

Bases de Datos NoSQL - Tipos

1. Diferencia principal entre bases de datos SQL y NoSQL:

- **SQL:** Basadas en un modelo relacional, utilizan tablas con filas y columnas.
- **NoSQL:** No están basadas en el modelo relacional y ofrecen más flexibilidad en cuanto al tema de datos y escalabilidad horizontal.

2. Tres tipos de bases de datos NoSQL:

- **Orientadas a documentos:** Almacenan datos en documentos (generalmente en formato JSON o BSON). Ejemplo: MongoDB.
- **De columna:** Almacenan datos en columnas en lugar de filas, lo que es eficiente para consultas analíticas. Ejemplo: Apache Cassandra.
- **Orientadas a grafos:** Optimizado para manejar relaciones complejas entre datos. Ejemplo: Neo4j.

3. Base de datos orientada a documentos y un ejemplo:

- **Definición:** Almacena datos en documentos que pueden tener una estructura flexible. Los documentos suelen estar en formato JSON, BSON o XML.
 - **Ejemplo:** MongoDB.

4. Bases de datos de columnas y aplicaciones ideales:

- **Ideal para:** Aplicaciones que realizan consultas analíticas y procesan grandes volúmenes de datos, como sistemas de recomendación y análisis de datos en tiempo real.

5. Base de datos orientada a grafos:

- **Definición:** Diseñada para manejar datos que están interconectados mediante relaciones, facilitando consultas complejas sobre las conexiones entre datos.
 - **Ejemplo:** Neo4j.

6. Diferencia entre base de datos clave-valor y orientada a documentos:

- **Clave-valor:** Almacena datos como pares clave-valor, donde cada clave es única y el valor puede ser un dato simple o complejo. Ejemplo: Redis.
- **Orientada a documentos:** Almacena datos en documentos con estructuras más complejas y jerárquicas, permitiendo búsquedas y consultas más sofisticadas. Ejemplo: MongoDB.

7. Ventaja de usar una base de datos NoSQL orientada a grafos:

- **Ventaja principal:** Permiten realizar consultas complejas sobre relaciones entre datos de manera eficiente, ideal para aplicaciones como redes sociales y sistemas de recomendación.

8. Tipo de base de datos NoSQL ideal para estructuras de datos cambiantes:

- **Orientada a documentos:** Ofrecen flexibilidad en el esquema, permitiendo que los documentos tengan estructuras diferentes y cambien con el tiempo.

9. Ventaja y desventaja de las bases de datos orientadas a documentos:

- **Ventaja:** Flexibilidad en la estructura del esquema y capacidad para manejar documentos de diferentes formatos.
- **Desventaja:** Puede ser menos eficiente para realizar consultas que involucren múltiples documentos complejos y relaciones entre ellos.

10. Gestión de escalabilidad en bases de datos NoSQL:

- **Escalabilidad horizontal:** Añadiendo más servidores o nodos para distribuir la carga y el almacenamiento, en lugar de escalar verticalmente aumentando la capacidad de un solo servidor.

Bases de Datos NoSQL - Drivers

MongoDB Native

11. ¿Qué es el driver MongoDB Native?

- Es un paquete de software que permite a los programas interactuar directamente con la base de datos MongoDB desde diferentes lenguajes de programación.

12. Lenguajes de programación disponibles para el driver MongoDB Native:

- **JavaScript (Node.js), Python, Java, C#, C++, Ruby, PHP, entre otros.**

13. Función principal del driver MongoDB Native:

- Proporcionar una interfaz de programación para conectar, consultar y manipular datos en una base de datos MongoDB.

Mongoose

14. ¿Qué es Mongoose y en qué se diferencia del driver MongoDB Native?

- **Mongoose:** Es una biblioteca de modelado de objetos para MongoDB y Node.js que proporciona una capa adicional sobre el driver nativo, facilitando la creación de modelos y esquemas.
- **Diferencia:** Mongoose ofrece un esquema estructurado y una interfaz de alto nivel para manejar datos, mientras que el driver nativo proporciona una interfaz más directa y básica.

15. Ventajas de Mongoose al trabajar con MongoDB:

- Facilita la definición de esquemas, la validación de datos, y proporciona un modelo de objetos para trabajar con datos de manera más estructurada.

16. Definición de un esquema en Mongoose:

- Un esquema en Mongoose define la estructura y los tipos de datos de los documentos en una colección, así como las validaciones y métodos asociados.

Esquemas

17. ¿Para qué sirven los esquemas en bases de datos NoSQL?

- Definen la estructura y formato de los documentos, proporcionando una guía para cómo deben ser almacenados y validados los datos.

18. Importancia de los esquemas en bases de datos orientadas a documentos:

- Aunque son flexibles, los esquemas ayudan a garantizar la consistencia de los datos y facilitan la validación y las consultas.

Modelos

19. ¿Qué es un modelo en el contexto de Mongoose?

- Un modelo en Mongoose es una clase que representa una colección en la base de datos y proporciona una interfaz para interactuar con los documentos en esa colección.

20. ¿Cómo se crean y utilizan los modelos en Mongoose?

- Se crean a partir de un esquema usando `mongoose.model()`, y se utilizan para crear, leer, actualizar y eliminar documentos en la colección asociada.

Documentos

21. ¿Qué es un documento en una base de datos orientada a documentos?

- Es una unidad de datos que se almacena en la base de datos, típicamente en formato JSON o BSON, y puede contener múltiples campos y valores.

22. ¿Cómo se inserta un nuevo documento usando el driver MongoDB Native?

- Usando el método `insertOne()` o `insertMany()` en una colección.

23. ¿Cómo se actualiza un documento existente con Mongoose?

- Usando el método `updateOne()`, `updateMany()` o `findOneAndUpdate()` en un modelo.

24. Diferencia entre un documento y una fila en una base de datos relacional:

- **Documento:** Puede tener una estructura flexible y jerárquica.
- **Fila:** Tiene una estructura fija definida por el esquema de la tabla.

25. Función del método `save` en Mongoose al trabajar con documentos:

- Guarda un documento en la base de datos, actualizando un documento existente si tiene un ID o insertando uno nuevo si no lo tiene.

26. ¿Cómo se pueden consultar documentos por su ID usando Mongoose?

- Usando el método `findById()` o `findOne({_id: id})` en un modelo.

27. ¿Es posible embeber documentos dentro de otros documentos en MongoDB?

- Sí, MongoDB permite anidar documentos dentro de otros documentos, creando estructuras jerárquicas.

28. ¿Cómo se eliminan documentos en MongoDB utilizando Mongoose?

- Usando los métodos `deleteOne()`, `deleteMany()`, o `findByIdAndDelete()` en un modelo.

29. ¿Qué son los operadores de consulta en MongoDB y cómo se utilizan para filtrar documentos?

- Son herramientas que permiten especificar criterios de búsqueda complejos. Ejemplos incluyen `$eq`, `$gt`, `$lt`, `$in`, entre otros.

30. ¿Cómo se manejan las relaciones entre documentos en MongoDB?

- Mediante referencias (almacenando el ID de un documento en otro) o embebiendo documentos (anidando documentos dentro de otros).

Bases de Datos NoSQL - Herramientas

Moon Modeler

31. ¿Para qué sirve Moon Modeler?

- Es una herramienta para diseñar bases de datos NoSQL, permitiendo la creación de esquemas y la visualización de la estructura de datos.

32. Ventajas de utilizar Moon Modeler:

- Facilita el diseño visual de esquemas, la generación de scripts de importación/exportación y la documentación de la base de datos.

33. Tipos de diagramas que permite crear Moon Modeler:

- Diagramas de entidades-relaciones, diagramas de esquemas de MongoDB, y otros.

DBeaver

34. ¿Qué es DBeaver y para qué plataformas está disponible?

- DBeaver es una herramienta de administración de bases de datos que soporta múltiples sistemas de gestión de bases de datos. Está disponible para Windows, macOS y Linux.

35. Tipos de bases de datos compatibles con DBeaver:

- Relacionales (como MySQL, PostgreSQL) y NoSQL (como MongoDB, Cassandra).

36. Funcionalidades destacadas de DBeaver:

- Interfaz gráfica, soporte para múltiples bases de datos, herramientas de diseño y administración, y consultas SQL.

Adminer

37. ¿Qué es Adminer y cuál es su principal uso?

- Adminer es una herramienta de administración de bases de datos escrita en PHP, usada para gestionar bases de datos SQL.

****38.** Lenguajes de programación en los que está