

UNIR - Universidad Internacional de la Rioja

Ciudad de México

ACTIVIDAD 2:

Uso avanzado de bases de datos NoSQL

Alumnos:

Paola Michelle Figueroa Benitez

Lowenski Paredes Rosario

Carlos Damian Rodriguez Uitzil

Cuenca Roa Leonard Jose

Grupo: 1001

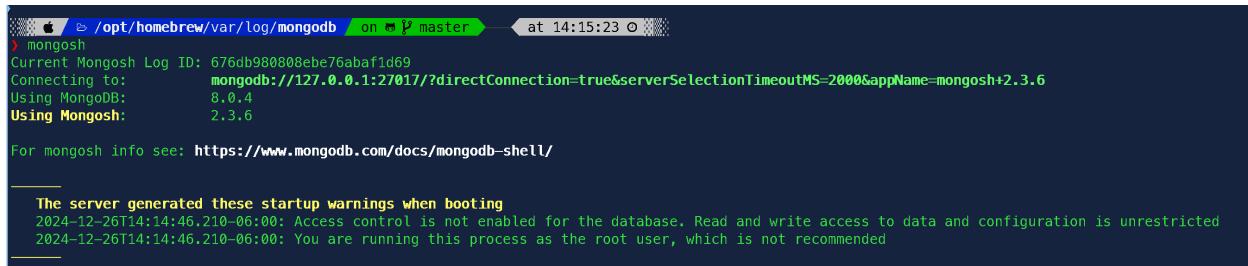
Equipo 01C

ÍNDICE

Prepara el entorno de trabajo	3
Inicia la monitorización en MongoDB	3
Importar la base de datos	4
Exportar la base de datos	12
Restaurar una base de datos	13
Caso de uso	15

Prepara el entorno de trabajo

Se implementó un entorno de desarrollo local consistente en MongoDB versión 8.0.4 y Mongosh 2.3.6 en cada equipo de los participantes. Esta configuración permitió a cada miembro del equipo trabajar de manera independiente y sincronizada, facilitando la colaboración y el testing.



```
/opt/homebrew/var/log/mongodb on master at 14:15:23
> mongosh
Current Mongosh Log ID: 676db980808ebe76abaf1d69
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.6
Using MongoDB: 8.0.4
Using Mongosh: 2.3.6

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

The server generated these startup warnings when booting
2024-12-26T14:14:46.210-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-12-26T14:14:46.210-06:00: You are running this process as the root user, which is not recommended
```

Inicia la monitorización en MongoDB

Para llevar a cabo esta actividad, se implementó una estrategia de ejecución simultánea en tres terminales. En la primera, se utilizó el demonio de MongoDB (mongod) para crear la base de datos 'inspections' y 'countries' y las colecciones 'act-grupal-city_inspections', 'act-grupal-countries-big' y 'act-grupal-countries-small'. Esta configuración inicial se realizó directamente desde la línea de comandos de MongoDB.

Con el objetivo de monitorear el rendimiento y la salud de la base de datos durante el proceso, se abrieron dos terminales adicionales. En estas, se ejecutaron las herramientas de monitoreo mongotop y mongostat. El comando 'mongotop 10' proporcionó una vista en tiempo real de las operaciones más costosas en la base de datos, mientras que 'mongostat --discover' ofreció un resumen general del estado del servidor y las estadísticas de las bases de datos.

Creación de Base Datos y sus colecciones

```
// Crear la base de datos
use inspections;
// crear colecciones
db.createCollection('act-grupal-city_inspections')
// visualizar lo que se generó
show collections
//Otra forma de ver
db.getCollectionNames()
```

```
inspections> show collections
act-grupal-city_inspections
inspections> |
```

Creación de Base Datos y sus colecciones

```
// Crear la base de datos
use countries;
// crear colecciones
db.createCollection('act-grupal-countries-big')
db.createCollection('act-grupal-countries-small')
// visualizar lo que se generó
show collections
//Otra forma de ver
db.getCollectionNames()
```

```
inspections> use countries
switched to db countries
countries> db.createCollection('act-grupal-countries-big')
{ ok: 1 }
countries> db.createCollection('act-grupal-countries-small')
{ ok: 1 }
countries> show collections
act-grupal-countries-big
act-grupal-countries-small
countries> |
```

Importar la base de datos

Ahora para monitoreo de la carga de los datos la actividad número dos, nos proporciono los ficheros que contiene los datos json para las colecciones creadas se procedió a cargar, validar y analizar los resultados que arroja las terminales de monitoreo, Para esta tarea de

importación de datos, se aprovechó la rapidez y facilidad que ofrece Compass, una excelente herramienta de administración de bases de datos MongoDB. Previamente, se configuró Compass para utilizar la base de datos instalada localmente

Importar datos

Se importó los datos en la BD llamada **inspections**,

validando que en efecto esté el total de datos 81047.

The screenshot shows the MongoDB Compass application interface. On the left, there's a sidebar titled "My Queries" and "CONNECTIONS (2)". Under "CONNECTIONS", there are two entries: "localhost/mongo4" and "LOCAL". The "LOCAL" connection is expanded, showing collections: "addresses", "config", "countries", "experiences", "inspections", "local", and "privates". The "inspections" collection is selected and shown in the main pane. The main pane has tabs for "Documents", "Aggregation", "Schema", "Indexes", and "Validation". The "Documents" tab is active, displaying 81047 documents. A search bar at the top of this tab allows querying by field value or generating a query. Below the search bar are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". The list of documents shows fields like "_id", "date", "business_name", "sector", and "address". The "address" field is highlighted in blue. At the bottom of the document list, there's a note about the "deprecated" use of "estimatedDocumentCount". The bottom right corner of the main pane shows a "MongoDB Shell" icon. To the right of the main pane, there's another window titled "MongoDB Compass - LOCAL\$shell" which contains a mongo shell command line.

Importar datos

Se importó los datos en la BD llamada **countries**, validando que en efecto esté el total de datos 21640 para la colección **act-grupal-countries-big**.

MongoDB Compass - LOCAL\countries.act-grupal-countries-big

Documents 21640 Aggregations Schema Indexes Validation

Import completed. 1 document imported.

MongoDB Compass - LOCAL\countries.act-grupal-countries-small

Documents 21640 Aggregations Schema Indexes Validation

Import completed. 1 document imported.

Importar datos

Se importó los datos en la BD llamada **countries**, validando que en efecto esté el total de 1 para la colección **act-grupal-countries-small**.

MongoDB Compass - LOCAL\countries.act-grupal-countries-small

Documents 1 Aggregations Schema Indexes Validation

Import completed. 1 document imported.

MongoDB Compass - LOCAL\countries

Documents 1 Aggregations Schema Indexes Validation

Import completed. 1 document imported.

Análisis del comando mongotop al importar “act-grupal-city_inspections.json”

Al momento de importar los datos, se nos presenta un proceso dividido en tres fases que reflejan el tiempo total empleado, el cual es la suma del tiempo invertido en las operaciones de lectura y escritura realizadas por el procesador en ese instante preciso.

Primera fase: En esta etapa, se utilizó la operación de lectura durante 37 milisegundos.

Segunda fase: Continuando con el proceso, se mantuvo el uso de la operación de lectura, esta vez durante 68 milisegundos.

Tercera fase: En esta fase, se observa un cambio en la operación, pasando a utilizar la operación de escritura, la cual consumió 560 milisegundos.

Análisis:

A partir de los datos obtenidos, podemos concluir que las operaciones de escritura son más complejas que las de lectura, lo que conlleva un mayor tiempo de procesamiento por parte del procesador. Asimismo, se puede deducir que, durante el proceso de importación, la herramienta realiza una lectura completa del documento JSON antes de proceder a escribir los datos en la base de datos denominada 'inspections'

mongotop 10				
	ns	total	read	write
	countries.act-grupal-countries-big	37ms	37ms	0ms
	countries.act-grupal-countries-small	37ms	37ms	0ms
	admin.\$cmd.aggregate	0ms	0ms	0ms
	admin.atlascli	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	config.collections	0ms	0ms	0ms
	config.system.sessions	0ms	0ms	0ms
	config.transactions	0ms	0ms	0ms
inspections.act-grupal-city_inspections		0ms	0ms	0ms
	local.system.replset	0ms	0ms	0ms
	ns	total	read	write
	inspections.act-grupal-city_inspections	68ms	68ms	0ms
	config.collections	1ms	1ms	0ms
	admin.\$cmd.aggregate	0ms	0ms	0ms
	admin.atlascli	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	config.system.sessions	0ms	0ms	0ms
	config.transactions	0ms	0ms	0ms
	countries.act-grupal-countries-big	0ms	0ms	0ms
	countries.act-grupal-countries-small	0ms	0ms	0ms
	local.system.replset	0ms	0ms	0ms
	ns	total	read	write
	inspections.act-grupal-city_inspections	560ms	0ms	560ms
	admin.\$cmd.aggregate	0ms	0ms	0ms
	admin.atlascli	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	config.collections	0ms	0ms	0ms
	config.system.sessions	0ms	0ms	0ms
	config.transactions	0ms	0ms	0ms
	countries.act-grupal-countries-big	0ms	0ms	0ms
	countries.act-grupal-countries-small	0ms	0ms	0ms
	local.system.replset	0ms	0ms	0ms
	ns	total	read	write
	inspections.act-grupal-city_inspections	11ms	4ms	7ms
	admin.\$cmd.aggregate	0ms	0ms	0ms
	admin.atlascli	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	config.collections	0ms	0ms	0ms
	config.system.sessions	0ms	0ms	0ms

Análisis del comando mongostat --discover al importar “act-grupal-city_inspections.json”

Este tipo de herramienta nos proporciona en tiempo real la cantidad de operaciones que se realizan por segundo, para el caso de importar la colección nos arroja que se realizó un conjunto de inserciones sobre la colección mencionada, durante la instancia de Mongo DB en local nos arroja que se realizaron mas de 21.000 mil operaciones durante un segundo terminando el proceso de inserción en 4 bloques de procesos registrados usando un promedio de memoria de 0.725%.

mongostat --discover																		
localhost	*0	*0	*0	*0	0	3 0	0.0%	0.0%	0	393G	45.0M	0 0	0 0	2,15k	87.9k	29 Dec 26 16:16:02.576		
localhost	*0	*0	*0	*0	0	3 0	0.0%	0.0%	0	393G	45.0M	0 0	0 0	312b	83.7k	29 Dec 26 16:16:03.575		
localhost	*0	*0	*0	*0	0	0 0	0.0%	0.0%	0	393G	45.0M	0 0	0 0	111b	82.8k	29 Dec 26 16:16:04.578		
localhost	*0	*0	*0	*0	0	2 0	0.0%	0.0%	0	393G	43.0M	0 0	0 0	246b	83.5k	29 Dec 26 16:16:05.576		
localhost	*0	*0	*0	*0	0	0 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	111b	82.8k	29 Dec 26 16:16:06.578		
localhost	*0	*0	*0	*0	0	1 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	112b	83.3k	29 Dec 26 16:16:07.575		
localhost	*0	*0	*0	*0	0	1 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	173b	83.2k	29 Dec 26 16:16:08.577		
host insert query update delete getmore command dirty used flushes vszie res qrw arw net_in net_out conn time																		
localhost	*0	*0	*0	*0	0	1 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	245b	83.1k	29 Dec 26 16:16:09.579		
localhost	*0	*0	*0	*0	0	0 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	111b	83.0k	29 Dec 26 16:16:10.579		
localhost	*0	*0	*0	*0	0	6 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	560b	84.7k	29 Dec 26 16:16:11.578		
localhost	*0	*0	*0	*0	0	2 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	198b	86.7k	29 Dec 26 16:16:12.576		
localhost	*0	*0	*0	*0	0	2 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	311b	83.6k	29 Dec 26 16:16:13.577		
localhost	*0	*0	*0	*0	0	1 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	112b	83.1k	29 Dec 26 16:16:14.576		
localhost	*0	*0	*0	*0	0	1 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	245b	83.1k	29 Dec 26 16:16:15.578		
localhost	*0	*0	*0	*0	0	0 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	111b	82.9k	29 Dec 26 16:16:16.580		
localhost	*0	*0	*0	*0	0	2 0	0.0%	0.0%	0	393G	44.0M	0 0	0 0	176b	84.5k	29 Dec 26 16:16:18.577		
localhost no data received host insert query update delete getmore command dirty used flushes vszie res qrw arw net_in net_out conn time																		
localhost	21890	*0	*0	*0	0	1 0	0.3%	0.3%	0	393G	59.0M	0 0	0 0	6,17m	84.9k	29 Dec 26 16:16:19.575		
localhost	22936	*0	*0	*0	0	0 0	0.6%	0.6%	0	393G	72.0M	0 0	0 0	6,48m	84.9k	29 Dec 26 16:16:20.577		
localhost	24082	*0	*0	*0	0	6 0	0.9%	0.9%	0	393G	86.0M	0 0	0 0	6,80m	87.1k	29 Dec 26 16:16:21.574		
localhost	11991	4	*0	*0	0	3 0	1.1%	1.1%	0	393G	93.0M	0 0	0 0	3,38m	99.2k	29 Dec 26 16:16:22.579		
localhost	*0	*0	*0	*0	0	3 0	1.1%	1.1%	0	393G	93.0M	0 0	0 0	312b	83.8k	29 Dec 26 16:16:23.577		
localhost	*0	*0	*0	*0	0	0 0	1.1%	1.1%	0	393G	93.0M	0 0	0 0	111b	82.8k	29 Dec 26 16:16:24.579		
localhost	*0	*0	*0	*0	0	2 0	1.1%	1.1%	0	393G	66.0M	0 0	0 0	247b	83.7k	29 Dec 26 16:16:25.575		
localhost	*0	*0	*0	*0	0	1 0	1.1%	1.1%	0	393G	56.0M	0 0	0 0	112b	83.2k	29 Dec 26 16:16:26.572		
localhost	*0	*0	*0	*0	0	0 0	1.1%	1.1%	0	393G	55.0M	0 0	0 0	111b	83.0k	29 Dec 26 16:16:27.572		
localhost	*0	*0	*0	*0	0	1 0	1.1%	1.1%	0	393G	54.0M	0 0	0 0	173b	82.9k	29 Dec 26 16:16:28.577		

Análisis del comando mongotop al importar “act-grupal-countries-big.json”

Al momento de importar los datos, se nos presenta un proceso en una sola fase que reflejan el tiempo total empleado, el cual es la suma del tiempo invertido en las operaciones de lectura y escritura realizadas por el procesador en ese instante preciso.

Primera fase: En esta etapa, se utilizó la operación de lectura durante 246 milisegundos y en segundo plano las operaciones de escritura generando un tiempo de 117 milisegundos con un total de la operación de 363 milisegundos.

Análisis:

A partir de los datos obtenidos, podemos concluir que las operaciones de lectura son más complejas que las de escritura, lo que conlleva un mayor tiempo de procesamiento por parte del procesador. Asimismo, se puede deducir que, durante el proceso de importación, la herramienta realiza una lectura completa del documento JSON antes de proceder a escribir los datos en la base de datos denominada ‘countries’.

```
mongotop 10
  ns      total    read   write   2024-12-26T17:18:05+00:00
  admin.$cmd.aggregate   0ms   0ms   0ms
  admin.atlascli        0ms   0ms   0ms
  admin.system.version  0ms   0ms   0ms
  config.collections    0ms   0ms   0ms
  config.system.sessions 0ms   0ms   0ms
  config.transactions   0ms   0ms   0ms
  countries.act-grupal-countries-big 0ms   0ms   0ms
  countries.act-grupal-countries-small 0ms   0ms   0ms
  inspections.act-grupal-city_inspections 0ms   0ms   0ms
  local.system.replset  0ms   0ms   0ms

  ns      total    read   write   2024-12-26T17:18:13+06:00
countries.act-grupal-countries-big 363ms  246ms  117ms
  config.collections   1ms   1ms   0ms
  admin.$cmd.aggregate 0ms   0ms   0ms
  admin.atlascli        0ms   0ms   0ms
  admin.system.version  0ms   0ms   0ms
  config.system.sessions 0ms   0ms   0ms
  config.transactions   0ms   0ms   0ms
  countries.act-grupal-countries-small 0ms   0ms   0ms
  inspections.act-grupal-city_inspections 0ms   0ms   0ms
  local.system.replset 0ms   0ms   0ms
```

Análisis del comando mongostat --discover al importar “act-grupal-countries-big.json”

Esta herramienta nos proporciona un seguimiento instantáneo del número de operaciones que se ejecutan por segundo en la base de datos. Durante la importación de la colección, se observó un alto volumen de inserciones, superando las 18,000 operaciones por segundo en la instancia local de MongoDB. El proceso se finalizó en dos fases distintas, consumiendo un promedio de solo 0.1% de la memoria disponible.

mongostat --discover																		
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.1%	0	393G 36.0M 0 0 0 0	299b	83.5k	35	Dec 26	17:17:52.625			
localhost	*0	*0	*0	*0	0	3 0	0.0%	1.1%	0	393G 37.0M 0 0 0 0	301b	87.1k	35	Dec 26	17:17:53.625			
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.1%	0	393G 37.0M 0 0 0 0	246b	83.5k	35	Dec 26	17:17:54.623			
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.1%	0	393G 36.0M 0 0 0 0	112b	83.3k	35	Dec 26	17:17:56.614			
localhost	no data received																	
localhost	*0	*0	*0	*0	0	0 0	0.0%	1.1%	0	393G 36.0M 0 0 0 0	110b	82.0k	35	Dec 26	17:17:57.626			
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.1%	0	393G 37.0M 0 0 0 0	303b	82.3k	35	Dec 26	17:17:58.642			
	host insert query update delete getmore command dirty used flushes vsize res qrw arw net_in net_out conn time																	
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.1%	0	393G 37.0M 0 0 0 0	114b	84.6k	35	Dec 26	17:17:59.623			
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.1%	0	393G 39.0M 0 0 0 0	297b	83.6k	35	Dec 26	17:18:00.624			
localhost	*0	*0	*0	*0	0	3 0	0.0%	1.1%	0	393G 36.0M 0 0 0 0	323b	84.0k	35	Dec 26	17:18:01.619			
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.1%	0	393G 36.0M 0 0 0 0	259b	83.2k	35	Dec 26	17:18:02.622			
localhost	*0	5	*0	*0	0	5 0	0.0%	1.1%	0	393G 44.0M 0 0 0 0	2.22k	88.4k	35	Dec 26	17:18:03.622			
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.1%	0	393G 44.0M 0 0 0 0	246b	83.5k	35	Dec 26	17:18:04.620			
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.1%	0	393G 42.0M 0 0 0 0	297b	83.4k	35	Dec 26	17:18:05.623			
localhost	*0	*0	*0	*0	0	0 0	0.0%	1.1%	0	393G 43.0M 0 0 0 0	111b	82.8k	35	Dec 26	17:18:06.625			
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.1%	0	393G 43.0M 0 0 0 0	112b	83.3k	35	Dec 26	17:18:07.622			
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.1%	0	393G 43.0M 0 0 0 0	307b	83.5k	35	Dec 26	17:18:08.624			
	host insert query update delete getmore command dirty used flushes vsize res qrw arw net_in net_out conn time																	
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.1%	0	393G 41.0M 0 0 0 0	112b	83.6k	35	Dec 26	17:18:09.617			
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.1%	0	393G 41.0M 0 0 0 0	296b	83.3k	35	Dec 26	17:18:10.622			
localhost	18045	*0	*0	*0	0	3 0	0.1%	1.3%	0	393G 49.0M 0 0 0 0	1.70m	85.5k	35	Dec 26	17:18:11.619			
localhost	3621	4	*0	*0	0	3 0	0.2%	1.3%	0	393G 51.0M 0 0 0 0	351k	88.3k	35	Dec 26	17:18:12.624			
localhost	*0	*0	*0	*0	0	4 0	0.2%	1.3%	0	393G 45.0M 0 0 0 0	303b	87.6k	35	Dec 26	17:18:13.620			
localhost	*0	*0	*0	*0	0	1 0	0.2%	1.3%	0	393G 45.0M 0 0 0 0	245b	83.2k	35	Dec 26	17:18:14.622			
localhost	*0	*0	*0	*0	0	3 0	0.2%	1.3%	0	393G 45.0M 0 0 0 0	298b	83.7k	35	Dec 26	17:18:15.622			
localhost	*0	*0	*0	*0	0	1 0	0.2%	1.3%	0	393G 45.0M 0 0 0 0	112b	83.1k	35	Dec 26	17:18:16.621			
localhost	*0	*0	*0	*0	0	0 0	0.2%	1.3%	0	393G 45.0M 0 0 0 0	111b	82.7k	35	Dec 26	17:18:17.625			
localhost	*0	*0	*0	*0	0	3 0	0.2%	1.3%	0	393G 45.0M 0 0 0 0	309b	84.1k	35	Dec 26	17:18:18.620			

Análisis del comando mongotop al importar “act-grupal-countries-small.json”

Al momento de importar los datos, se nos presenta un proceso en dos fases que reflejan el tiempo total empleado, el cual es la suma del tiempo invertido en las operaciones de lectura y escritura realizadas por el procesador en ese instante preciso.

Primera fase: En esta etapa, se utilizó la operación de lectura durante 60 milisegundos y en escritura de 0 milisegundos.

Segunda fase: En esta etapa, se utilizó la operación de lectura durante 13 milisegundos y una escritura de 2 milisegundos.

Análisis:

A partir de los datos obtenidos, podemos concluir que las operaciones cuando las colecciones son de menor complejidad o tamaño se pueden ejecutar en un mismo bloque, para este caso se uso mas tiempo para la lectura y poco segundos para la escritura.

```
mongotop 10
 0 0 0
          ns      total    read   write
config.system.sessions    0ms    0ms    0ms
config.transactions     0ms    0ms    0ms
countries.act-grupal-countries-big 0ms    0ms    0ms
countries.act-grupal-countries-small 0ms    0ms    0ms
inspections.act-grupal-city_inspections 0ms    0ms    0ms
local.system.replset     0ms    0ms    0ms

          ns      total    read   write
countries.act-grupal-countries-small 60ms   60ms   0ms
config.collections       7ms    7ms    0ms
admin.$cmd.aggregate     0ms    0ms    0ms
admin.atlascli           0ms    0ms    0ms
admin.system.version     0ms    0ms    0ms
config.system.sessions   0ms    0ms    0ms
config.transactions     0ms    0ms    0ms
countries.act-grupal-countries-big 0ms    0ms    0ms
inspections.act-grupal-city_inspections 0ms    0ms    0ms
local.system.replset     0ms    0ms    0ms

          ns      total    read   write
countries.act-grupal-countries-small 16ms   13ms   2ms
admin.$cmd.aggregate       0ms    0ms    0ms
admin.atlascli             0ms    0ms    0ms
admin.system.version       0ms    0ms    0ms
config.collections        0ms    0ms    0ms
config.system.sessions    0ms    0ms    0ms
config.transactions       0ms    0ms    0ms
countries.act-grupal-countries-big 0ms    0ms    0ms
inspections.act-grupal-city_inspections 0ms    0ms    0ms
local.system.replset      0ms    0ms    0ms

          ns      total    read   write
admin.$cmd.aggregate     0ms    0ms    0ms
admin.atlascli            0ms    0ms    0ms
admin.system.version     0ms    0ms    0ms
config.collections       0ms    0ms    0ms
config.system.sessions   0ms    0ms    0ms
config.transactions      0ms    0ms    0ms
countries.act-grupal-countries-big 0ms    0ms    0ms
countries.act-grupal-countries-small 0ms    0ms    0ms
inspections.act-grupal-city_inspections 0ms    0ms    0ms
local.system.replset     0ms    0ms    0ms
```

Análisis del comando mongostat --discover al importar “act-grupal-countries-small.json”

La herramienta reportó un promedio de [número] operaciones por segundo durante la importación de la colección. Se registró una inserción inicial y 5 consultas adicionales para validar los datos. La instancia local de MongoDB manejó la carga sin problemas, utilizando solo 1.3% de memoria y un promedio de 0.0% de CPU. El análisis del tráfico de red mostró 3.84 KB de datos entrantes y 91.3 KB de datos salientes.

mongostat --discover																	
host	insert	query	update	delete	getmore	command	dirty	used	flushes	vsize	res	qrw	arw	net_in	net_out	conn	time
localhost	*0	*0	*0	*0	0	4 0	0.0%	1.3%	0	393G 41.0M 0 0 0 0	457b	83.2K	35 Dec 26 17:20:21.020				
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 42.0M 0 0 0 0	112b	83.2K	35 Dec 26 17:20:22.618				
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 38.0M 0 0 0 0	112b	83.3K	35 Dec 26 17:20:24.617				
localhost	no data received																
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 38.0M 0 0 0 0	244b	82.7K	35 Dec 26 17:20:25.625				
localhost	*0	*0	*0	*0	0	3 0	0.0%	1.3%	0	393G 38.0M 0 0 0 0	298b	83.7K	35 Dec 26 17:20:26.624				
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 38.0M 0 0 0 0	112b	83.3K	35 Dec 26 17:20:27.621				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 38.0M 0 0 0 0	174b	83.4K	35 Dec 26 17:20:28.620				
localhost	host insert query update delete getmore command dirty used flushes vsize res qrw arw net_in net_out conn time																
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 38.0M 0 0 0 0	245b	83.3K	35 Dec 26 17:20:29.621				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 38.0M 0 0 0 0	164b	83.4K	35 Dec 26 17:20:30.620				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 39.0M 0 0 0 0	307b	83.5K	35 Dec 26 17:20:31.622				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 39.0M 0 0 0 0	260b	83.6K	35 Dec 26 17:20:32.620				
localhost	*0	*0	*0	*0	0	4 0	0.0%	1.3%	0	393G 39.0M 0 0 0 0	447b	87.0K	35 Dec 26 17:20:33.625				
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 39.0M 0 0 0 0	112b	83.2K	35 Dec 26 17:20:34.624				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 39.0M 0 0 0 0	247b	83.7K	35 Dec 26 17:20:35.619				
localhost	*0	5	*0	*0	0	4 0	0.0%	1.3%	0	393G 42.0M 0 0 0 0	2.22k	84.4K	35 Dec 26 17:20:36.626				
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 41.0M 0 0 0 0	112b	83.3K	35 Dec 26 17:20:37.623				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 41.0M 0 0 0 0	174b	83.4K	35 Dec 26 17:20:38.622				
localhost	host insert query update delete getmore command dirty used flushes vsize res qrw arw net_in net_out conn time																
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 41.0M 0 0 0 0	245b	83.1K	35 Dec 26 17:20:39.625				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 41.0M 0 0 0 0	164b	83.6K	35 Dec 26 17:20:40.622				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 41.0M 0 0 0 0	306b	83.4K	35 Dec 26 17:20:41.625				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 41.0M 0 0 0 0	260b	83.4K	35 Dec 26 17:20:42.625				
localhost	1	5	*0	*0	0	7 0	0.0%	1.3%	0	393G 43.0M 0 0 0 0	3.84k	91.3K	35 Dec 26 17:20:43.623				
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 42.0M 0 0 0 0	112b	83.1K	35 Dec 26 17:20:44.622				
localhost	*0	*0	*0	*0	0	1 0	0.0%	1.3%	0	393G 42.0M 0 0 0 0	244b	83.0K	35 Dec 26 17:20:45.626				
localhost	*0	*0	*0	*0	0	3 0	0.0%	1.3%	0	393G 42.0M 0 0 0 0	298b	83.9K	35 Dec 26 17:20:46.623				
localhost	*0	*0	*0	*0	0	0 0	0.0%	1.3%	0	393G 43.0M 0 0 0 0	111b	82.8K	35 Dec 26 17:20:47.625				
localhost	*0	*0	*0	*0	0	2 0	0.0%	1.3%	0	393G 43.0M 0 0 0 0	174b	83.5K	35 Dec 26 17:20:48.624				
localhost	host insert query update delete getmore command dirty used flushes vsize res qrw arw net_in net_out conn time																

Exportar la base de datos

Usando comando mongoexport

Para validar el funcionamiento de este comando, se siguió la sugerencia de la actividad y se generó el siguiente comando: “mongoexport --db inspections --collection act-grupal-city_inspections --out inspections.json;” Al obtener el resultado esperado, se adjunta una captura de pantalla para mostrar la operación.

```
apple ~ % /opt/homebrew/var/log/mongodb on master
> mongoexport --db inspections --collection act-grupal-city_inspections --out inspections.json;
2024-12-26T18:30:13.913-0600    connected to: mongodb://localhost/
2024-12-26T18:30:14.511-0600    exported 81047 records
>
apple ~ % /opt/homebrew/var/log/mongodb on master
> ls -l
total 47928
-rw-r--r--  1 leonard  admin  24269552 Dec 26 18:30 inspections.json
-rw-----  1 root     admin    246908 Dec 26 18:30 mongo.log
-rw-r--r--  1 leonard  admin        0 Dec 16 22:15 output.log
>
```

Restaurar una base de datos

Usando comando mongoexport

Siguiendo las indicaciones de la actividad, se descomprimió el archivo comprimido 'act-grupal-people.zip', obteniendo el archivo en formato BSON 'people.bson'. A continuación, se ejecutó el comando 'mongorestore people.bson' desde la ruta donde se encontraba el archivo descomprimido para verificar su correcto funcionamiento.

```
mongorestore people.json  
2024-12-26T18:48:44.098-0600 checking for collection data in people.json  
2024-12-26T18:48:44.099-0600 reading metadata for test.people from people.metadata.json  
2024-12-26T18:48:44.145-0600 Restoring test.people from people.json  
2024-12-26T18:48:46.504-0600 finished restoring test.people (1000000 documents, 0 failures)  
2024-12-26T18:48:46.504-0600 no indexes to restore for collection test.people  
2024-12-26T18:48:46.504-0600 1000000 document(s) restored successfully. 0 document(s) failed to restore.
```

The screenshot shows the MongoDB Compass interface. The left sidebar displays connections, with 'cluster0.nhica.mongodb.net' and 'LOCAL' selected. Under 'LOCAL', several databases are listed: admin, config, countries, inspections, local, and test. The 'people' collection is currently selected under the test database. The main pane shows the 'Documents' tab with 1.0M documents. A search bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE. The results list shows five documents, each with an '_id' field (ObjectId), 'padding' (a large hex string), 'signup' (date/time), 'points' (integer), and 'name' (string). The first document's name is 'Dan', and its points are 656304. The second document's name is 'Nick' and its points are 30922. The third document's name is 'Nick' and its points are 30922. The fourth document's name is 'Wendy' and its points are 707029. The fifth document's name is 'Wendy' and its points are 690231.

_id	padding	signup	points	name
ObjectId('57605d5dc3d5a2429db0bcfc')	"00..."	2015-01-18T15:34:35.000+00:00	656304	Dan
ObjectId('57605d5dc3d5a2429db0bcfd')	"00..."	2013-09-04T04:35.12.000+00:00	30922	Nick
ObjectId('57605d5dc3d5a2429db0bcfe')	"00..."	2013-09-04T04:35.12.000+00:00	30922	Nick
ObjectId('57605d5dc3d5a2429db0bcff')	"00..."	2013-05-03T04:46:28.000+00:00	690231	Wendy

Caso de uso

Restricción de terrazas en Madrid por covid-19. Contratan a vuestro equipo para actualizar las restricciones de ciertos locales y terrazas en Madrid por cuestiones del covid-19. Los datos para actualizar están en el fichero llamado act-grupal-openDataLocalesMadrid.csv y se os pide que consolidéis dichos cambios en una base de datos MongoDB llamada Madrid con la colección Terrazas. La información del dataset a utilizar lo tenéis en la siguiente URL: Open Data Censo de locales, sus actividades y terrazas de hostelería y restauración (Terrazas).

Se genera la Base de datos Madrid y su colección Terrazas.

```
test> use Madrid
switched to db Madrid
Madrid> db.createCollection('Terrazas');
{ ok: 1 }
Madrid> show collection
MongoshInvalidInputError: [COMMON-10001] 'collection' is not a valid argument for "show".
Madrid> show collections
Terrazas
Madrid> |
```