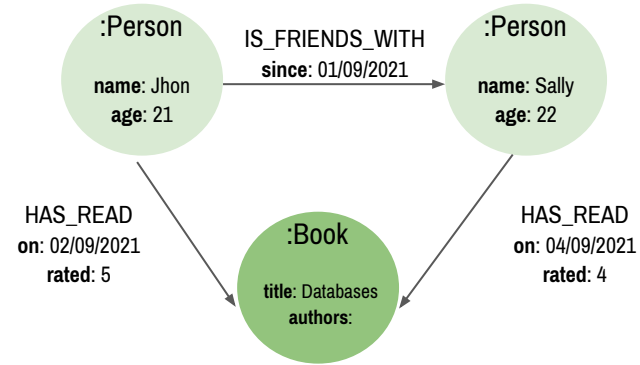
```
ALTER TABLE users ADD todo map<timestamp, text>;
UPDATE users SET todo =   { '2012-9-24' : 'enter mordor',
                  '2014-10-2 12:00' : 'throw ring into mount doom' }
WHERE user_id = 'frodo';

SELECT user_id, todo FROM users WHERE user_id = 'frodo';
```

Casos de usos

- Las escrituras superan a las lecturas
- Logs de transacciones
- IoT

Tipos de BD NoSQL: Base de datos de Grafos

Modelo de datos



Caracteristicas

- Estructuras de grafos: vértices y aristas
- Tipos de datos
  - Neo4j: incluye property types, composite types

Lenguaje de consulta: Cipher

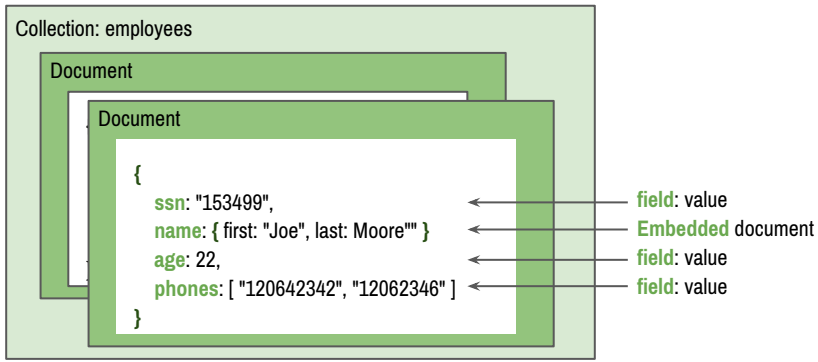
```
//find the top 3 people who have the most friends
MATCH (p:Person)-[r:IS_FRIENDS_WITH]-(other:Person)
RETURN p.name, count(other.name) AS numberOfFriends
ORDER BY numberOfFriends DESC
LIMIT 3
```

Casos de usos

- Redes sociales
- Análisis y detección de fraudes
- Sistema de recomendaciones

Tipos de BD NoSQL: Base de datos de Documentos

Modelo de datos



Caracteristicas

- Estructuras de datos similares a JSON
- Relaciones obvias usando arrays y documentos anidados
- Tipos de datos
  - MongoDB: incluye numbers, strings, date, arrays, objects

Lenguaje de consulta: MQL

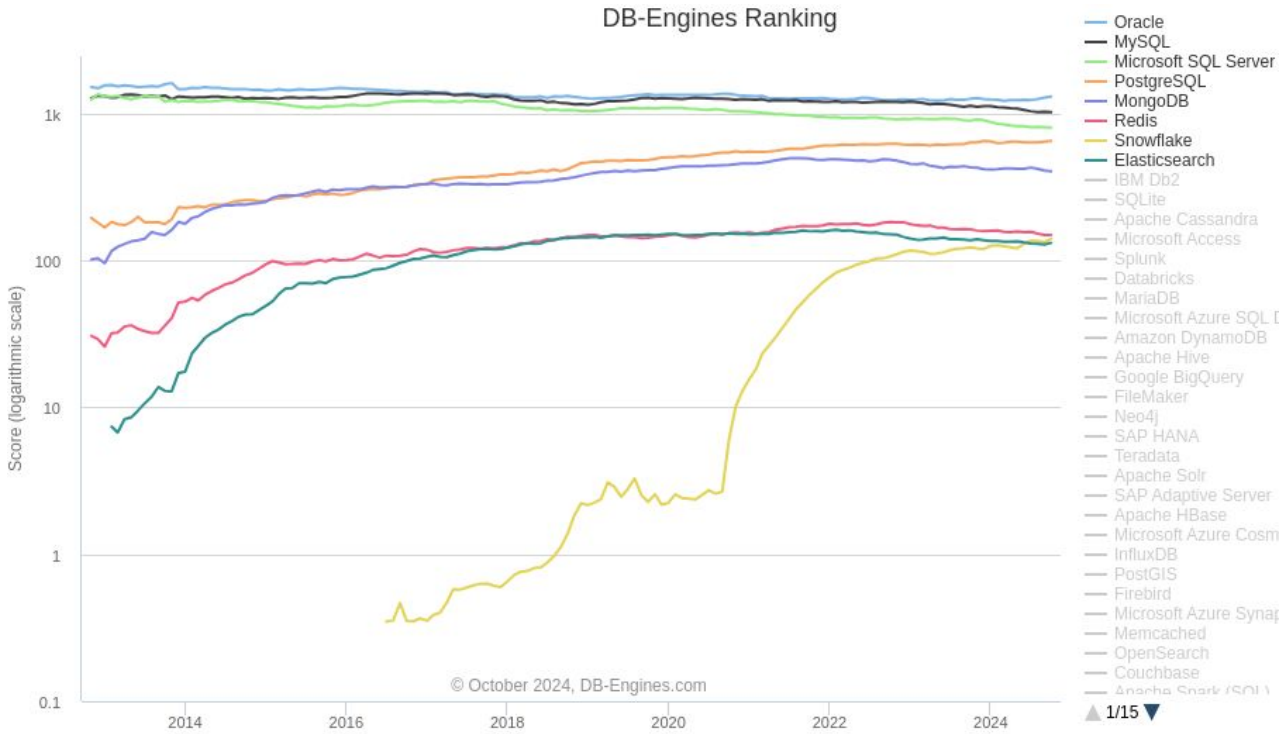
```
//find the employees whose last name starts with letter 'M'

db.employees.find(

      { name.last: /^M/},                //Query filter

      { ssn: 1, name: 1, age: 1 }        //Project fields

).sort( { age: -1 } ).limit( 10 )
```

Casos de usos

- De proposito general
- Catalogos
- IoT
- Gestión de contenidos

DB-Engines Ranking

- 423 bases de datos en el [ranking]



¿Cómo elegir la base de datos adecuada?

https://towardsdatascience.com/how-to-choose-the-right-database-afcf95541741

ACID vs BASE

Diferencia entre una base de datos ACID y otra BASE

- ACID y BASE son modelos de transacciones de bases de datos que determinan cómo una base de datos organiza y manipula los datos.
- Una transacción debe completarse en su totalidad para que la base de datos mantenga su consistencia.
- Las bases de datos ACID dan prioridad a la consistencia sobre la disponibilidad.
  - toda la transacción falla si se produce un error en cualquier paso de la transacción
- Las bases de datos BASE priorizan la disponibilidad sobre la consistencia.
  - En lugar de fallar la transacción, los usuarios pueden acceder temporalmente a datos inconsistentes.
  - La consistencia de los datos se consigue, pero no inmediatamente

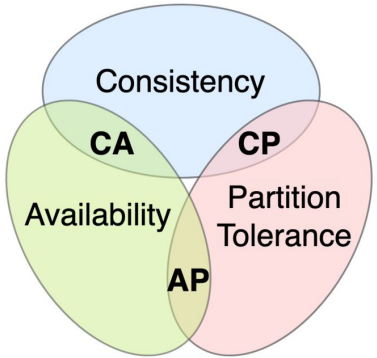
¿Por qué son importantes ACID y BASE?

Las **bases de datos NoSQL** son bases de datos distribuidas que replican los datos en varios nodos conectados por una red.

- Los usuarios esperan que los datos sigan siendo consistentes en todos los nodos al final de la transacción.

El teorema CAP afirma que cualquier base de datos distribuida sólo puede ofrecer dos de las tres propiedades siguientes:

- Consistencia: cada operación de lectura recibe los datos actualizados más recientemente o un error.
- Disponibilidad: cada operación de lectura contiene datos, pero es posible que no sean los más recientes..
- Tolerancia a la partición: el sistema sigue funcionando a pesar de mensajes caídos o retrasados entre los nodos distribuidos.



BASE

BASE es el acronimo de **B**asically **A**vailable, **S**oft state, and **E**ventually consistent.

- Basically Available: es la accesibilidad concurrente de la base de datos por parte de los usuarios en todo momento. Un usuario no necesita esperar a que otros terminen la transacción para actualizar el registro.
- Soft state: permite que los datos sean inconsistente y delega el diseño de las inconsistencia a los desarrolladores de las aplicaciones.
- Eventually consistent: significa que el registro alcanzará la consistencia cuando se hayan completado todas las actualizaciones concurrentes. En ese momento, las aplicaciones que consulten el registro verán el mismo valor.

Comparación entre ACID y BASE

	ACID	BASE
Escalabilidad	Vertical	Horizontal
Flexibilidad	Menos flexible. Bloquea registros específicos de otras aplicaciones al procesarlos.	Más flexible. Permite que varias aplicaciones actualicen el mismo registro simultáneamente.
Performance	El rendimiento disminuye al procesar grandes volúmenes de datos.	Capaz de manejar grandes volúmenes de datos no estructurados con un alto rendimiento.