Actividad 3 - Uso avanzado de BD NoSQL

Actividad del Proyecto

**Actividad Grupal**

**Presentación**

Esta actividad está asociada al proyecto transversal del título y para su desarrollo, tendrás que utilizar algunos de los siguientes recursos:

1. La base de datos generada en la **Actividad 2 – Laboratorio** de esta asignatura.
2. Los ficheros adjuntos en esta actividad.

Para el desarrollo de la actividad, revisa los objetivos del proyecto y el ámbito de este para la organización de los datos. Recuerda que los dataset no tienen por qué estar relacionados, la información elegida debe cubrir los entregables propuestos en esta actividad.

Objetivos

Esta actividad os va a permitir utilizar funcionalidades avanzadas de las bases de datos NoSQL, en concreto MongoDB y Neo4J. Realizaréis una serie de ejercicios donde podréis monitorizar la actividad de la base de datos MongoDB y diseñaréis un grafo que permita visualizar un conjunto de datos extraídos en Neo4J.

Descripción de la actividad y pautas de elaboración

Para esta actividad, vais a procesar diferentes ficheros sobre MongoDB mientras monitorizáis la actividad de la base de datos. Finalmente, con un conjunto de datos extraídos de MongoDB, tendréis que crear un grafo en Neo4J que muestre determinadas relaciones entre los datos.

La actividad la tendréis que realizar desarrollando los apartados a continuación:

**Prepara el entorno de trabajo**

1. **Opción 1**: descarga una imagen Docker de MongoDB versión más reciente y levanta un contenedor con dicha versión.
2. **Opción 2**: descarga e instala MongoDB en su versión más reciente en tu ordenador local.
3. **Instala las *Tools* de MongoDB:** en la siguiente URL podrás descargar las *tools* de MongoDB necesarias para este laboratorio:
   1. <https://docs.mongodb.com/database-tools/installation/installation/>
   2. <https://www.mongodb.com/try/download/database-tools>

Este paso seguro ya lo has hecho, elegid en grupo la mejor opción para trabajar en este laboratorio. No es necesario documentar en el trabajo final.

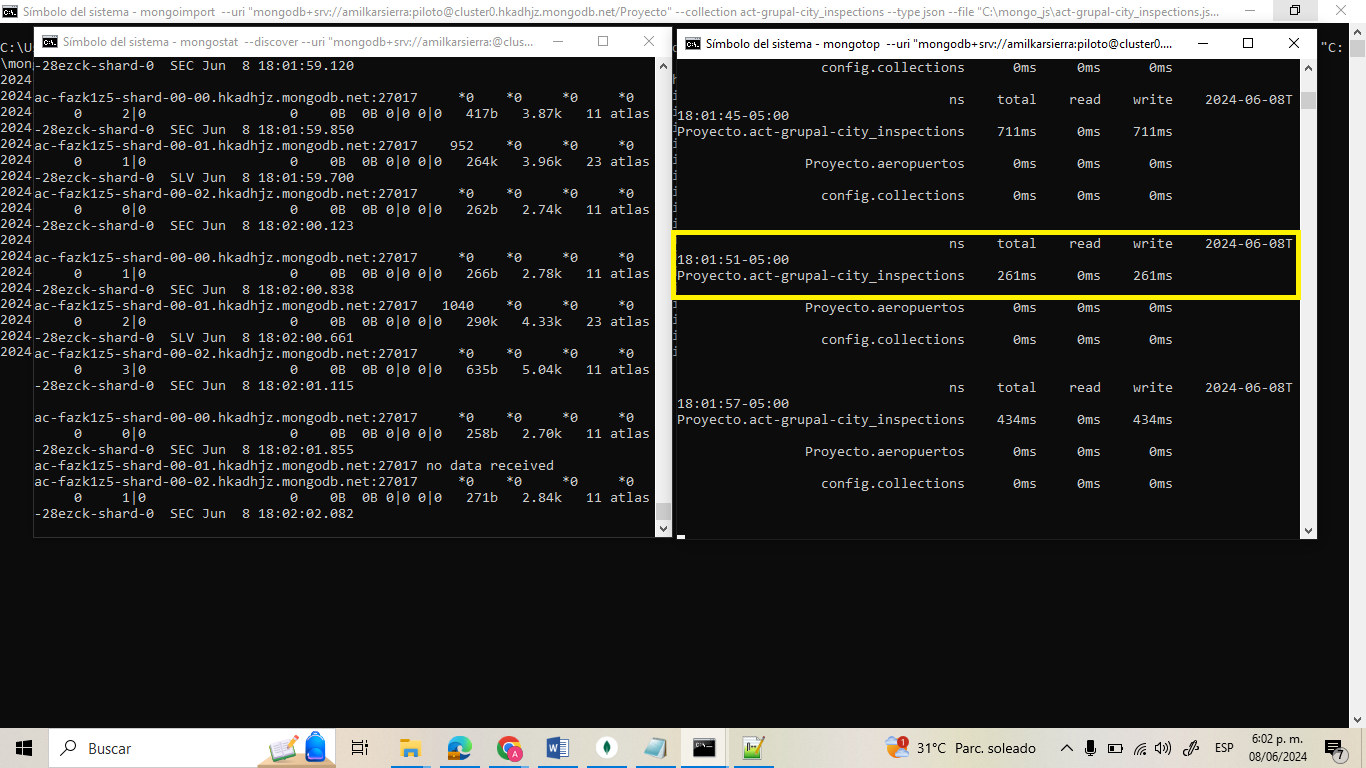
1. **Inicia la monitorización en MongoDB,** ejecuta las siguientes instrucciones:
   1. En una consola diferente, visualiza las estadísticas con la instrucción mongostat --discover

\*\*\*

mongostat --discover --uri "mongodb+srv://amilkarsierra:piloto@cluster0.hkadhjz.mongodb.net/Proyecto" \*\*\*\*

* 1. En otra consola distinta, visualiza el *top* de las operaciones contra la base de

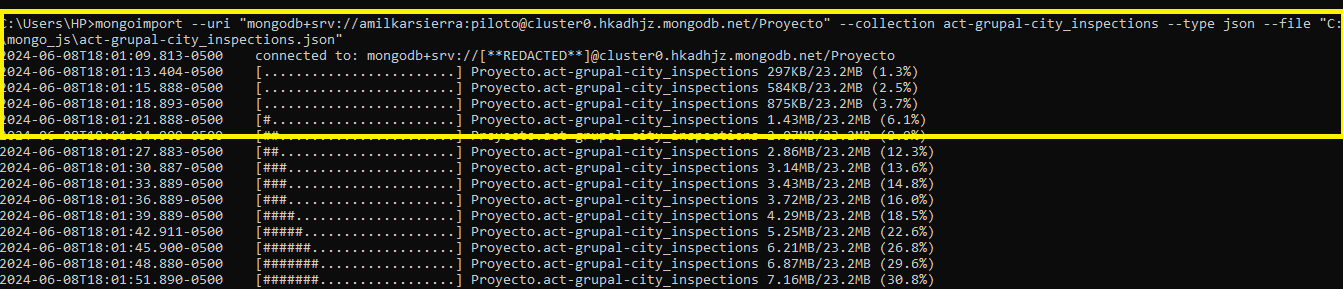
datos, utiliza la instrucción mongotop 5 donde 5 es el tiempo en segundos que debe refrescar la visualización.

Mongotop --uri "mongodb+srv://amilkarsierra:piloto@cluster0.hkadhjz.mongodb.net/Proyecto" 5 

* 1. **Tened estas dos consolas siempre activas y visibles mientras realizáis los siguientes puntos.** **Interpretad su salida** y describid **brevemente** qué muestran cuando se ejecutan dichas instrucciones (utilizad capturas para ilustrar vuestra respuesta).

1. **Importar colecciones**:
   1. En la base de datos *Proyecto* importa la colección que corresponde a los datos del fichero *act-grupal-city\_inspections.json*.

C:\Users\HP>mongoimport --uri "mongodb+srv://amilkarsierra:piloto@cluster0.hkadhjz.mongodb.net/Proyecto" --collection act-grupal-city\_inspections --type json --file "C:\mongo\_js\act-grupal-city\_inspections.json"



* 1. Importa también las colecciones que corresponden a los ficheros *act-grupal-countries-small.json* y act-g*rupal-countries-big.json*.

mongoimport --uri "mongodb+srv://amilkarsierra:piloto@cluster0.hkadhjz.mongodb.net/Proyecto" --collection aact-grupal-countries-small --type json --file "C: \mongo\_js\act-grupal-countries-small.json"

mongoimport --uri "mongodb+srv://amilkarsierra:piloto@cluster0.hkadhjz.mongodb.net/Proyecto" --collection act-grupal-countries-big.json --type json --file "C: \mongo\_js\act-grupal-countries-big.json"

Reto: propón una única instrucción en el punto a y en el b. En el b amplía tu visión y saca provecho a las bondades de *mongoshell.*

#!/bin/bash

COLECCIONES=("act-grupal-countries-small.json","act-grupal-countries-big.json")

for COLECCION in "${COLECCIONES [@]}"

do

mongoimport --uri "mongodb+srv://amilkarsierra:piloto@cluster0.hkadhjz.mongodb.net/Proyecto" --collection ${COLECCION%.json} --type json --file "C: /mongo\_js/${COLECCION}.json"

done

1. **Exportar la base de datos:**
   1. Haced un *export* de todas las colecciones de la base de datos *Proyecto*.

Nota: el ejecutable que permite hacer el *export* es *mongoexport*, la sintaxis es: mongoexport --collection=<name-collection> --db=<name-db> --out=<name-file>.json

1. **Restaurar una base de datos:**
   1. Para restaurar una base de datos, debéis indicar la ruta del directorio *dump* donde se almacenan los ficheros de la base de datos *(json y bson)*.
      * dump
        1. people
        2. --people.bson
        3. --people.metadata.json
   2. Descomprimid el fichero *act-grupal-people.zip* y ubicad los ficheros en la estructura de directorios indicada en el paso anterior.

Nota: recordad el uso de la instrucción: mongorestore --port xxxxx PATH\dump

1. **Caso de uso:** restricción deterrazas 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.cvs* y se os pide que consolidéis dichos cambios la base de datos *Proyecto* sobre la colección *Terrazas.*

La información del dataset a utilizar lo tenéis en las siguientes URL:

* + Open Data Censo de locales, sus actividades y terrazas de hostelería y restauración (Terrazas)
  + <https://datos.madrid.es/FWProjects/egob/Catalogo/Economia/Ficheros/Estructura_DS_FicheroCLA_Terrazas.pdf>

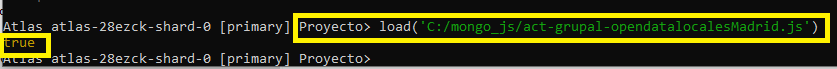
5.1. En el alcance del contrato se os pide:

* 1. Convertid el fichero CSV a formato JSON. Utilizad, por ejemplo, el servicio de <https://csvjson.com/csv2json>
  2. Descargad el fichero generado en el paso anterior, en formato JSON indicando la opción Array (todos los documentos estarán dentro de un único *array*).
     + En el fichero hay un problema de espacios en muchos campos String, intentad corregirlo en esta etapa.
  3. Editad el fichero generado y al principio del todo añadid la línea:

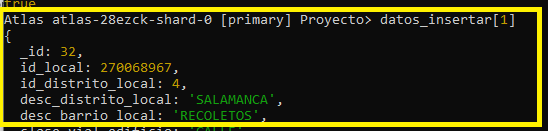
var datos\_insertar = […]

* 1. Guardad el fichero en una ruta conocida y cambiad su extensión a formato .js. ¿Sabéis qué tipo de fichero es un js?
  2. Abrid el cliente *mongo* y ejecutad las siguientes instrucciones:
     + load(“PATH\fichero.js”)
     + datos\_insertar[0]
     + Describid brevemente qué hacen las instrucciones anteriores. Con “load” cargamos el archivo.

load(‘C:/mongo\_js/act-grupal-openDataLocalesMadrid.js’)



y con “datos\_insertar[0]”, accededmos al elemento de la posición del arreglo, en este caso, la posicio[1].



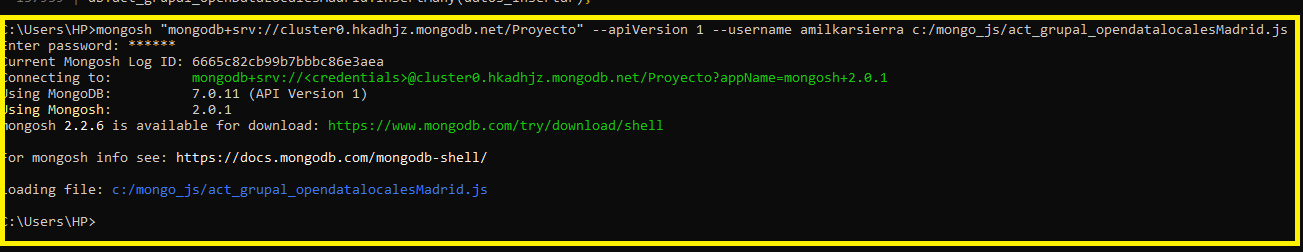
* 1. Con el elemento datos\_insertar, ¿podríais realizar un *insert* masivo? ¿Cómo?

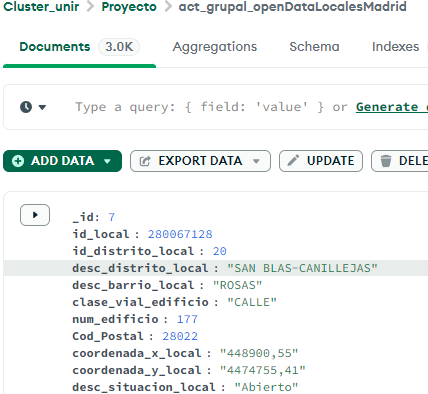
RTA/: Si podemos, le enviamos la variable “datos\_insert” a la función insertMany.

db.act-grupal-openDataLocalesMadrid.insertMany(datos\_insertar);

Desde la terminal, podríais proponer una alternativa que evite realizar los pasos anteriores (descripción de dos líneas máximo y no vale indicar el uso MongoCompass 😊).

Rta:/ Aca estamos conectando el cluster desde el Shell de mongo, al base de datos “Proyecto”, y llamamos el act\_grupal\_opendatalocalesMadrid.js c:>mongosh "mongodb+srv://cluster0.hkadhjz.mongodb.net/Proyecto" --apiVersion 1 --username amilkarsierra c:/mongo\_js/act\_grupal\_opendatalocalesMadrid.js



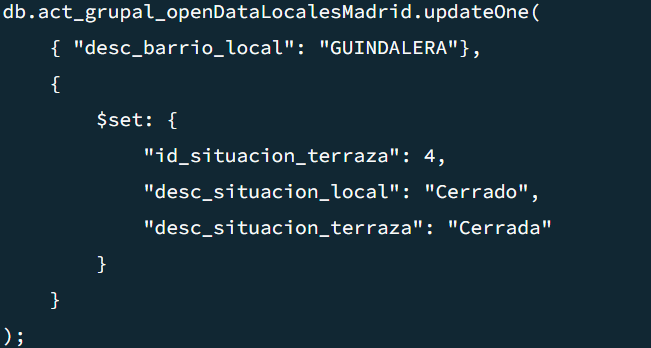


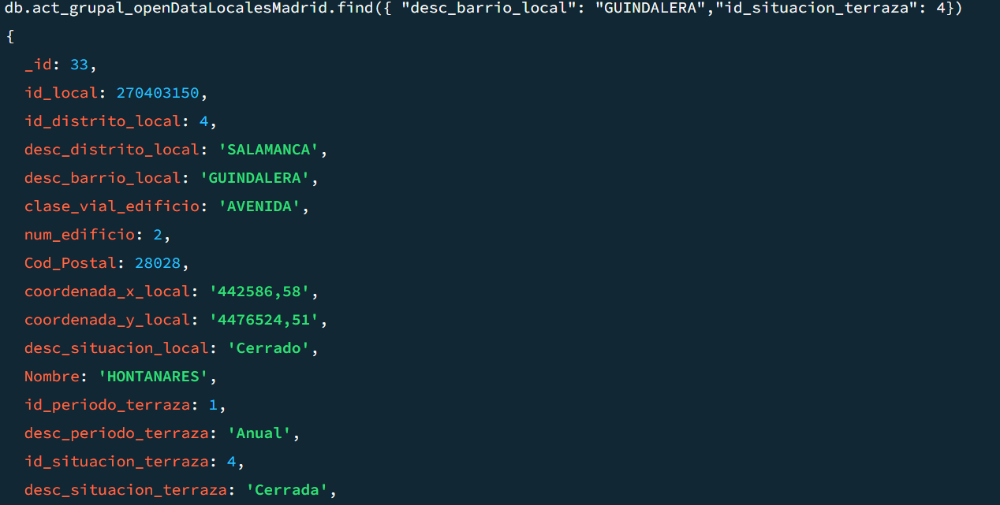
5.2. Realizad las siguientes actualizaciones sobre los datos (cuando los cambios afecten a campos \_es o \_ra, haced el *update* sobre los datos del local en período estacional):

* 1. Los locales del barrio Guindalera de Salamanca, por motivos de la desescalada, no podrán abrir y tendrán que permanecer cerrados. Cambiad la situación del local a *Cerrado* utilizando el *id* correspondiente.
     + Recordad que debéis constatar los tipos de situaciones que existan y seguir con la misma codificación.
     + A estos mismos locales, cambiad la situación de la terraza a (id 4) “Cerrado” *cerrada* siguiendo las mismas premisas anteriores.
* Lo primero es realizar una búsqueda con la siguiente sentencia para saber que documentos pertenecen al barrio Guindalera de Salamanca



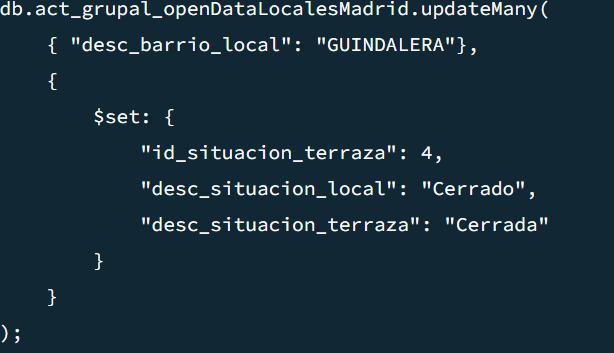
* Después procedemos a realizar la consulta para modificar los cambios con un solo documento de prueba para comprobar la salida esperada



Resultado

Cambios realizados en los campos solicitados

Viendo que se realizaron los documentos correctamente procedemos a modificar los documentos que cumplan con la condición

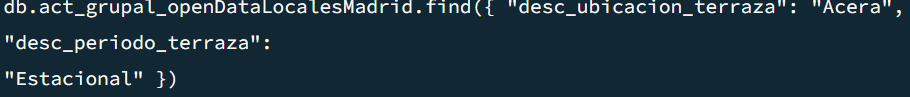


* 1. A todas las terrazas que se ubiquen en la *acera,* añadid un campo llamado *inspeccionar* y estableced el siguiente valor:
     + Si dispone de más de 10 mesas, true.
     + Si dispone de menos de 10 mesas, false.

Vemos que los cambios afectan a los documentos dependiendo las mesas que su atributo es **mesas\_es.**

**Hay que tener encuenta esta especificación (cuando los cambios afecten a campos \_es o \_ra, haced el update sobre los datos del local en período estacional):**

* Lo primero es realizar una búsqueda con la siguiente sentencia para saber cuales documentos son de Periodo Estacional y se ubican en Acera

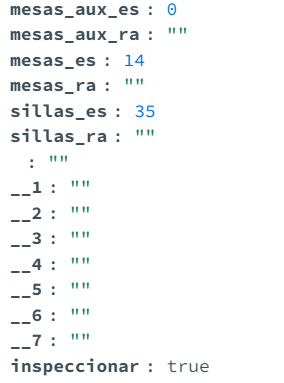
****

* Ya sabiendo cuales son los documentos procedemos a modificarlos documentos que cumplen con la condición

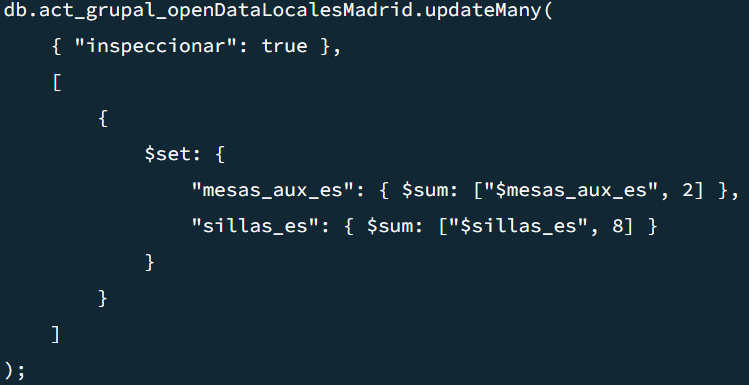


Creamos el campo inspeccionar y le indicamos los parámetros con la sentencia **$cond**

Salida esperada



* 1. A las terrazas que se deban inspeccionar, añadid 2 mesas más auxiliares y 8 sillas más disponibles (tanto en período estacional como el resto del año).



Lo que hacemos es sumar 2 mesas auxiliares y 8 sillas alas que ya el documento tiene

* 1. A las terrazas que no deban ser inspeccionadas, añadid el campo *estado* con el valor:
     + 1 si el número de sillas es menor que 10.
     + 2 si el número de sillas está entre 10 y 20.
     + 3 si cuenta con más de 20 sillas.

Creación de un switch que cumplan las condiciones establecidas en la variable estado



Salida esperada con la condición antes mencionada

* 1. De lunes a jueves, ningún local podrá cerrar más allá de las 00:00:00, actualizad el horario de cierre a esta nueva hora límite.

Buscar los documentos con hora de cierre mayor 00:00:00



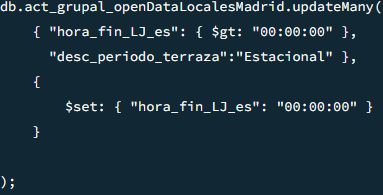
Salida esperada documentos con hora de salida de lunes a jueves mayor a 00:00:00



Modificamos los documentos para que cierren ala hora 00:00:00

Vemos que los cambios afectan a los documentos dependiendo la hora final de lunes a jueves que su atributo es hora\_fin\_LJ\_es.

Hay que tener encuenta esta especificación **(cuando los cambios afecten a campos \_es o \_ra, haced el update sobre los datos del local en período estacional):**

****

**Salida esperada**

****

* 1. De viernes a sábado, los locales que cierren a las 2:30:00 ahora tendrán que hacerlo a las 2:00:00.

Buscar los documentos con hora de cierre de 2:30:00



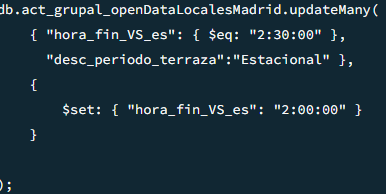
Salida esperada documentos con hora de salida de viernes a sábado de 2:30:00



Modificamos los documentos para que cierren a la hora 2:30:00

Vemos que los cambios afectan a los documentos dependiendo la hora final de lunes a jueves que su atributo es hora\_fin\_LJ\_es.

Hay que tener encuenta esta especificación **(cuando los cambios afecten a campos \_es o \_ra, haced el update sobre los datos del local en período estacional):**



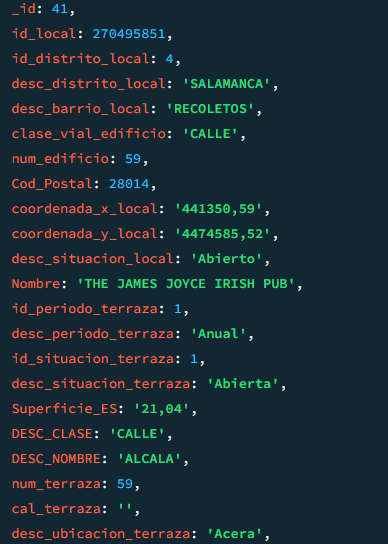
**Salida esperada**

* 1. A todos los locales que estén sobre la calle Alcalá en Madrid se les debe inspeccionar.

Buscar los locales que estén sobre la calle Alcalá en Madrid

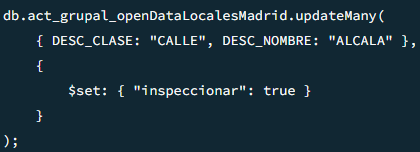


Salida esperada



Campos con los que identificamos si el edificio estén sobre la calle Alcalá

Agregamos la variable inspeccionar



* 1. A todos los locales con terrazas abiertas, añadid un campo *revision* cuyo valor sea un documento con la siguiente información: {prox\_inspeccion: 10, puntuacion: 80, comentario: “separar las mesas”}

Buscar documentos que cuya terraza este abierta

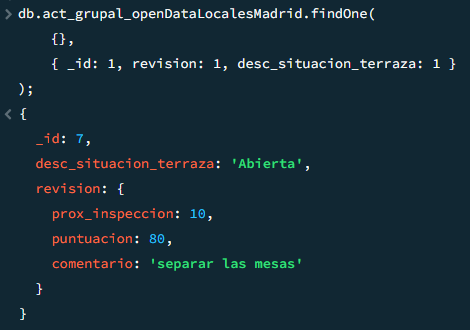


Salida esperada

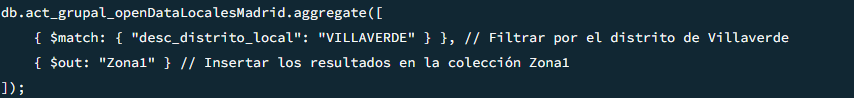


Campo que nos indica si el documento tiene la terraza abierta

Ver la salida del campo agregado



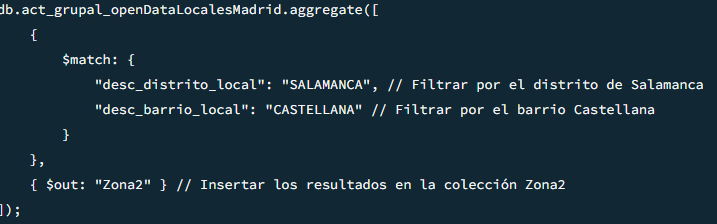
* 1. Cread una nueva colección llamada *Zona1,* con todos los locales que pertenezcan al distrito de Villaverde.



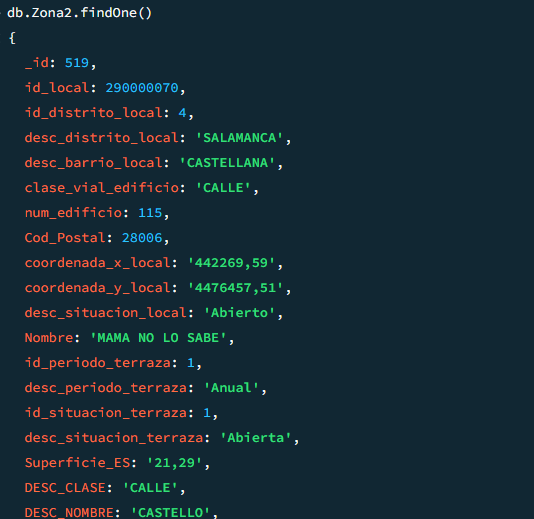
Colección creada: Buscamos un documento para ver si creamos la colección



* 1. Cread una nueva colección llamada *Zona2,* con todos los locales que pertenezcan al distrito de Salamanca y barrio Castellana.



Colección creada: Buscamos un documento para ver si creamos la colección



1. En Neo4J, proponed un modelo de datos donde sea posible visualizar los locales de cada barrio y los tipos de terrazas que estos tienen, **además de cualquier otra información relacionada con el proyecto del máster**. Indicad los nodos y sus relaciones, cread una única consulta que visualice el grafo total (incluidas las relaciones). Añadid tanto a nodos como a relaciones sus atributos, para ello elegid los campos que más convengan a vuestro modelo.

**Modelo de Grafo**

* **Local:**

**Atributos:** id\_local, nombre, desc\_situacion\_local, num\_edificio, coordenada\_x, coordenada\_y, desc\_clase, desc\_nombre

**Descripción:** Representa un local específico con sus detalles.

* **Barrio:**

**Atributos:** id\_barrio, nombre\_barrio

**Descripción:** Representa un barrio dentro de un distrito.

* **Distrito:**

**Atributos:** id\_distrito, nombre\_distrito

**Descripción:** Representa un distrito que contiene varios barrios.

* **Terraza:**

**Atributos:** id\_terraza, desc\_situacion\_terraza, superficie, cal\_terraza, desc\_ubicacion\_terraza, hora\_ini\_LJ, hora\_fin\_LJ, hora\_ini\_VS, hora\_fin\_VS, mesas, sillas

**Descripción:** Representa una terraza asociada a un local, incluyendo detalles sobre su situación y horario.

Relaciones:

**LOCAL\_EN\_BARRIO:**

Descripción: Relación que indica que un local está ubicado en un barrio específico.

**BARRIO\_EN\_DISTRITO:**

Descripción: Relación que indica que un barrio pertenece a un distrito específico.

**TIENE\_TERRAZA:**

Descripción: Relación que indica que un local tiene una terraza.

**Descripción del Modelo:**

Cada Local está asociado a un Barrio a través de la relación **LOCAL\_EN\_BARRIO.**

Cada Barrio está asociado a un Distrito a través de la relación **BARRIO\_EN\_DISTRITO.**

Cada Local tiene una o más Terrazas a través de la relación **TIENE\_TERRAZA.**

**De este punto se espera**:

* 1. Un grafo genérico o diagrama que represente vuestro modelo y donde se reflejen todos los campos utilizados (nodos, relaciones y atributos).
  2. Una breve descripción del modelo propuesto.
  3. La descripción del proceso utilizado para cargar los datos en Neo4J a partir de los datos en MongoDB.
  4. Una consulta CQL y capturas con la visualización del grafo generado con los datos. Al menos debe haber una captura con un gran número de elementos del grafo.

Entrega y extensión

Elaborad un informe con los elementos habituales (portada, índice, nombre, fecha, numeración de páginas, etc.), en el que deis respuesta a los apartados planteados anteriormente. Cada uno de los apartados se debe responder en una sección diferente. En cada apartado responded a los distintos literales que se proponen. **La respuesta debe incluir**: la respuesta misma al literal si la hay, la consulta realizada *(script)* y, cuando así se indique, una captura de pantalla con el resultado. **En el caso de que no respondáis a un literal concreto, ponedlo en la memoria y dejadlo en blanco. Todos los *scripts* de las consultas deben agruparse por sección en un único documento txt.**

Utilizad las páginas que necesitéis para redactar el informe. No se permiten anexos. Fuente utilizada Georgia 12 e interlineado simple.

Al finalizar la actividad grupal, **todos los miembros del equipo harán la entrega del mismo informe final** a través del apartado «Envío de actividades» del aula virtual. El informe y *script* para entregar deben ir nombrados como se muestra a continuación:

* APELLIDO1\_APELLIDO2\_NOMBRE\_actividad\_grupal.pdf (en pdf)
* APELLIDO1\_APELLIDO2\_NOMBRE\_actividad\_grupal.txt

No uséis tildes, apóstrofes ni ningún otro carácter que pudiera resultar conflictivo.

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad grupal: Uso avanzado de bases de datos NoSQL | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Apartados 1, 2, 3 y 4:  - Cada literal tiene una respuesta clara, breve y correcta. | 3.0 | 30 % |
| Criterio 2 | Apartado 5:  - Cada literal tiene una respuesta clara, breve y correcta.  - Se observa la consulta o *script* utilizado, la cual se ejecuta sin error alguno. | 3.5 | 35 % |
| Criterio 3 | Apartado 6:  - Se propone un modelo simple donde se muestran los nodos, las relaciones y los atributos propuestos.  - La consulta muestra claramente las relaciones.  - La consulta o *script* se ejecuta sin error alguno. | 3.5 | 35 % |
|  |  | **10** | **100 %** |

Todos los miembros del equipo deben hacer la entrega en el aula virtual y han de adjuntar el mismo documento.

Se debe indicar en el informe y en el *script* el nombre de todos los componentes del equipo. En el informe, además, se adjuntará cumplimentada la siguiente tabla de valoración individual.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sí** | **No** | **A veces** |
| Todos los miembros se han integrado en el trabajo del grupo. |  |  |  |
| Todos los miembros participan activamente. |  |  |  |
| Todos los miembros respetan otras ideas aportadas. |  |  |  |
| Todos los miembros participan en la elaboración del informe. |  |  |  |
| Me he preocupado por realizar un trabajo cooperativo con mis compañeros. |  |  |  |
| Señala si consideras que algún aspecto del trabajo en grupo no ha sido adecuado. |  |  |  |