

Lectura #3 Solución

Kevin Córdoba Chevez - 2020100920

1. **¿Es posible utilizar una base de datos SQL como un abase de datos key-value? ¿Cómo la implementaría? Comente implicaciones de rendimiento.**

Según lo mencionado en la lectura, en una base de datos key-value los items se almacenan como una llave y su valor, donde el valor solo puede ser consultado por dicha llave. Por lo que considero que sí es posible utilizar una base de datos SQL como key-value. Para esto definiría en la tabla que deseo crear un valor "key" o clave que sería a su vez la llave primaria, y otro valor que sería el "value", de modo que pueda realizar consultas por medio del "key".

En cuanto a rendimiento según lo que comprendí de la lectura, el tipo de base key-value permite escalabilidad y fácil optimización en las bases de datos NoSQL, por lo que pienso que no sería tan óptimo ni tan eficiente utilizar este método en una base SQL ya que existen métodos más efectivos y este en concreto tendría tiempos de respuesta más lentos.

2. **¿En qué consisten los datos polimórficos? Explique la razón por la cual estos son un buen caso de uso en bases de datos documentales.**

En la lectura se menciona que al contrario de una base de datos relacional, los esquemas son dinámicos en una base de datos documental gracias a los datos polimórficos.

Estos datos polimórficos son datos que permiten la flexibilidad cuando se modelan datos cuyas estructuras pueden cambiar entre cada registro.

La importancia que tienen estos datos aparte de la flexibilidad que aportan, es que hacen más fácil la evolución de una aplicación durante su ciclo de vida, por ejemplo al añadirle más campos.

3. **Presente 5 ejemplos de sistemas/casos de uso que podrían soportar consistencia eventual, Explique.**

Cuando se presenta consistencia eventual, cuando se realiza una actualización en la base de datos, este no se actualiza de inmediato en las réplicas, sino que lo hace eventualmente.

Algunos ejemplos de este comportamiento son:

- **En plataformas de redes sociales:** según tengo entendido este tipo de consistencia se utiliza en redes sociales, ya que no es necesario que se pueda acceder a una publicación en tiempo real apenas esta es publicada, y no afecta que las réplicas no tengan la publicación en el tiempo exacto.
- **En mensajería:** pienso que en aplicaciones de mensajería como en Telegram o Whatsapp, al participar en un grupo por ejemplo se podría utilizar este tipo de consistencia, de modo que los mensajes vayan llegando poco a poco ayudando a que se mantenga el sentido en la conversación.
- **En páginas web de ventas:** en clase el profesor mencionó un ejemplo sobre que Amazon en algún momento no tenía consistencia entre los productos que mostraban y la cantidad en stock, sin embargo pienso que en ciertos comercios electrónicos dicha consistencia puede ser de utilidad sobre todo en un lugar tan grande como Amazon que cuenta con grandes bodegas, esta ligera desincronización podría dar pie a mayor volumen de ventas.

- **Plataformas estudiantiles:** tomando en consideración el funcionamiento del Tec Digital, pienso que en este tipo de plataformas online se puede utilizar la consistencia eventual, ya que no es necesario que los estudiantes accedan de forma inmediata a la nota de una evaluación, si existe un pequeño retraso entre un estudiante y otro no presentará mayor inconveniente mientras este no se extienda a días.
- **Videojuegos en línea:** por ejemplo los logros en videojuegos en línea, al obtenerse se despliegan en el perfil del usuario, no es necesario que se muestren de manera inmediata, de hecho suelen tomarse un tiempo en aparecer, por lo que lo asocio a la consistencia eventual.

4. **¿Por qué es importante que nativamente una base de datos NoSQL implemente un REST API?**

Según lo que se menciona en la lectura, los RESTful API son de gran importancia debido a que proporcionan simplicidad y familiaridad, además de que se apoyan en la herencia de latencias asociadas a HTTP, se menciona que algunos permiten a los desarrolladores ejecutar operaciones CRUD, usar solicitudes HTTPS e incluso reducir la curva de aprendizaje.

5. **¿Por qué la geo localización de la base de datos NoSQL puede ayudar a mantener leyes de Data sovereignty?**

La definición de soberanía de datos es la siguiente:

"La soberanía de los datos se refiere al concepto de que los datos están sujetos a las leyes y la gobernanza del país y/o región en que se recogen, procesan y almacenan." (SEON, 2023), tomando en cuenta su definición se puede observar que la importancia de la geolocalización de una base de datos NoSQL reside en que le permite a los desarrolladores respetar las leyes de soberanía de datos al saber del lugar del que estos proceden, de este modo los desarrolladores pueden cumplir con las leyes para el almacenamiento y procesamiento de datos según la región en que se encuentren, se reducen riesgos legales y facilita el proceso.

Referencias

SEON (2023). Soberanía de los datos. Recuperado de:

<https://seon.io/es/recursos/glosario/soberania-de-los-datos/>