01 Introducción

Concepto de documento en MongoDB

Tres ventajas:

- Correspondencia con tipos nativos de lenguajes (objetos).
- Los arrays o documentos embebidos reducen la necesidad de joins.
- Esquemas dinámicos que soportan polimorfismo.

JSON en MongoDB

Los tipos nativos soportados por JSON son:

Objects { ... }
Arrays [...]
Primitives:
strings,
numbers,
boolean values true/false,
null.

BSON en MongoDB

BSON es un formato de serialización para almacenar documentos y permitir procedimientos de llamadas remotas en MongoDB. Es un acrónimo de las palabras binario y JSON y se puede pensar en éll como una representación binaria de JSON.

Límites

El máximo tamaño de un documento BSON es de 16 megabytes
MongoDB no soporta más de 100 niveles de anidado de documentos BSON.

Documentos, colecciones y bases de datos en MongoDB

MongoDB almacena los documentos en colecciones, siendo estas análogas a las tablas de las bases de datos relacionales.

Además de las colecciones, MongoDB soporta:

- Vistas de solo lectura (desde la versión MongoDB 3.4)
- Vistas On-Demand Materialized (desde la versión in MongoDB 4.2).

Las colecciones están contenidas en una base de datos y cada combinación base de datos/colección define una namespace con la notación del punto.

Campo _id

Todos los documentos contienen el campo _id.

• Si el campo _id no es especificado cuando se inserta un documento, MongoDB lo añadirá como un tipo ObjectId.
• La mayoría de los driver crean también el ObjectId si el campo _id no es especificado.
• Tiene las siguientes restricciones:

El _id es inmutable.
No puede ser un array pero si un documento.
El _id será único para una colección, actuando como clave primaria para replicación.

Tipos de datos en MongoDB

MongoDB soporta una amplia variedad de tipos BSON. Cada tipo tiene asociado un número y un alias de tipo string que pueden ser usados en consultas que seleccionen documentos por su tipo BSON.

Double 1 “double”
String 2 “string”
Object 3 “object”
Array 4 “array”
Binary data 5 “binData”
ObjectId 7 “objectId”
Boolean 8 “bool”
Date 9 “date”
Null 10 “null”
Regular Expression 11 “regex”
JavaScript 13 “javascript”
JavaScript (w/ scope) 15 “javascriptWithScope”
32-bit integer 16 “int”
Timestamp 17 “timestamp”
64-bit integer 18 “long”
Decimal128 19 “decimal”
Min key -1 “minKey”
Max key 127 “maxKey”

https://docs.mongodb.com/manual/reference/bson-types/

Shell de MongoDB

https://docs.mongodb.com/manual/mongo/

La shell de mongo es una interfaz interactiva de MongoDB que puede ser usada tanto para consultar y actualizar datos como para realizar operaciones de administración.

Inicio convencional

mongo --host mongodb0.example.com --port 28015

Inicio con usuario, contraseña y definición de base de datos

mongo --username alice --password --authenticationDatabase admin --host mongodb0.examples.com --port 28015

Además dispone de otros inicios con SSL, réplicas, etc.

Working with the mongo Shell

show dbs

db

use db // crea si no existe

show collections

Las operaciones tienen la sintaxis:

db.collection.método1(documentos).método2(documentos)...

Alternativa para identificadores de colección no permitidos:

db.getCollection("colección").método1(documentos)....

Operaciones básicas

Crear base de datos

use clinica

Crear colección y su primer registro:

db.pacientes.insert({nombre: ‘Juan’, apellidos: ‘Pérez Fernández’, edad: 28})

Crear colección sin primer registro:

db.createCollection(‘empleados’)

Devolver la base de datos actual:

db // alternativamente db.getName()

Eliminar base de datos:

db.dropDatabase()

Eliminar colección:

db.pacientes.drop()

Ayudas de la Shell

Las habituales y multilínea.

Uso de JavaScript

La shell de mongo interpreta JavaScript, por ejemplo:

```
> let nombres = ['Laura','Juan','Fernando','María']
> let apellidos = ['Fernández','González','Pérez','López']
> let pacientes = []
> for (i=0; i<100; i++) {
... pacientes.push({
... nombre: nombres[Math.floor(Math.random()*nombres.length)],
... apellido1: apellidos[Math.floor(Math.random()*apellidos.length)],
... apellido2: apellidos[Math.floor(Math.random()*apellidos.length)],
... })
... }
> db.pacientes.insert(pacientes)
```

Formato de salida por pantalla

db.pacientes.find() // Convencional a un cursor con 20 documentos iterable con it

let variable = db.pacientes.find() // sin cursor

db.pacientes.find().pretty() // formateo a objeto ‘pretty’

printjson(variable) // formateo a JSON

Comprobación de tipos

typeof
instanceof

```
> let prPacient = db.clinica.findOne()
> prPacient
{
        "_id" : ObjectId("5e257c5d44610cd37a7f8f23"),
        "nombre" : "María",
        "apellido" : "López"
}
> typeof prPacient.nombre
string
> prPacient._id instanceof ObjectId
true
```

Ayudas de la shell

Ayuda de conexión

mongo --help

Ayuda de comandos disponibles

help

Ayuda de métodos de la base de datos

db.help()

Ayuda de métodos de la colección

db.coleccion.help()

Ayuda de métodos (realmente devuelve el código)

db.método // sin paréntesis

Cierre de la shell de Mongo

quit() o CTRL + C

Referencia de la Shell

https://docs.mongodb.com/manual/reference/mongo-shell/#keyboard-shortcuts