MODULO 04 - EJERCICIO 03-B

ALEXIS YURI M.

Diseñe la estructura de una base de datos NoSQL para una app de mensajería en tiempo real (tipo WhatsApp). Debe incluir el tipo de base de datos elegido, cómo se modelarían los datos y cómo se escalaría.

1. Determinación del Tipo de Base de Datos NoSQL.

Para una aplicación de mensajería como WhatsApp, el tipo de base de datos NoSQL más adecuado sería documental, ya que permite almacenar mensajes, usuarios y conversaciones como documentos JSON o BSON flexibles y anidados. Cada documento de mensaje puede contener el ID del remitente, el ID del receptor, el contenido, la fecha y hora, y el estado de entrega. Esto simplifica la recuperación de datos para una conversación completa y la adaptación a nuevos tipos de mensajes (por ejemplo, con archivos multimedia o stickers).

2. Esbozar un Modelo de Datos Simplificado.

Un modelo de datos simplificado para una aplicación de mensajería podría consistir en las siguientes colecciones:

- Usuarios: Almacena los perfiles de los usuarios.

Clave: \_id (ID único del usuario).

Valores: Un documento con campos como nombre, telefono, foto\_perfil, estado, y un arreglo de contactos que almacena los IDs de otros usuarios.

- Conversaciones: Almacena los chats grupales o individuales.

Clave: \_id (ID único de la conversación).

Valores: Un documento con participantes (un arreglo de IDs de usuario) y mensajes (un arreglo de documentos anidados).

- Mensajes: Para mayor escalabilidad, los mensajes se podrían almacenar en una colección separada, referenciando la conversación a la que pertenecen.

Clave: \_id (ID único del mensaje).

Valores: Un documento con conversacion\_id, remitente\_id, texto\_contenido, fecha\_envio, estado (por ejemplo, "enviado", "entregado", "leído").

3. Explicación del manejo de la Escalabilidad (Horizontal vs. Vertical).

La escalabilidad horizontal es la más adecuada para una aplicación de mensajería de este tipo.

La escalabilidad horizontal consiste en añadir más servidores a la base de datos para distribuir la carga. Es la opción ideal para una aplicación de mensajería. Esto se logra mediante la fragmentación o sharding, donde los datos se dividen y se distribuyen en múltiples servidores. Por ejemplo, los mensajes podrían fragmentarse por conversacion\_id o por remitente\_id, de modo que los datos de una conversación o de un usuario queden en el mismo servidor o en un grupo de servidores cercanos.

Por otro lado, la escalabilidad vertical consiste en aumentar la capacidad de un solo servidor (más CPU, memoría RAM, HDD). Esta opción es limitada, ya que un solo servidor no puede manejar la carga de millones de usuarios y peticiones por segundo.

4. Resumen.

Cada decisión tomada resuelve los desafíos del caso de la siguiente manera:

Modelo de datos documental: Es un esquema flexible que permite adaptar la estructura de los mensajes para incluir nuevos tipos de contenido (imágenes, audios, documentos) sin necesidad de complejos cambios en el esquema de la base de datos, algo que sería muy costoso en una base de datos relacional.

Diseño de colecciones: Un modelo que separa usuarios, conversaciones y mensajes permite un acceso rápido y eficiente a los datos. Las consultas para una conversación específica pueden optimizarse para recuperar solo los mensajes de esa conversación, sin tener que escanear toda la base de datos de mensajes.

Escalabilidad horizontal: Asegura que la aplicación pueda manejar millones de usuarios. Al distribuir los datos y la carga entre varios servidores, el sistema puede seguir funcionando con baja latencia y alta disponibilidad, incluso si un servidor falla. Es una solución nativa de las bases de datos NoSQL y es crítica para un servicio global y de alta concurrencia como una app de mensajería.