Creación de los objetos de la base de datos

Cristian Camilo Carvajal Montes

Servicio Nacional de Aprendizaje SENA

Complejo Tecnológico, Turístico Y Agroindustrial Del Occidente Antioqueño

Manizales, Colombia

13 de agosto de 2023

Introducción

En el contexto de esta actividad de aprendizaje, exploraremos el uso de MongoDB para la creación de una base de datos NoSQL. Utilizaremos los datos proporcionados en esta actividad para demostrar cómo se generan instrucciones, se insertan datos y se realizan consultas. Para llevar a cabo este proceso, aprovecharemos las herramientas MongoDB Compass y la terminal MongoDB Shell, que serán nuestras aliadas en el desarrollo de esta actividad.

Objetivos

El propósito fundamental de esta actividad es adquirir un conocimiento sólido sobre los diversos comandos necesarios para ejecutar tareas clave, como la creación de colecciones, la inserción de datos y la realización de consultas en MongoDB. Al finalizar, habremos alcanzado los siguientes objetivos:

1. **Comprender los Fundamentos de MongoDB:** A través de la manipulación directa de MongoDB Compass y MongoDB Shell, desarrollaremos un entendimiento profundo de cómo se estructuran y gestionan las bases de datos NoSQL.

2. **Dominar la Creación de Colecciones:** Aprenderemos a utilizar los comandos adecuados para la creación de colecciones, lo que nos permitirá organizar y clasificar nuestros datos de manera eficiente.

3. **Perfeccionar la Inserción de Datos:** Exploraremos el proceso de inserción de datos en las colecciones, asegurándonos de comprender cómo almacenar la información de manera precisa y coherente.

4. **Desarrollar Habilidades de Consulta Avanzada:** Profundizaremos en la realización de consultas, explorando cómo recuperar información específica de manera efectiva y aplicando filtros que se adapten a nuestras necesidades.

5. **Familiarizarse con la Interfaz de Usuario y la Terminal:** Ganaremos confianza en el uso de MongoDB Compass y MongoDB Shell como herramientas esenciales para interactuar con nuestras bases de datos.

Al finalizar esta actividad, se habrá logrado una sólida base en la manipulación de bases de datos NoSQL con MongoDB, lo que les brindará habilidades valiosas para su desarrollo profesional en el ámbito de la gestión de datos y la tecnología.

Sentencias de Creación de Colecciones

Para la creación de una sentencia de debe de acceder a la base en la cual se desea crear, para esto podemos el comando “use” y especificar la base de datos que se vaya a usar, en mi caso se usará una que está almacenada localmente. Una vez estemos en la base que se desea trabajar, podemos proceder a crear las colecciones que se necesiten usando el comando “db.createCollection(“nombre de la colección”), este se usará para crear cada una de las colecciones que se necesiten.

Inserciones en Bases de Datos No Relacionales

En esta ocasión, se pueden insertar datos con el comando “Insert” que a su vez permite ingresa sólo uno o varios a la vez “insertOne” o “insertMany”, dependiendo de lo que necesitemos hacer, se usará uno o el otro, para este caso se usa el comando “insertMany” para crear todos los documentos necesarios de una vez. Cabe aclara que se debe especificar la colección en la que se van a agregar dichos datos, esto se haría de la siguiente manera:

```
db.carros.insertMany([{
  "_id": {
    "$oid": "64e2ddd9a54a21ffbbf4007b"
  },
  "Placa": "",
  "Número de Serie": 0,
  "Modelo": 0,
  "Marca": "",
  "Kilometraje": 0,
  "Tipo": ""
},
{
  "_id": {
    "$oid": "64e2e56ca54a21ffbbf4007c"
  },
  "Placa": "ASD457",
  "Número de Serie": 123456789,
  "Modelo": 2013,
  "Marca": "Chevrolet",
  "Kilometraje": 34609,
  "Tipo": "Sedan"
},
{
  "_id": {
    "$oid": "64e2e56ca54a21ffbbf4007d"
  },
  "Placa": "HJK534",
  "Número de Serie": 234567890,
  "Modelo": 2010,
  "Marca": "Mazda",
  "Kilometraje": 236921,
  "Tipo": "Camioneta"
},
{
  "_id": {
    "$oid": "64e2e56ca54a21ffbbf4007e"
  },
  "Placa": "ARF092",
  "Número de Serie": 345678901,
  "Modelo": 2019,
  "Marca": "Renault",
  "Kilometraje": 18898,
  "Tipo": "Sedan"
},
{
```

```
  "_id": {
    "$oid": "64e2e56ca54a21ffbbf4007f"
  },
  "Placa": "CBN123",
  "Número de Serie": 456789012,
  "Modelo": 2017,
  "Marca": "Nissan",
  "Kilometraje": 98034,
  "Tipo": "Camioneta"
},
{
  "_id": {
    "$oid": "64e2e56ca54a21ffbbf40080"
  },
  "Placa": "HDJ786",
  "Número de Serie": 567890123,
  "Modelo": 2016,
  "Marca": "Volkswagen",
  "Kilometraje": 73837,
  "Tipo": "Coupe"
}])
```


Se puede notar como cada documento está dentro de llaves y para saltar al siguiente, se usa el signo de la coma para indicar que se generará uno nuevo, con la excepción de que el último documento que se va a crear no finalizaría con este signo, sino que simplemente se cierran sus llaves y correspondientes corchetes y paréntesis, con esto se le indica al comando de que sea finalizado a la inserción de datos a la base de datos.

Consultas

Para ver los datos que se han ingresado a dicha colección se puede usar el comando “find()”, con este se podrán ver todos los datos que se encuentran allí.

db.carros.find()

Actualizar

Para la actualización de los documentos de una colección, MongoDB ofrece varios operadores según el tipo de modificación que se vaya a hacer en él.

1. $set: Este operador permite modificar el valor de un campo existente en un documento o agregar un campo nuevo si no existe. Por ejemplo:

```
db.collection.updateOne({ _id: ObjectId("id_del_documento") }, { $set: { nombre: "Nuevo Nombre", edad: 30 } })
```

2. $unset: Elimina un campo específico de un documento. Por ejemplo:

```
db.collection.updateOne({ _id: ObjectId("id_del_documento") }, { $unset: { campo_a_eliminar: 1 } })
```

3. $inc: Este operador se utiliza para incrementar (o decrementar si se proporciona un valor negativo) el valor de un campo numérico. Por ejemplo:

```
db.collection.updateOne({ _id: ObjectId("id_del_documento") }, { $inc: { cantidad: 5 } })
```

4. $push: Agrega un valor a un campo que es un array. Por ejemplo:

```
db.collection.updateOne({ _id: ObjectId("id_del_documento") }, { $push: { elementos: "nuevo_elemento" } })
```

5. $pull: Elimina elementos de un array en función de ciertos criterios. Por ejemplo:

```
db.collection.updateOne({ _id: ObjectId("id_del_documento") }, { $pull: { elementos: "elemento_a_eliminar" } })
```

6. $addToSet: Agrega un elemento a un array solo si el elemento no existe en el array. Por ejemplo:

```
db.collection.updateOne({ _id: ObjectId("id_del_documento") }, { $addToSet: { elementos: "nuevo_elemento" } })
```

Estos son solo algunos ejemplos de los operadores de actualización que MongoDB ofrece. Los operadores de actualización permiten modificar documentos de manera precisa y eficiente, lo que es esencial en el desarrollo y mantenimiento de tus bases de datos.

Para el caso de actualizar uno de los documentos en donde se desee cambiar el nombre de la marca Volkswagen por Cupra, se usaría el operador $set para modificarlo, quedando de la siguiente forma,

```
db.carros.updateOne({Marca:"Volkswagen"},{$set:{Marca:"Cupra"}})
```

Conclusión

Con práctica constante y siguiendo buenas pautas, podemos aprender rápidamente cómo gestionar bases de datos usando las operaciones CRUD. Además, esta práctica nos ayuda a entender cómo funcionan estas bases de datos.

Al practicar una y otra vez, nos volvemos cómodos con la creación, lectura, actualización y eliminación de datos. Esto nos da confianza y nos permite tomar decisiones informadas. Al aplicar estos conceptos a situaciones reales, desarrollamos una comprensión sólida de cómo fluyen los datos y cómo resolver problemas.

A través de este enfoque, podemos mejorar nuestras habilidades en la gestión de bases de datos de manera rápida y efectiva. Esta habilidad no solo es valiosa, sino que también nos empodera para abordar desafíos en el mundo de la tecnología y los datos con confianza."