

Taller Mongo

José Santiago Lemus Durán

ID:832819

William Alexander Matallana Porras

Bases de Datos Masivas

21/03/2025

Introducción.

MongoDB es una base de datos NoSQL orientada a documentos que almacena datos en BSON (Binary JSON), lo que le otorga flexibilidad y escalabilidad frente a bases de datos relacionales. A diferencia de SQL, no usa JOINS, sino que relaciona datos mediante documentos embebidos, referencias (ObjectId) o agregaciones con \$lookup. Sus principales ventajas incluyen esquema dinámico, escalado horizontal (Sharding) y alta disponibilidad (Replica Sets). Las operaciones CRUD se realizan con comandos como insertOne(), find(), updateOne() y deleteOne(), permitiendo una manipulación eficiente de datos sin estructuras rígidas.

1. ¿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 de código abierto, siendo una base de datos no relacional, esta puede procesar datos estructurados, semiestructurados y no estructurados, utiliza un modelo de datos no relacional orientado a documentos y un lenguaje de consulta no estructurado.

Resumen diferencias entre mongoDB y MySQL son:

- Modelo de datos:

MongoDB: Almacena los datos en documentos JSON y, a continuación, los organiza en colecciones.

MySQL: Almacena los datos en columnas y filas. El almacenamiento de datos es tabular y relacional.

- Escalabilidad:

MongoDB: Usa la replicación y la partición para escalar horizontalmente.

MySQL: Usa el escalado vertical y lee réplicas para mejorar el rendimiento a escala

- Lenguaje de consulta:

MongoDB: Usa el lenguaje de consulta MongoDB.

MySQL: Usa SQL

- Rendimiento:

MongoDB: Destaca en la inserción o actualización de una gran cantidad de registros.

MySQL: Es más rápido cuando se selecciona una gran cantidad de registros.

- Flexibilidad:

MongoDB: No tiene un esquema, lo que proporciona más flexibilidad y le permite trabajar con datos estructurados, semiestructurados y no estructurados.

MySQL: Tiene un esquema rígido que funciona bien con datos estructurados.

- Seguridad:

MongoDB: Usa los certificados Kerberos, X.509 y LDAP para autenticar a los usuarios.

MySQL: Usa métodos de autenticación integrados.

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

Las colecciones son agrupaciones de documentos, son equivalentes a las tablas en bases de datos relacionales, pero no imponen un esquema o estructura rígida para guardar la información, los documentos son registros dentro de las colecciones.

Diferencias entre MongoDB y MySQL:

- Estructura:

MongoDB: Almacena datos en documentos BSON.

SQL: Almacena datos en tablas relacionales.

- Esquema:

MongoDB: No tiene un esquema rígido.

SQL: Tiene un esquema rígido.

- Flexibilidad:

MongoDB: Es más flexible.

SQL: Funciona bien para sistemas complicados.

- Lenguaje de consulta:

MongoDB: Usa el lenguaje de consulta MongoDB

SQL: Usa SQL.

MongoDB es ideal para proyectos que necesitan ser flexibles, capaces de crecer fácilmente y construirse rápidamente, SQL funciona bien para sistemas complicados que implican muchas conexiones entre datos.

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

MongoDB usa un formato de almacenamiento de documentos llamado BSON, que es una forma binaria de JSON(JavaScript Object Notation) que puede acomodar mas tipos de datos.

4. Explica las diferencias entre JSON y BSON en MongoDB

JSON y BSON son formatos de almacenamiento y transferencia de datos utilizados n MongoDB, sus diferencias clave son:

- Formato y almacenamiento:

JSON: Es un formato de texto legible por humanos y ampliamente utilizado en la web para intercambiar datos.

BSON: Es un formato binario que extiende JSON, optimizado para el rendimiento y almacenamiento en MongoDB.

- Estructura y tipos de datos:

JSON: Soporta tipos de datos básicos como cadenas, números, booleanos, listas y objetos.

BSON: Agrega tipos adicionales como fechas, binarios, ObjectId, y números enteros de 32 y 64 bits.

- Eficiencia y velocidad:

JSON: Al ser texto, puede ser más grande y menos eficiente en almacenamiento y procesamiento.

BSON: Al estar en binario, es más compacto y rápido para leer y escribir en la base de datos.

5. Estructura de los archivos JSON.

La estructura de los datos JSON se basa en dos elementos: una colección de pares clave / valor y una lista ordenada de valores. Esto es similar a los tipos de datos utilizados en muchos lenguajes de programación, como los objetos y los arrays.

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

MongoDB tiene varias ventajas sobre las bases de datos relacionales (SQL) en términos de escalabilidad y flexibilidad, lo que lo hace ideal para aplicaciones modernas con grandes volúmenes de datos.

- Escalabilidad

Escalado horizontal (Sharding): Permite distribuir datos en múltiples servidores mediante sharding, lo que facilita el manejo de grandes cantidades de información y tráfico.

Las bases de datos relacionales suelen escalar verticalmente (aumentando el hardware del servidor), lo que puede ser costoso y tener un límite físico.

Alta disponibilidad y replicación: Utiliza Replica Sets, que replican los datos en múltiples servidores para garantizar la disponibilidad en caso de fallos.

En SQL, la replicación suele ser más compleja y requiere configuraciones avanzadas.

- Flexibilidad en la estructura de datos

Modelo de datos NoSQL basado en documentos: En MongoDB, los datos se almacenan en documentos BSON (similares a JSON), permitiendo esquemas flexibles.

En una base de datos relacional, el esquema es rígido y cualquier cambio en la estructura requiere migraciones complejas.

Soporte para datos no estructurados o semiestructurados: Ideal para datos variados, como registros de actividad, contenido multimedia o IoT.

SQL requiere definir tablas y relaciones estrictas antes de almacenar datos.

7. Comandos para realizar CRUD en MongoDB.

Crear (Create): Inserta documentos en una colección:

- **Inserta un solo documento:**

```
db.usuarios.insertOne({  
  nombre: "Juan",  
  edad: 30,  
  email: "juan@example.com"  
})
```

- **Inserta múltiples documentos:**

```
db.usuarios.insertMany([  
  { nombre: "Ana", edad: 25, email: "ana@example.com" },
```

```
{ nombre: "Carlos", edad: 35, email: "carlos@example.com" }  
])
```

Leer (Read): Consulta documentos en una colección:

- **Obtener todos los documentos:**

```
db.usuarios.find()
```

- **Buscar un solo documento:**

```
db.usuarios.findOne({ nombre: "Juan" })
```

Actualizar (update): Modifica documentos existentes:

- **Actualizar un solo documento:**

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

- **Actualizar múltiples documentos:**

```
db.usuarios.updateMany(  
  { edad: { $lt: 30 } },  
  { $set: { estado: "activo" } }  
)
```

8. ¿Cómo se pueden relacionar datos en MongoDB sin usar joins como en SQL?

- **Documentos Embebidos (Denormalización)**

Se almacenan datos relacionados dentro del mismo documento.

Rápido acceso a los datos.

Puede generar redundancia y documentos grandes.

Ejemplo: Un usuario con sus direcciones dentro del mismo documento.

- **Referencias entre Documentos (Normalización)**

Se usa un ObjectId para referenciar otro documento en otra colección.

Evita duplicación de datos y facilita actualizaciones.

Requiere consultas adicionales para obtener los datos relacionados.

Ejemplo: Un usuario guarda una lista de ObjectId que apuntan a direcciones.

- **Agregaciones con \$lookup (Simulación de JOINS)**

Se combinan datos de múltiples colecciones en una consulta con \$lookup.

Permite obtener datos relacionados sin redundancia.

Puede ser menos eficiente en grandes volúmenes de datos.

Ejemplo: Unir usuarios con direcciones en una consulta agregada.

- **Elección ideal:**

Datos consultados juntos → Documentos embebidos.

Datos con cambios frecuentes → Referencias entre documentos.

Consulta puntual con datos relacionados → \$lookup.

Bibliografías.

<https://www.purestorage.com/es/knowledge/what-is-mongodb.html#:~:text=MongoDB%20es%20una%20base%20de,lenguaje%20de%20consulta%20no%20estructurado.>

<https://aws.amazon.com/es/compare/the-difference-between-mongodb-vs-mysql/#:~:text=MongoDB%20no%20tiene%20un%20esquema,funciona%20bien%20con%20datos%20estructurados.&text=MongoDB%20usa%20los%20certificados%20Kerberos,para%20autenticar%20a%20los%20usuarios.>

<https://www.purestorage.com/es/knowledge/what-is-mongodb.html#:~:text=MongoDB%20usa%20un%20formato%20de,acomodar%20m%C3%A1s%20tipos%20de%20datos.>

<https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve>