

MONGODB



Laura Fernández Suárez

ÍNDICE

1. ¿Qué son las Bases de Datos NoSQL?	2
1.1. Comparación State-of-the-Art	3
2. MongoDB	4
2.1. Arquitectura	5
2.2. Modelo de datos	6
2.3. Consultas y agregaciones	6
2.4. Índices	6
2.5. Transacciones	6
2.6. Transacciones	6
2.7. Transacciones	7
2.8. Comunidad y soporte	7
3. Ejemplo de MongoDB + JSON:	8

1. ¿Qué son las Bases de Datos NoSQL?

Las bases de datos NoSQL son sistemas de gestión de bases de datos que difieren significativamente de las bases de datos relacionales tradicionales. A diferencia de las bases de datos relacionales, las bases de datos NoSQL no utilizan una estructura de tabla fija y no requieren una definición de esquema previa para almacenar datos. En su lugar, utilizan modelos de datos más flexibles que permiten la gestión de grandes volúmenes de datos no estructurados o semiestructurados.

Las bases de datos NoSQL son populares entre las empresas que necesitan manejar grandes cantidades de datos no estructurados o semi-estructurados, así como entre los desarrolladores que necesitan escalabilidad y flexibilidad en sus aplicaciones. Aunque las bases de datos NoSQL tienen algunas limitaciones, como la falta de soporte para consultas complejas y transacciones, se han convertido en una alternativa popular a las bases de datos relacionales tradicionales debido a su escalabilidad y flexibilidad.

Entre los modelos de datos NoSQL más comunes se incluyen los siguientes:

- ➔ Bases de datos de documentos: Almacenan datos en documentos en formato JSON o BSON, en los cuales cada documento representa un registro y puede contener campos con valores estructurados o no estructurados. Ejemplos de bases de datos de documentos son MongoDB, Couchbase y CouchDB.
- ➔ Bases de datos de clave-valor: Almacenan datos en pares de clave-valor, donde la clave es una cadena de caracteres que identifica el registro y el valor puede ser cualquier objeto, como una cadena de caracteres, un número, una lista o un objeto JSON. Ejemplos de bases de datos de clave-valor son Redis, Riak y DynamoDB.
- ➔ Bases de datos de columnas: Almacenan datos en columnas en lugar de filas, lo que les permite manejar grandes volúmenes de datos de manera más eficiente y con un rendimiento superior. Las columnas se organizan en grupos de columnas llamados familias de columnas. Ejemplos de bases de datos de columnas son Cassandra, HBase y Bigtable.
- ➔ Bases de datos de grafos: Almacenan datos en nodos y relaciones entre ellos en un grafo. Los nodos representan entidades como personas, lugares o cosas, mientras que las relaciones representan conexiones entre ellos. Las bases de datos de grafos permiten realizar consultas y análisis complejos de datos relacionales. Ejemplos de bases de datos de grafos son Neo4j, ArangoDB y OrientDB.

1.1. Comparación State-of-the-Art

Las bases de datos NoSQL se han vuelto cada vez más populares en los últimos años debido a su capacidad para manejar grandes volúmenes de datos no estructurados. Aquí hay una comparación state-of-the-art de algunas de las bases de datos NoSQL más populares:

➔ MongoDB

MongoDB es una base de datos de documentos muy popular que es altamente escalable y flexible. Utiliza un modelo de datos de documentos que permite almacenar datos en un formato semiestructurado como JSON o BSON. MongoDB es compatible con muchos lenguajes de programación y tiene una comunidad muy activa. Es muy popular entre los desarrolladores web y móviles debido a su capacidad para manejar grandes volúmenes de datos y su capacidad de escalamiento horizontal.

➔ **Cassandra**

Cassandra es una base de datos de columnas altamente escalable que fue desarrollada por Facebook. Utiliza un modelo de datos de columnas que permite un procesamiento más rápido de datos y una mayor escalabilidad. Cassandra es muy popular entre las empresas que necesitan manejar grandes cantidades de datos distribuidos en múltiples centros de datos. Cassandra es conocida por su capacidad de tolerancia a fallos y su alta disponibilidad.

➔ **Redis**

Redis es una base de datos de clave-valor en memoria que es muy rápida y escalable. Redis es popular para el almacenamiento en caché y la gestión de sesiones en aplicaciones web. Redis también tiene características de procesamiento de datos en tiempo real, lo que lo hace popular entre los sistemas de juegos y publicidad en línea. Redis es altamente configurable y tiene una sintaxis fácil de usar, lo que lo hace popular entre los desarrolladores.

➔ **Neo4j**

Neo4j es una base de datos de grafos muy popular que es altamente escalable y flexible. Utiliza un modelo de datos de grafos que permite modelar y analizar relaciones complejas entre datos. Es popular entre las empresas que necesitan modelar relaciones de datos en tiempo real, como la banca, la ciencia de la vida y la logística. Neo4j es altamente configurable y tiene una sintaxis fácil de usar, lo que lo hace popular entre los desarrolladores.

➔ **Couchbase**

Couchbase es una base de datos de documentos altamente escalable que es compatible con un modelo de datos de documentos y un modelo de datos de clave-valor. Utiliza una arquitectura de nube distribuida y ofrece características de alta disponibilidad y replicación de datos. Couchbase es popular entre las empresas que necesitan manejar grandes volúmenes de datos distribuidos en múltiples centros de datos.

En resumen, las bases de datos NoSQL ofrecen una alternativa flexible y escalable a las bases de datos relacionales tradicionales. Cada modelo de datos NoSQL tiene sus propias fortalezas y debilidades, por lo que es importante elegir la base de datos adecuada para el caso de uso específico. Al elegir la base de datos correcta, se puede mejorar el rendimiento, la escalabilidad y la eficiencia en general de una aplicación o sistema de base de datos.

2. MongoDB

MongoDB es un sistema de gestión de bases de datos (DBMS) de código abierto y NoSQL. Fue desarrollado por la empresa MongoDB Inc. y su primera versión fue lanzada en 2009. Es una de las bases de datos NoSQL más populares en la actualidad, utilizada por empresas y organizaciones de todo el mundo.

Características principales:

- ➔ Es una base de datos orientada a documentos, lo que significa que almacena los datos en forma de documentos BSON (Binary JSON), que son similares a los objetos JSON (JavaScript Object Notation). Cada documento puede contener diferentes campos y tipos de datos.

- ➔ MongoDB no utiliza esquemas predefinidos como lo hacen las bases de datos relacionales. Esto significa que los desarrolladores pueden añadir o modificar campos en cualquier momento sin necesidad de realizar cambios en la estructura de la base de datos.
- ➔ Es escalable y flexible, lo que significa que puede manejar grandes volúmenes de datos y puede escalar horizontalmente a medida que crecen los datos.
- ➔ MongoDB soporta consultas complejas y ofrece una amplia variedad de operadores para realizar consultas. También proporciona herramientas para el análisis de datos y la agregación de datos.
- ➔ MongoDB es multiplataforma, lo que significa que se puede utilizar en diferentes sistemas operativos, incluyendo Windows, Linux y Mac OS.
- ➔ Ofrece una alta disponibilidad y replicación de datos para asegurar la disponibilidad de los datos y la tolerancia a fallos.
- ➔ MongoDB es compatible con la mayoría de los lenguajes de programación, lo que facilita la integración con otras aplicaciones y sistemas.

Ventajas:

- ➔ MongoDB es altamente escalable y puede manejar grandes volúmenes de datos.
- ➔ Es flexible y fácil de usar, ya que no requiere esquemas predefinidos.
- ➔ Ofrece una alta disponibilidad y replicación de datos para asegurar la disponibilidad de los datos y la tolerancia a fallos.
- ➔ MongoDB es compatible con la mayoría de los lenguajes de programación, lo que facilita la integración con otras aplicaciones y sistemas.
- ➔ Es una base de datos de código abierto, lo que significa que es gratuito y tiene una gran comunidad de desarrolladores que pueden proporcionar soporte y ayuda.

Desventajas:

MongoDB puede tener un rendimiento inferior en comparación con las bases de datos relacionales en algunas situaciones específicas, como en consultas complejas y en operaciones que involucren transacciones.

MongoDB puede requerir más recursos de hardware y de memoria para mantener un rendimiento adecuado.

MongoDB no es compatible con ACID (Atomicity, Consistency, Isolation, Durability), que es un conjunto de propiedades que garantizan que las transacciones se procesen de manera confiable.

En conclusión, MongoDB es una base de datos NoSQL orientada a documentos, altamente escalable y flexible, que ofrece una amplia variedad de herramientas para consultas y análisis de datos. Sin embargo, también tiene algunas desventajas en cuanto a rendimiento y falta de compatibilidad con ACID. En general, es una excelente opción para aplicaciones que requieren alta escalabilidad y flexibilidad en la gestión de datos.

2.1. Arquitectura

MongoDB sigue una arquitectura distribuida que permite escalar horizontalmente, lo que significa que los datos se distribuyen en varios servidores en lugar de un solo servidor. Los datos se almacenan en colecciones que pueden tener diferentes documentos BSON. Cada colección está alojada en un nodo o servidor, y los nodos se agrupan en conjuntos de réplicas. Los conjuntos de réplicas permiten que los datos se repliquen en múltiples nodos y proporcionan tolerancia a fallos y alta disponibilidad.

2.2. Modelo de datos

MongoDB utiliza un modelo de datos orientado a documentos. Un documento es similar a una fila en una tabla en una base de datos relacional, pero en lugar de usar una estructura rígida de columnas, los documentos pueden tener campos y valores arbitrarios. MongoDB no tiene esquemas fijos, lo que significa que los desarrolladores pueden agregar, cambiar o eliminar campos de documentos individuales o colecciones completas sin tener que modificar la estructura de la base de datos.

2.3. Consultas y agregaciones

MongoDB utiliza un lenguaje de consulta llamado Query Language, que proporciona una variedad de operadores para filtrar, ordenar y limitar los documentos que se devuelven en una consulta. Además, MongoDB ofrece la posibilidad de realizar agregaciones complejas que permiten hacer análisis de datos y calcular estadísticas.

2.4. Índices

MongoDB utiliza índices para acelerar el rendimiento de las consultas y las operaciones de actualización. Los índices se crean en campos específicos y permiten que MongoDB encuentre documentos más rápidamente. MongoDB soporta varios tipos de índices, como índices de un solo campo, índices compuestos e índices geoespaciales.

2.5. Transacciones

A partir de la versión 4.0, MongoDB admite transacciones, que permiten a los desarrolladores agrupar operaciones y ejecutarlas como una sola unidad atómica. Las transacciones garantizan la integridad de los datos y la consistencia de las operaciones. Las transacciones en MongoDB funcionan en el nivel de colección y son compatibles con los conjuntos de réplicas y la escalabilidad horizontal.

2.6. Transacciones

A partir de la versión 4.0, MongoDB admite transacciones, que permiten a los desarrolladores agrupar operaciones y ejecutarlas como una sola unidad atómica. Las transacciones garantizan la integridad de los datos y la consistencia de las operaciones. Las transacciones en MongoDB funcionan en el nivel de colección y son compatibles con los conjuntos de réplicas y la escalabilidad horizontal.

2.7. Transacciones

A partir de la versión 4.0, MongoDB admite transacciones, que permiten a los desarrolladores agrupar operaciones y ejecutarlas como una sola unidad atómica. Las transacciones garantizan la integridad de los datos y la consistencia de las operaciones. Las transacciones en MongoDB funcionan en el nivel de colección y son compatibles con los conjuntos de réplicas y la escalabilidad horizontal.

2.8. Comunidad y soporte

MongoDB es una base de datos de código abierto con una gran comunidad de desarrolladores que proporcionan soporte y contribuyen con mejoras y nuevas funcionalidades. MongoDB Inc. ofrece servicios de soporte y consultoría para empresas que necesitan ayuda adicional.

3. Ejemplo de MongoDB + JSON:

JSON:

```
{
  "name": "Camiseta",
  "description": "Camiseta de algodón suave",
  "price": 29.99,
  "category": "Ropa"
}
```

En este ejemplo, el documento JSON tiene cuatro propiedades: “name”, “description”, “price” y “category”. Cada propiedad tiene un valor asociado. Por ejemplo, la propiedad name tiene el valor “Camiseta”, la propiedad description tiene el valor “Camiseta de algodón suave”, y así sucesivamente.

MONGODB:

```
const MongoClient = require('mongodb').MongoClient;
const url = 'mongodb://localhost:27017';
const dbName = 'mydatabase';
```

```
MongoClient.connect(url, function(err, client) {
  if (err) throw err;
  const db = client.db(dbName);
  const collection = db.collection('products');
  const product = {
    "name": "Camiseta",
    "description": "Camiseta de algodón suave",
    "price": 29.99,
    "category": "Ropa"
  };
  collection.insertOne(product, function(err, result) {
    if (err) throw err;
    console.log('Producto insertado con éxito');
    client.close();
  });
});
```

Este código conecta con nuestra base de datos MongoDB, crea una colección llamada “products”, inserta el documento JSON del producto en la colección y luego cierra la conexión.