

UNIVERSIDAD DON BOSCO



Asignatura:

Estudiantes:

Jairo Jubynni Soto García

SG080419

Foro 1

Base de datos en entornos móviles

¿Diferencias de base datos SQL y NoSQL?

Modelo de datos:

SQL: Utiliza un modelo relacional, donde los datos se organizan en tablas con filas y columnas. Las relaciones entre las tablas se establecen mediante claves primarias y claves externas.

NoSQL: No se basa en un modelo relacional y puede tener diferentes estructuras de datos según el tipo de base de datos NoSQL. Algunas de las estructuras comunes son documentos, columnas, gráficos y clave-valor.

Escala y rendimiento:

SQL: Las bases de datos SQL son buenas para aplicaciones que requieren transacciones complejas y una fuerte coherencia de datos. Pueden escalar verticalmente agregando más capacidad de hardware a un servidor existente.

NoSQL: Las bases de datos NoSQL están diseñadas para manejar grandes volúmenes de datos y escalabilidad horizontal, lo que significa agregar más servidores a la infraestructura. Son adecuadas para aplicaciones con necesidades de rendimiento y escalabilidad masivas.

Flexibilidad del esquema:

SQL: Utiliza un esquema rígido donde se define la estructura de las tablas y las columnas de antemano. Se requiere que los datos se ajusten a este esquema establecido.

NoSQL: Proporciona flexibilidad en el esquema, lo que significa que los datos no tienen que seguir una estructura fija. Puedes agregar nuevos campos o modificar la estructura de los datos sin afectar la base de datos existente.

Consultas y lenguaje de consulta:

SQL: Utiliza el lenguaje SQL para consultar y manipular los datos. Proporciona una amplia gama de operaciones para recuperar y modificar los datos en las tablas.

NoSQL: Los sistemas NoSQL tienen diferentes formas de consultar los datos, dependiendo del tipo de base de datos. Algunas utilizan un lenguaje de consulta propio, mientras que otras ofrecen una API para acceder y manipular los datos.

Consistencia y disponibilidad:

SQL: En general, las bases de datos SQL priorizan la consistencia de los datos, lo que significa que garantizan que los datos estén siempre en un estado válido según las restricciones definidas. Pueden tener una disponibilidad más limitada en caso de fallas o problemas.

NoSQL: Las bases de datos NoSQL a menudo priorizan la disponibilidad y la tolerancia a fallos. Pueden permitir la escritura y lectura de datos incluso en situaciones de partición de red o fallos, lo que puede llevar a una consistencia eventual.

Estas son solo algunas diferencias generales entre las bases de datos SQL y NoSQL. La elección entre ellas depende de los requisitos específicos de tu aplicación, como la estructura de los datos, la escalabilidad, el rendimiento y la consistencia necesaria.

Firestore y Realtime Database son dos servicios de base de datos en la nube proporcionados por Firebase, una plataforma de desarrollo de aplicaciones móviles y web. A continuación, te mencionaré algunas diferencias clave entre ellos:

Modelo de datos:

Firestore: Utiliza un modelo de documentos y colecciones. Los datos se organizan en documentos (representados en formato JSON) que se agrupan en colecciones. Cada documento tiene un identificador único y los datos se almacenan en forma de pares clave-valor.

Realtime Database: Utiliza un modelo de datos en tiempo real basado en árboles JSON. Los datos se organizan en una única estructura de árbol, y cada nodo en el árbol puede contener datos o subárboles.

Escalabilidad:

Firestore: Proporciona una escalabilidad automática y transparente a medida que aumenta el tamaño de los datos y la carga de usuarios. Puede manejar fácilmente grandes volúmenes de datos y una alta concurrencia.

Realtime Database: También ofrece escalabilidad, pero está más orientado a aplicaciones con un número muy alto de usuarios simultáneos y una baja latencia en las actualizaciones en tiempo real.

Consultas:

Firestore: Permite realizar consultas avanzadas y filtrar los datos en función de múltiples campos y condiciones. Admite consultas en tiempo real y puede realizar búsquedas de texto completo.

Realtime Database: No ofrece la misma flexibilidad en las consultas que Firestore. Las consultas se limitan a ordenar los datos y filtrarlos en función de un solo campo.

Actualizaciones en tiempo real:

Firestore: Proporciona actualizaciones en tiempo real a través de sus suscripciones a cambios en los documentos y las colecciones. Los clientes pueden recibir notificaciones automáticas cuando los datos cambian en el servidor.

Realtime Database: Se centra en las actualizaciones en tiempo real y sincronización en tiempo real entre clientes y servidor. Los cambios en los datos se propagan instantáneamente a todos los dispositivos conectados.

Estructura de costos:

Firestore: Cobra por el número de operaciones de lectura, escritura y eliminación de documentos y por el volumen de almacenamiento de datos.

Realtime Database: Cobra por el volumen de datos almacenados, el ancho de banda utilizado y el número de operaciones de lectura/escritura realizadas.

En general, Firestore es más adecuado para aplicaciones que requieren consultas complejas, escalabilidad y una estructura de datos más flexible. Por otro lado, Realtime Database es ideal para aplicaciones que necesitan una actualización en tiempo real y sincronización instantánea. La elección entre ellos dependerá de las necesidades específicas de tu aplicación.

Conclusiones

Al trabajar en esta oportunidad con el tipo de base de datos SQL y Firestore, se puede concluir que:

- Es importante elegir el tipo de base de datos tomando en cuenta el proyecto con el que se esta trabajando, tomando en cuenta diversos puntos, como el tamaño del proyecto, el tipo de mantenimiento que se pretende dar, los costos, entre otras cosas.
- Para el problema planteado en este foro, se considera que Firestore da un poco de facilidad a la hora de hacer cambios, de agregar o quitar campos, entre otros, y esto puede ser de beneficio para el tiempo de ejecución que se tiene planificado para el proyecto que se está trabajando.
- En cuanto a SQL considero que es un tipo de base de datos mas potente, y con el cual se pueden trabajar bases de datos mas grandes y complejas de una manera mas sencilla, sin embargo, es importa que para trabajar una base de datos SQL es necesario tener gente capacitada para hacerlo.
- También se considera que Firestore puede ser un tipo de base de datos mas sencilla de ser creada por usuarios que no tengan tanta experiencia en el manejo de base de datos.