Taller Mongo

Edwin Camilo Rodriguez Arredondo

ID:841424

Bases de Datos Masiva

6° Semestre Ingeniería de Sistemas

Ingeniería de Sistemas

Universidad UNIMINUTO

21 de marzo de 2025

Zipaquirá, Cundinamarca

Contenido

Introducción

MongoDB es una base de datos de estilo NoSQL, creada para guardar grandes volúmenes de datos en un formato más versátil que las bases de datos convencionales. En vez de emplear tablas y filas como lo hacen sistemas relacionales como MySQL, MongoDB estructura los datos en documentos parecidos a archivos JSON, lo que facilita la adaptación a diversos tipos de datos y estructuras.

Este taller abordará los principios básicos de MongoDB, incluyendo su estructura, sus beneficios en comparación con una base de datos relacional, y la manera de ejecutar operaciones fundamentales como la creación, lectura, actualización y supresión de datos. Además, se tratará el uso de MongoDB con Docker y se mostrarán herramientas gráficas que simplifican su manejo. Este entendimiento facilitará la comprensión de cuándo y de qué manera emplear MongoDB en diferentes clases de proyectos tecnológicos.

Objetivo general

Examinar el funcionamiento y los atributos esenciales de MongoDB como base de datos NoSQL, a través de la comparación con sistemas relacionales, la aplicación de comandos CRUD y su aplicación en ambientes virtualizados con Docker, con el objetivo de potenciar las habilidades en la gestión de tecnologías actuales de almacenamiento de datos.

Objetivos específicos

- Reconocer las diferencias fundamentales entre MongoDB y las bases de datos relacionales convencionales como MySQL.
- Describir la estructura de almacenamiento de información en MongoDB, que incluye la utilización de los formatos JSON y BSON.
- Implementar instrucciones CRUD en MongoDB para administrar colecciones y documentos de forma eficaz.
- Detallar las tácticas para vincular datos en MongoDB sin recurrir a los operadores JOIN.
- Realizar la implementación de MongoDB empleando Docker.
- Identificar instrumentos gráficos que facilitan la representación y gestión de datos en MongoDB.

¿Qué tipo de base de datos es MongoDB y en qué se diferencia de una base de datos relacional como MySQL?

MongoDB es una base de datos NoSQL orientada a documentos. En contraposición a una base de datos relacional como MySQL, MongoDB no emplea tablas ni esquemas establecidos. En su lugar, guarda los datos en documentos de tipo JSON (internamente BSON), proporcionando así una estructura adaptable y dinámica.

Diferencias clave entre MongoDB y MySQL:

MongoDB (NoSQL):

- Es una base de datos no relacional, orientada a documentos.
- Almacena los datos en formato JSON o BSON, no en tablas.
- Tiene una estructura flexible y dinámica, no necesita un esquema fijo.
- Es altamente escalable horizontalmente, ideal para grandes volúmenes de datos.
- Utiliza un lenguaje de consultas propio basado en JSON.
- No utiliza JOINs tradicionales para relacionar datos.
- Tiene menos herramientas de análisis en comparación con MySQL.
- Es muy flexible, ideal para proyectos donde los datos cambian constantemente.

MySQL (SQL):

- Es una base de datos relacional, basada en tablas.
- Organiza la información en filas y columnas con un esquema definido.
- Requiere un modelo de base de datos detallado antes de usarse.
- La escalabilidad es más limitada y compleja (normalmente vertical).
- Utiliza el lenguaje SQL, un estándar ampliamente usado.
- Permite realizar JOINs para relacionar varias tablas.
- Cuenta con una amplia variedad de herramientas de análisis y reportes.
- Es más estructurada, ideal para datos estables y bien definidos.

¿Qué es una colección en MongoDB y en qué se diferencia de una tabla en SQL?

Una colección en MongoDB es un conjunto de documentos que se almacenan juntos dentro de una base de datos. Cada documento tiene una estructura flexible y está basado en el formato BSON (una versión binaria de JSON), lo que permite que los documentos dentro de una misma colección puedan tener campos diferentes, o estructuras distintas, sin seguir un esquema rígido.

En cambio, en una base de datos SQL como MySQL, los datos se almacenan en tablas, que sí tienen una estructura fija. Cada fila en una tabla representa un registro, y cada columna representa un campo, con un tipo de dato específico. Es decir, todas las filas deben seguir exactamente el mismo formato (el mismo número y tipo de columnas).

Principales diferencias:

- Estructura:
  - ✓ Una colección en MongoDB contiene documentos con estructuras flexibles.
  - ✓ Una tabla en SQL contiene filas con estructura fija (todas las filas deben coincidir en columnas y tipos de datos).
- Esquema:
  - ✓ MongoDB no necesita un esquema definido. Puedes tener documentos con diferentes campos dentro de la misma colección.
  - ✓ En SQL, sí necesitas definir un esquema antes de insertar datos (por ejemplo: nombre VARCHAR, edad INT).
- Formato de almacenamiento:
  - ✓ MongoDB usa documentos BSON/JSON.
  - ✓ SQL usa filas y columnas en formato tabular.
- Flexibilidad:
  - ✓ MongoDB es más flexible y dinámico, ideal para datos que cambian constantemente.
  - ✓ SQL es más estructurado y seguro, ideal para datos estables que requieren consistencia.

- Ejemplo sencillo:

✓ En SQL, si tienes una tabla llamada ***USUARIOS***, todos los registros deben tener los mismos campos: ***nombre, edad, correo.***

✓ En MongoDB, una colección llamada ***usuarios*** puede tener un documento con ***nombre y edad***, otro solo con ***nombre y correo***, y otro con ***nombre, edad, correo*** y un campo adicional como ***hobbies.***

¿Cómo se almacena la información en MongoDB y qué formato utiliza?

MongoDB almacena la información en un formato llamado BSON (Binary JSON), que es una representación binaria del formato JSON (JavaScript Object Notation). Este formato permite una mayor eficiencia en el almacenamiento y procesamiento de datos, ya que es más rápido de interpretar por las máquinas y soporta una gama más amplia de tipos de datos, como fechas, enteros de 64 bits y binarios, que JSON por sí solo no permite.

La información en MongoDB se estructura de la siguiente manera:

- **Bases de datos (Databases):** Son los contenedores principales que agrupan conjuntos de datos relacionados.

- **Colecciones (Collections):** Dentro de cada base de datos existen colecciones, que equivalen a las tablas en las bases de datos relacionales.

- **Documentos (Documents):** Son la unidad básica de almacenamiento en MongoDB. Cada documento es una estructura de datos compuesta por pares **clave-valor** y se representa en formato BSON.

Un documento puede albergar diversos tipos de información, tales como secuencias de texto, números, listas, fechas, objetos anidados e incluso otros documentos similares. Esta adaptabilidad posibilita que los documentos en la misma colección posean distintas estructuras, lo que convierte a MongoDB en una base de datos con dinámica esquemática.

```
pipeline = [
        { $match : { … } },
        { $group : { … } },
        { $sort : { … } }
       ]
```

Explica la diferencia entre JSON y BSON en MongoDB.

MongoDB emplea un modelo de datos centrado en documentos, utilizando dos formatos clave para representar y manejar los datos: y formatos BSON. A pesar de que están vinculados, hay diferencias significativas en su estructura, objetivo y funcionalidad.

JSON (JavaScript Object Notation)

Es un formato ligero de intercambio de datos, ampliamente utilizado por su simplicidad y legibilidad. Se basa en una estructura de pares clave-valor, similar a los objetos en JavaScript.

Características principales:

- **Legibilidad:** Es fácilmente entendible por humanos.
- **Portabilidad:** Es compatible con la mayoría de los lenguajes de programación.
- **Uso común:** Se utiliza principalmente para enviar y recibir datos entre aplicaciones, especialmente en servicios web y APIs.

BSON (Binary JSON)

Es una representación binaria de JSON. Fue diseñado específicamente para que MongoDB pueda almacenar y procesar documentos de manera más eficiente.

Características principales:

- **Eficiencia:** Al ser un formato binario, permite una lectura y escritura más rápida por parte del motor de base de datos.
- **Ampliación de tipos de datos:** Soporta tipos de datos adicionales que no existen en JSON, como ObjectId, Date, Binary, Decimal128, entre otros.
- **Uso interno:** MongoDB utiliza BSON internamente para el almacenamiento y transmisión de documentos

Estructura de los archivos JSON

JSON admite dos tipos principales de estructuras:

- Matriz (Array)

Representa una lista ordenada de valores, separados por comas y delimitados por corchetes [ ]. Los elementos pueden ser de cualquier tipo válido en JSON: números, cadenas, booleanos, objetos, otras matrices, etc.

- Ejemplo válido:

[1, "pepe", 3.14, "Pepito Conejo"]

- Objeto (Object)

Consiste en una colección de pares clave-valor, delimitados por llaves { }.

Las claves deben ir entre comillas dobles " y deben estar seguidas por dos puntos :.
Los valores pueden ser de cualquier tipo permitido.

- Ejemplo válido:

{"nombre": "Pepito Conejo",

"edad": 25,

"carnet de conducir": true}

- Reglas sintácticas fundamentales

Un archivo JSON debe contener un único elemento raíz, que puede ser un objeto o una matriz, pero no ambos al mismo nivel.

- Incorrecto:

[1, 2, 3], ["a", "b", "c"]

"nombre": "Pepito Conejo"

No se permite una coma final después del último elemento de una matriz u objeto.

- Incorrecto:

{ "nombre": "Pepito Conejo", }

["nombre", "edad",]

Los espacios en blanco y saltos de línea no son significativos, por lo que se pueden utilizar libremente para mejorar la legibilidad del documento.

- Ejemplo válido con formato estructurado:

[{"nombre": "Pepito Conejo",

"edad": 25,

"carnet de conducir": true},

{"nombre": "Ana Barberá",

"edad": 90,

"carnet de conducir": false}]

¿Qué ventajas tiene MongoDB sobre una base de datos relacional en términos de escalabilidad y flexibilidad?

- **Escalabilidad horizontal:** permite distribuir la carga entre varios servidores (sharding).
- **Esquema flexible:** no es necesario definir todas las columnas al inicio.
- Alto rendimiento con grandes volúmenes de datos no estructurados.
- Ideal para aplicaciones en tiempo real y Big Data.

Comandos para realizar CRUD en MongoDB

- Crear

```
db.usuarios.insertOne({nombre: "Ana", edad: 30})
```

- Leer

```
db.usuarios.find({edad: {$gt: 25}})
```

- Actualizar

```
db.usuarios.updateOne({nombre: "Ana"}, {$set: {edad: 31}})
```

- Eliminar

```
db.usuarios.deleteOne({nombre: "Ana"})
```

¿Cómo se pueden relacionar datos en Mongo sin usar joins como en SQL?

1. Documentos embebidos

Consiste en guardar los datos relacionados dentro del mismo documento. Es útil cuando los datos están muy relacionados y se usan juntos frecuentemente.

Ejemplo:

{ "nombre": "Laura",

"edad": 30,

"direccion": {

"calle": "Calle 10",

"ciudad": "Bogotá" } }

Aquí, la dirección está dentro del mismo documento de la persona.

2. Referencias

En este caso, un documento guarda el _id de otro documento relacionado, como una "referencia". Luego, se pueden usar varias consultas para obtener los datos relacionados.

Ejemplo:

{"_id": 1,

"nombre": "Pedro"}

{"_id": 101,

"producto": "Celular",

"persona_id": 1}

El campo `"persona_id"` guarda el ID de la persona que hizo el pedido.

Descargar imagen de Mongo en Docker

Herramientas similares a Workbench para visualizar los datos de Mongo

- MongoDB Compass
- DBeaver
- Robo 3T
- NoSQLBooster
- Studio 3T

Conclusiones

Este documento ha tratado de forma completa el funcionamiento y utilización de MongoDB, una base de datos NoSQL enfocada en documentos. Se han reconocido las diferencias clave entre JSON y BSON, además de su estructura y sintaxis, componentes cruciales para entender cómo se guardan y se ilustran los datos en MongoDB. Además, se investigaron los comandos CRUD que facilitan la ejecución de operaciones fundamentales como la creación, lectura, actualización y supresión de datos en las colecciones.

5. Bibliografía

- MongoDB Official Documentation: https://www.mongodb.com/docs
- Docker Hub Mongo Image: https://hub.docker.com/_/mongo
- Compass: https://www.mongodb.com/products/compass
- NoSQLBooster: https://nosqlbooster.com/
- Curso MongoDB – Platzi (2023)
- Stack Overflow – MongoDB tag
- Recluit. (s.f.). *¿Cuál es la diferencia entre SQL, MySQL y NoSQL?* Recluit. https://recluit.com/cual-es-la-diferencia-sql-mysql-y-nosql/
- IONOS. (s.f.). *MongoDB vs. SQL: diferencias y usos*. IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/servidores/know-how/mongodb-vs-sql/
- Pure Storage. (s.f.). *¿Qué es MongoDB?* Pure Storage. https://www.purestorage.com/es/knowledge/what-is-mongodb.html
- Studio 3T. (s.f.). *MongoDB Aggregation Framework*. Studio 3T. https://studio3t.com/es/knowledge-base/articles/mongodb-aggregation-framework/