

Conceptos básicos de MongoDB

Tomado del sitio [Guía para principiantes: Conceptos básicos de MongoDB | MongoDB](#)

Aspectos clave de MongoDB

Estos son algunos de los conceptos y términos clave que encontrará a medida que aprenda sobre MongoDB.

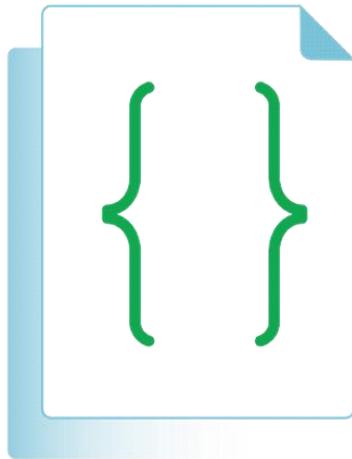
- - **Documentos: Los registros de una base de datos de documentos**
 - Colecciones: Agrupación de documentos
 - Conjuntos de réplicas: garantizar una alta disponibilidad
 - Sharding: escalabilidad para manejar el crecimiento masivo de datos
 - Índices: Mejora de la velocidad de consulta
 - Canalizaciones de agregación: flujos de datos rápidos
 - Lenguajes de programación: ¿MongoDB habla su idioma?
 - Cómo monitorear MongoDB
 - Nube de MongoDB

Documentos: Los registros de una base de datos de documentos

MongoDB almacena los datos como documentos JSON.

El modelo de datos de documento se asigna de forma natural a los objetos del código de la aplicación, lo que facilita el aprendizaje y el uso por parte de los desarrolladores. Los campos de un documento JSON pueden variar de un documento a otro. Compare eso con una tabla de base de datos relacional tradicional, donde agregar un campo significa agregar una columna a la propia tabla de la base de datos y, por lo tanto, a todos los registros de la base de datos.

Los documentos se pueden anidar para expresar relaciones jerárquicas y almacenar estructuras como arreglos



```
{  
  "model": "Volvo C70",  
  "year": 2007,  
  "bodyStyle": [  
    "coupe",  
    "convertible"  
  ],  
  "engine": {  
    "model": "D5",  
    "power": "178hp"  
  }  
}
```

El modelo de documento proporciona flexibilidad para trabajar con datos complejos, cambiantes y desordenados de numerosas fuentes. Permite a los desarrolladores ofrecer rápidamente nuevas funcionalidades de aplicaciones.

Para un acceso interno más rápido y para admitir más tipos de datos, MongoDB convierte los documentos en un formato llamado JSON binario o BSON. Pero desde la perspectiva del desarrollador, MongoDB es una base de datos JSON.

¿Qué es JSON? ([JSON y BSON | MongoDB](#))

JSON, o JavaScript Object Notation, es un formato de intercambio de datos legible por humanos, especificado a principios de la década de 2000. Aunque JSON se basa en un subconjunto del estándar del lenguaje de programación JavaScript, es completamente independiente del lenguaje.

Los objetos JSON son contenedores asociativos, en los que una clave de cadena se asigna a un valor (que puede ser un número, una cadena, un valor booleano, una matriz, un valor vacío nulo o incluso otro objeto). Casi cualquier lenguaje de programación tiene una implementación para esta estructura de datos abstracta: objetos en JavaScript, diccionarios en Python, tablas hash en Java y C#, matrices asociativas en C++, etc. Los objetos JSON son fáciles de entender para los humanos y para que las máquinas los analicen y generen:

```
{  
    "_id": 1,  
    "name": { "first" : "John", "last" : "Backus" },  
    "contribs": [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],  
    "awards": [  
        {  
            "award": "W.W. McDowell Award",  
            "year": 1967,  
            "by": "IEEE Computer Society"  
        }, {  
            "award": "Draper Prize",  
            "year": 1993,  
            "by": "National Academy of Engineering"  
        }  
    ]  
}
```

La conexión MongoDB-JSON

MongoDB fue diseñado desde sus inicios para ser una base de datos centrada en ofrecer una gran experiencia de desarrollo. La ubicuidad de JSON lo convirtió en la opción obvia para representar estructuras de datos en el modelo de datos de documentos de MongoDB.

Sin embargo, hay varios problemas que hacen que JSON no sea ideal para su uso dentro de una base de datos.

- - JSON solo admite un número limitado de tipos de datos básicos. En particular, JSON carece de compatibilidad con fechas y datos binarios.
 - Los objetos y propiedades JSON no tienen una longitud fija, lo que hace que el recorrido sea más lento.

Con el fin de hacer que MongoDB JSON sea lo primero, pero que siga siendo de alto rendimiento y propósito general, BSON se inventó para cerrar la brecha: una representación binaria para almacenar datos en formato JSON, optimizada para la velocidad, el espacio y la eficiencia. No es muy diferente de otros formatos de intercambio binario como Protocol Buffers o Thrift, en términos de enfoque.

¿Qué es BSON?

BSON significa "JSON binario", y eso es exactamente para lo que se inventó. La estructura binaria de BSON codifica la información de tipo y longitud, lo que permite que se recorra mucho más rápido en comparación con JSON.

BSON agrega algunos tipos de datos no nativos de JSON, como fechas y datos binarios, sin los cuales MongoDB habría perdido un soporte valioso.

A continuación se muestran algunos ejemplos de objetos JSON y sus correspondientes representaciones BSON.

```
{"hello": "world"} →  
\x16\x00\x00\x00          // total document size  
\x02                      // 0x02 = type String  
hello\x00                  // field name  
\x06\x00\x00\x00world\x00  // field value  
\x00                      // 0x00 = type E00 ('end of object')  
  
{"BSON": ["awesome", 5.05, 1986]} →  
\x31\x00\x00\x00  
\x04BSON\x00  
\x26\x00\x00\x00  
\x02\x30\x00\x08\x00\x00\x00awesome\x00  
\x01\x31\x00\x33\x33\x33\x33\x33\x33\x14\x40  
\x10\x32\x00\xc2\x07\x00\x00  
\x00  
\x00
```

¿MongoDB usa BSON o JSON?

MongoDB almacena datos en formato BSON tanto internamente como a través de la red, pero eso no significa que no pueda pensar en MongoDB como una base de datos JSON. Cualquier cosa que pueda representar en JSON se puede almacenar de forma nativa en MongoDB y recuperarse con la misma facilidad en JSON.

Al usar el controlador MongoDB para su lenguaje de programación favorito, trabaja con las estructuras de datos nativas para ese lenguaje. El controlador se encargará de convertir los datos a BSON y viceversa al consultar la base de datos.

A diferencia de los sistemas que almacenan JSON como valores codificados en cadenas, o blobs codificados en binarios, MongoDB utiliza BSON para ofrecer potentes funciones de indexación y consulta sobre el formato de datos más popular de la web.

Por ejemplo, MongoDB permite a los desarrolladores consultar y manipular objetos mediante claves específicas dentro del documento JSON/BSON, incluso en documentos anidados con muchas capas de profundidad en un registro, y crear índices de alto rendimiento en esas mismas claves y valores.

- - Los documentos BSON pueden contener objetos Date o Binary que no se pueden representar de forma nativa en JSON puro.

- Cada lenguaje de programación tiene su propia semántica de objetos. Los objetos JSON tienen claves ordenadas, por ejemplo, mientras que los diccionarios de Python (la estructura de datos nativa más cercana que es análoga a los objetos de JavaScript) no están ordenados, mientras que las diferencias en los tipos de datos numéricos y de cadena también pueden entrar en juego.
- BSON admite una variedad de tipos numéricos que no son nativos de JSON, y muchos lenguajes los representan de manera diferente.

Flexibilidad de esquemas y gobernanza de datos

Uno de los grandes atractivos para los desarrolladores que utilizan bases de datos con modelos de datos JSON y BSON es el esquema dinámico y flexible que proporcionan en comparación con los modelos de datos rígidos y tabulares utilizados por las bases de datos relacionales.

- - Los documentos de MongoDB son polimórficos: los campos pueden variar de un documento a otro dentro de una sola colección (análogo a las tablas de una base de datos relacional). Esta flexibilidad facilita el modelado de datos de cualquier estructura y la adaptación del modelo a medida que cambian los requisitos.
 - No es necesario declarar la estructura de los documentos en la base de datos: los documentos son autodescriptivos. Los desarrolladores pueden empezar a escribir código y conservar los objetos a medida que se crean.
 - Si es necesario agregar un nuevo campo a un documento, se puede crear sin afectar a todos los demás documentos de la colección, sin actualizar un catálogo del sistema central y sin desconectar la base de datos. Cuando necesita realizar cambios en el modelo de datos, la base de datos de documentos sigue almacenando los objetos actualizados sin necesidad de realizar operaciones costosas o, lo que es peor, sin tener que rediseñar el esquema desde cero. ALTER TABLE

A través de estas ventajas, la flexibilidad del modelo de datos de documentos se adapta bien a las demandas de las prácticas modernas de desarrollo de aplicaciones. Aunque un esquema flexible es una característica eficaz, hay situaciones en las que es posible que desee tener más control sobre la estructura de datos y el contenido de los documentos. La mayoría de las bases de datos de documentos devuelven la aplicación de estos controles al desarrollador para que los implemente en el código de la aplicación. Sin embargo, las bases de datos de documentos más avanzadas proporcionan validación de esquemas, utilizando enfoques como el estándar IETF JSON Schema adoptado por MongoDB.

Aspectos clave de MongoDB

Estos son algunos de los conceptos y términos clave que encontrará a medida que aprenda sobre MongoDB.

- - Documentos: Los registros de una base de datos de documentos
 - **Colecciones: Agrupación de documentos**
 - Conjuntos de réplicas: garantizar una alta disponibilidad
 - Sharding: escalabilidad para manejar el crecimiento masivo de datos
 - Índices: Mejora de la velocidad de consulta
 - Canalizaciones de agregación: flujos de datos rápidos
 - Lenguajes de programación: ¿MongoDB habla su idioma?
 - Cómo monitorear MongoDB
 - Nube de MongoDB

Colecciones: Agrupación de documentos

En MongoDB, una colección es un grupo de documentos.

Si está familiarizado con las bases de datos relacionales, puede pensar en una colección como una tabla. Pero las colecciones en MongoDB son mucho más flexibles. Las colecciones no aplican un esquema y los documentos de la misma colección pueden tener campos diferentes.

Cada colección está asociada a una base de datos MongoDB. Para mostrar qué colecciones se encuentran en una base de datos determinada, utilice el comando `listCollections` ([listCollections — MongoDB Manual](#))

```
sample_mflix> db.runCommand( { listCollections: 1.0, nameOnly: true } )
{
  cursor: {
    id: Long("0"),
    ns: 'sample_mflix.$cmd.listCollections',
    firstBatch: [
      { name: 'theaters', type: 'collection' },
      { name: 'users', type: 'collection' },
      { name: 'comments', type: 'collection' },
      { name: 'movies', type: 'collection' },
      { name: 'sessions', type: 'collection' }
    ],
    ok: 1
  }
sample_mflix>
```

Aspectos clave de MongoDB

Estos son algunos de los conceptos y términos clave que encontrará a medida que aprenda sobre MongoDB.

- - Documentos: Los registros de una base de datos de documentos
 - Colecciones: Agrupación de documentos
 - **Conjuntos de réplicas: garantizar una alta disponibilidad**
 - Sharding: escalabilidad para manejar el crecimiento masivo de datos
 - Índices: Mejora de la velocidad de consulta
 - Canalizaciones de agregación: flujos de datos rápidos
 - Lenguajes de programación: ¿MongoDB habla su idioma?
 - Cómo monitorear MongoDB
 - Nube de MongoDB

Conjuntos de réplicas: garantizar una alta disponibilidad

Una forma importante de garantizar una alta disponibilidad es mantener más de una copia de los datos. Con MongoDB, la alta disponibilidad está integrada en el diseño.

Un conjunto de réplicas ([What Is Replication In MongoDB? | MongoDB](#)) es un grupo de al menos tres instancias de MongoDB que replican continuamente los datos entre ellas, lo que ofrece redundancia y protección contra el tiempo de inactividad ante un fallo del sistema o un mantenimiento planificado.

Aspectos clave de MongoDB

Estos son algunos de los conceptos y términos clave que encontrará a medida que aprenda sobre MongoDB.

- - Documentos: Los registros de una base de datos de documentos
 - Colecciones: Agrupación de documentos
 - Conjuntos de réplicas: garantizar una alta disponibilidad
 - **Sharding: escalabilidad para manejar el crecimiento masivo de datos**
 - Índices: Mejora de la velocidad de consulta
 - Canalizaciones de agregación: flujos de datos rápidos
 - Lenguajes de programación: ¿MongoDB habla su idioma?
 - Cómo monitorear MongoDB
 - Nube de MongoDB

Sharding: escalabilidad para manejar el crecimiento masivo de datos

Una plataforma de datos moderna debe ser capaz de manejar consultas muy rápidas y conjuntos de datos masivos utilizando grupos cada vez más grandes de máquinas pequeñas. La fragmentación es el término para distribuir datos de forma inteligente entre varias máquinas.

¿Cómo funciona la fragmentación en MongoDB? MongoDB fragmenta los datos en el nivel de colección, distribuyendo los documentos de una colección entre las particiones de un clúster. El resultado es una arquitectura de escalabilidad horizontal que admite incluso las aplicaciones más grandes.



Aspectos clave de MongoDB

Estos son algunos de los conceptos y términos clave que encontrará a medida que aprenda sobre MongoDB.

- - Documentos: Los registros de una base de datos de documentos
 - Colecciones: Agrupación de documentos
 - Conjuntos de réplicas: garantizar una alta disponibilidad
 - Sharding: escalabilidad para manejar el crecimiento masivo de datos
 - **Índices: Mejora de la velocidad de consulta**

- Canalizaciones de agregación: flujos de datos rápidos
- Lenguajes de programación: ¿MongoDB habla su idioma?
- Cómo monitorear MongoDB
- Nube de MongoDB

Índices: Mejora de la velocidad de consulta

Los índices admiten la ejecución eficaz de consultas. MongoDB ofrece una variedad de estrategias de indexación diferentes, incluidos índices compuestos en múltiples campos. Si se eligen cuidadosamente, los índices aceleran las consultas, ya que las consultas examinan el índice en lugar de leer todos los documentos de la colección.

Todavía queda trabajo por hacer para analizar qué consultas podrían beneficiarse de la adición de un índice. Una herramienta que realiza este análisis por usted es el Asesor de rendimiento, que analiza las consultas y sugiere índices que mejorarían el rendimiento de las consultas.

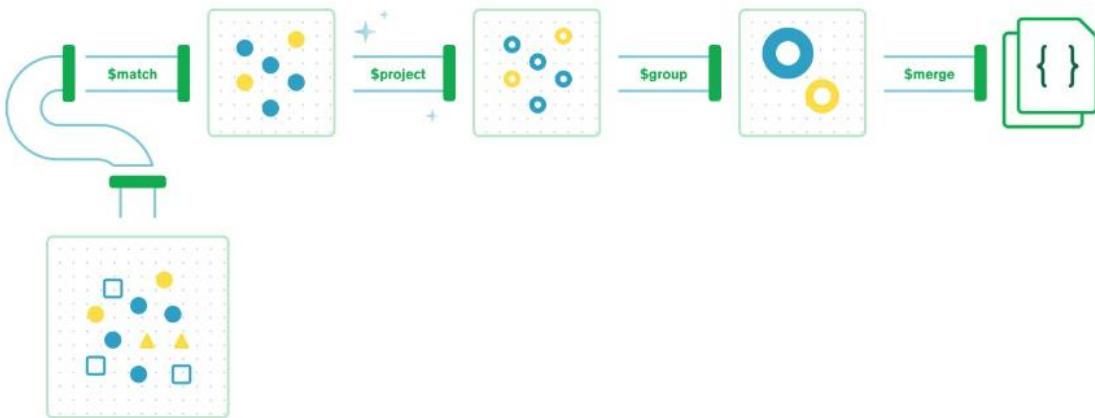
Aspectos clave de MongoDB

Estos son algunos de los conceptos y términos clave que encontrará a medida que aprenda sobre MongoDB.

- - Documentos: Los registros de una base de datos de documentos
 - Colecciones: Agrupación de documentos
 - Conjuntos de réplicas: garantizar una alta disponibilidad
 - Sharding: escalabilidad para manejar el crecimiento masivo de datos
 - Índices: Mejora de la velocidad de consulta
 - ***Canalizaciones de agregación: flujos de datos rápidos***
 - ***Lenguajes de programación: ¿MongoDB habla su idioma?***
 - ***Cómo monitorear MongoDB***
 - ***Nube de MongoDB***

Pipelines de agregación: flujos de datos rápidos

MongoDB ofrece un marco flexible para crear canalizaciones de procesamiento de datos denominadas canalizaciones de agregación. Cuenta con docenas de etapas y más de 150 operadores y expresiones, lo que le permite procesar, transformar y analizar datos de cualquier estructura a escala. Una adición reciente es la etapa de la Unión, que agrega de manera flexible los resultados de múltiples colecciones.



Lenguajes de programación

¿Qué lenguajes se pueden utilizar con MongoDB? La lista de lenguajes compatibles incluye Node.js, C, C++, C#, Go, Java, Perl, PHP, Python, Ruby, Rust, Scala y Swift. La biblioteca de cada idioma se mantiene activamente, lo que significa que se actualiza con nuevas funciones, correcciones de errores, parches de seguridad y mejoras de rendimiento.

Cómo monitorear MongoDB

Puede supervisar el estado y el rendimiento del clúster comprobando el estado de la instancia, las métricas de operaciones y conexiones del clúster, las métricas de hardware y mucho más con solo unas pocas utilidades y comandos. La supervisión puede ayudar a detectar y reaccionar ante problemas en tiempo real antes de que se vuelvan significativos.

Nube de MongoDB

MongoDB comenzó como una base de datos de código abierto y todavía se puede usar de esa manera a través de MongoDB Community Edition.

En MongoDB Enterprise Edition, las funciones avanzadas están disponibles a través de una licencia comercial.

MongoDB Atlas es una versión de base de datos como servicio de MongoDB Enterprise Edition que se ofrece en todas las nubes públicas.

MongoDB Atlas se ha ampliado de diversas maneras con funciones integradas y estrechamente integradas, como MongoDB Atlas Search, MongoDB Atlas Data Lake y funciones avanzadas para geolocalizar datos y realizar copias de seguridad.

MongoDB Cloud es el nombre de MongoDB Atlas y otros nuevos productos como Realm, un entorno informático sin servidor para crear aplicaciones móviles basadas en MongoDB Atlas.