iesgrancapitan-CEIABD-BDA / ud2-practica-1-mongo-db-Dansarasi...

Code Issues Pull requests Actions Projects Wiki Security

main ud2-practica-1-mongo-db-Dansarasix-DML / Readme.md

Dansarasix-DML change imgs 8122223 · 8 months ago

244 lines (171 loc) · 8.18 KB

Preview Code Blame Raw

Big Data Aplicado

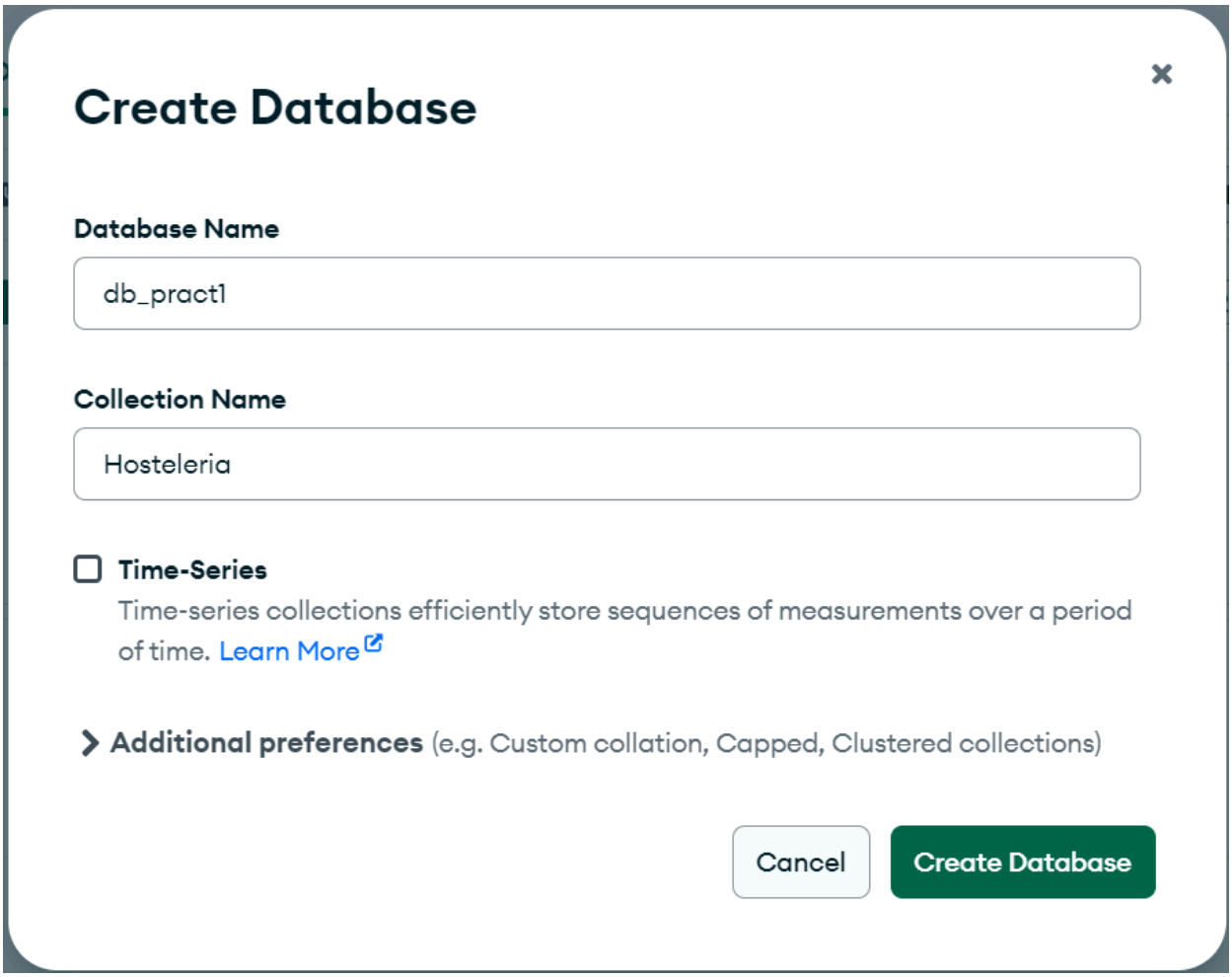
UD2 - Procesado y Presentación de Datos Almacenados

Práctica 1 MongoDB

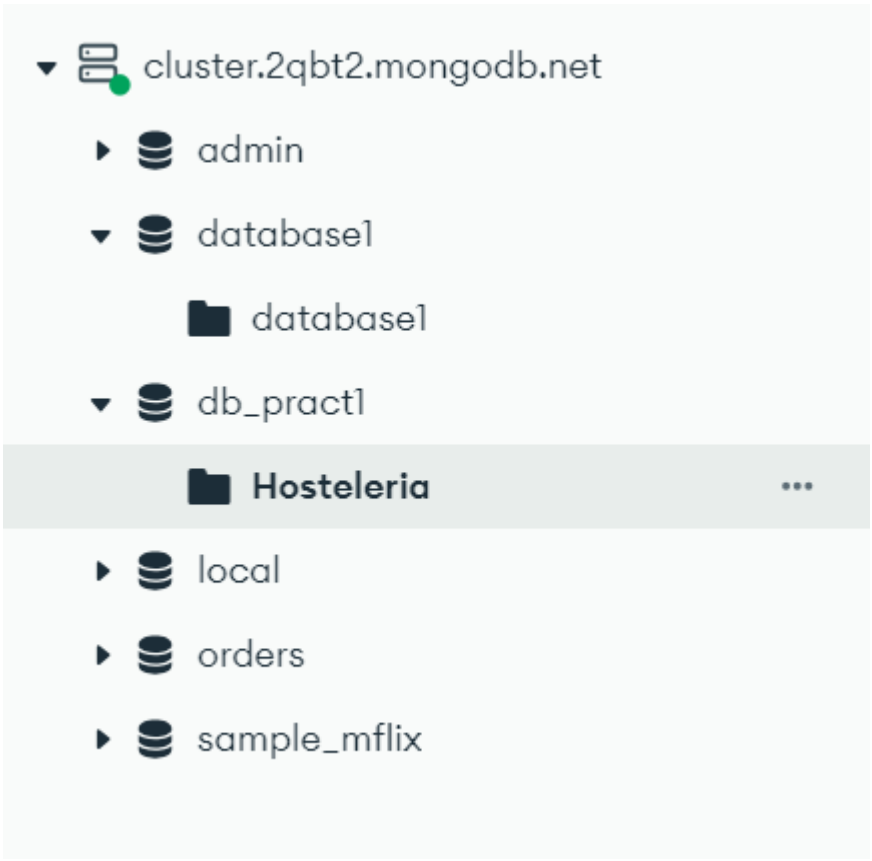
> Recuerda que para hacer la prácticas puedes optar por cualquiera de la las 3 opciones.
>
> - Instalarla en tu máquina local.
> - Usar Atlas MongoDB.
> - Crear un contenedor docker.

1. Crear una base de datos MongoDB llamada “db_pract1”, así como una colección con nombre "Hosteleria" en la que importes el contenido del archivo dataset_restaurantes.json

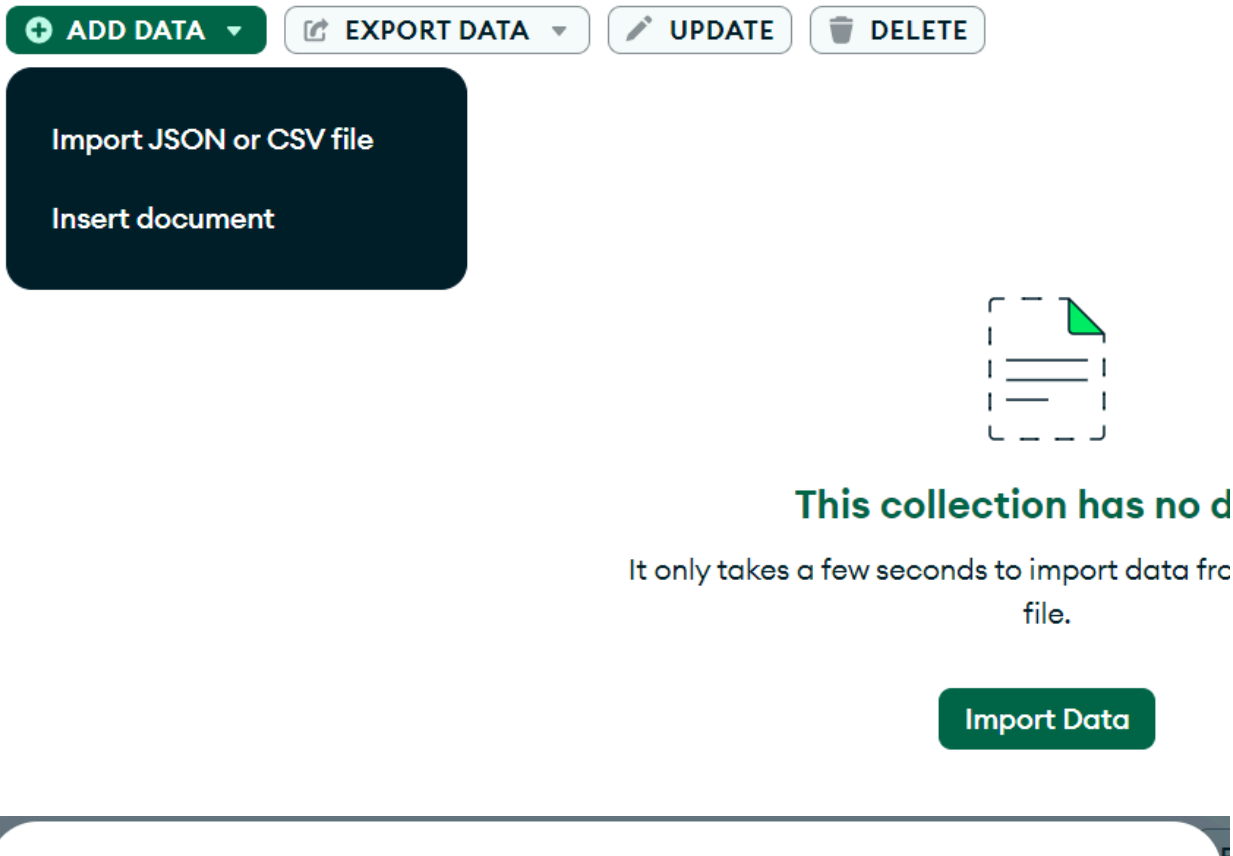
Creamos la base de datos en nuestro cluster en Mongo Atlas usando Compass.

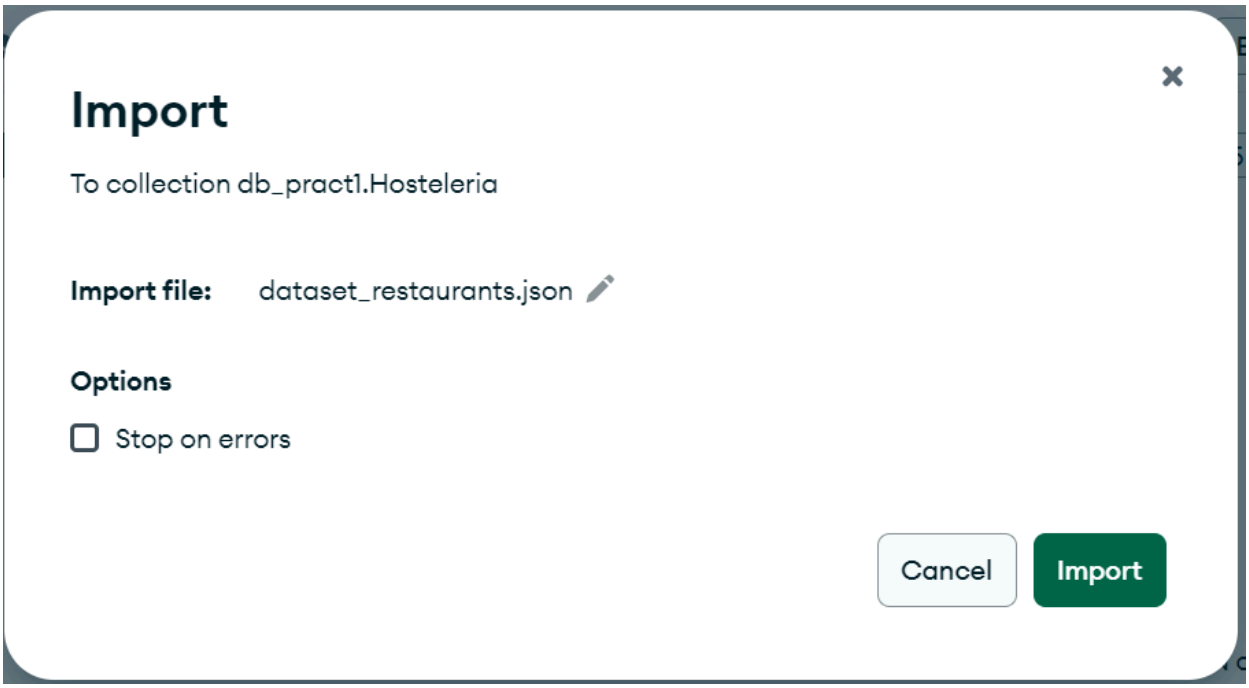


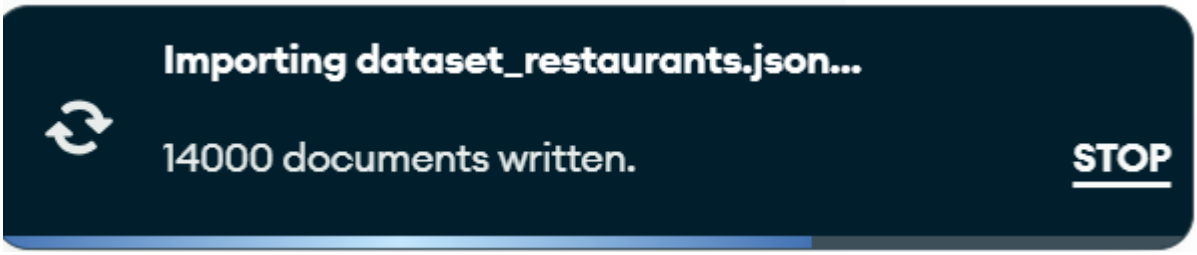
Y en un momento se crea nuestra base de datos con nuestra colección



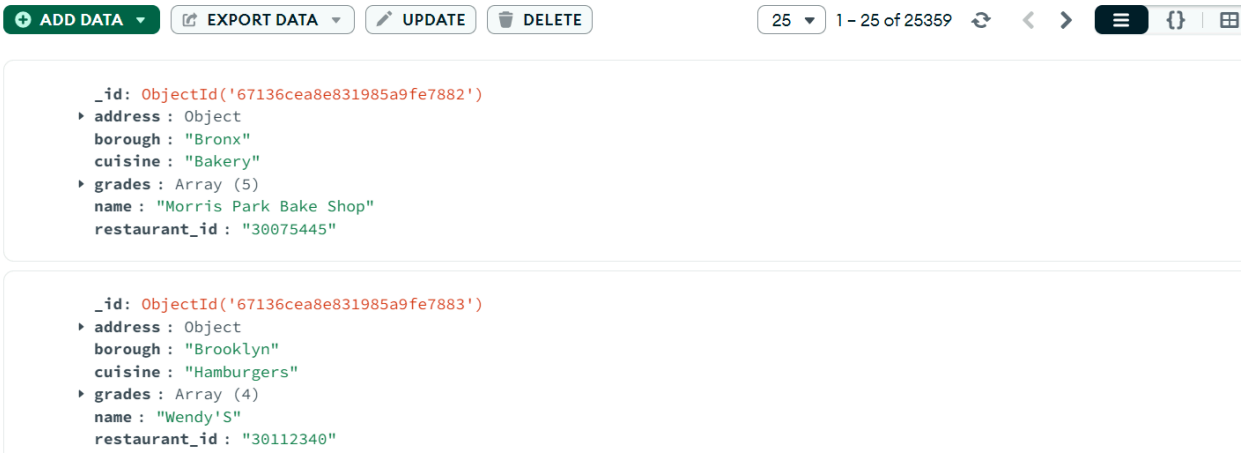
Ahora importamos la data del json.







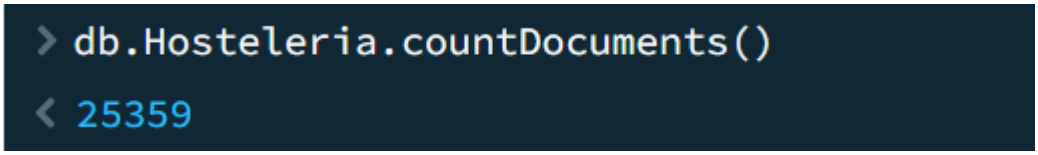
Y ya aparece la data en nuestra base de datos.



1. Sobre la base de datos creada anteriormente, realizar las siguientes consultas:
    - a. Obtén el número total de documentos que se han importado a la colección Restaurantes

Usamos la función `countDocuments()` para contar el número de documentos que se subieron.

```
db.Hosteleria.countDocuments()
```



    - b. Mostrar los datos de todos los restaurantes de cocina española (valor “Spanish”). ¿Cuántos restaurantes de este tipo de cocina hay en la colección? ¿Cuántos restaurantes sobre el total suponen los restaurantes de cocina española?

Usamos la función `find()` junto al valor que deseamos buscar.

```
db.Hosteleria.find( { cuisine: "Spanish" } )
```

```
> db.Hosteleria.find( { cuisine: "Spanish" } )
< {
    _id: ObjectId('67136cea8e831985a9fe78ed'),
    address: {
      building: '113',
      coord: [
        -73.9979214,
        40.7371344
      ],
      street: 'West   13 Street',
      zipcode: '10011'
    },
    borough: 'Manhattan',
    cuisine: 'Spanish',
    grades: [
      {
```

Para contar estos documentos usamos el mismo comando anterior pero añadiendo el valor que deseamos como el que pusimos en `find()` .

```
db.Hosteleria.countDocuments({ cuisine: "Spanish" })
```

```
> db.Hosteleria.countDocuments({ cuisine: "Spanish" })
< 637
```

MongoDB también permite hacer cálculos, por lo que para la última pregunta podemos juntar los comandos anteriores en una división y multiplicarlo por 10.

```
(db.Hosteleria.countDocuments({ cuisine: "Spanish"
})/db.Hosteleria.countDocuments())*100
```

```
> (db.Hosteleria.countDocuments({ cuisine: "Spanish" })/db.Hosteleria.countDocuments())*100
< 2.5119287038132416
```

El resultado obtenido es de 2.52% del total.

- c. Mostrar los nombres de todos los restaurantes de cocina portuguesa (valor "Portuguese") que se encuentren en el barrio de "Queens".

Para sacar los restaurantes con valor "Portuguese" y en el barrio "Queens" usamos `find()` y sobre el segundo valor hay que aplicarle la condición `$eq` .

```
db.Hosteleria.find( { cuisine: "Portuguese" , borough:
{$eq:"Queens"} } )
```

```
> db.Hosteleria.find( { cuisine: "Portuguese" , borough: {$eq:"Queens"} } )
< {
    _id: ObjectId('67136cea8e831985a9fe7c1b'),
    address: {
      building: '222-05',
      coord: [
        -73.732315,
        40.720725
      ],
      street: 'Jamaica Avenue',
      zipcode: '11428'
    },
    borough: 'Queens',
    cuisine: 'Portuguese',
    grades: [
      {
        date: 2014-09-17T00:00:00.000Z,
        grade: 'A',
```

Para solo mostrar los nombre añadimos otras llaves indicando el valor que solo queremos mostrar y `_id: 0` para ignorar el resto.

```
db.Hosteleria.find(
  { cuisine: "Portuguese", borough: { $eq: "Queens" } },
  { name: 1, _id: 0 }
)
```

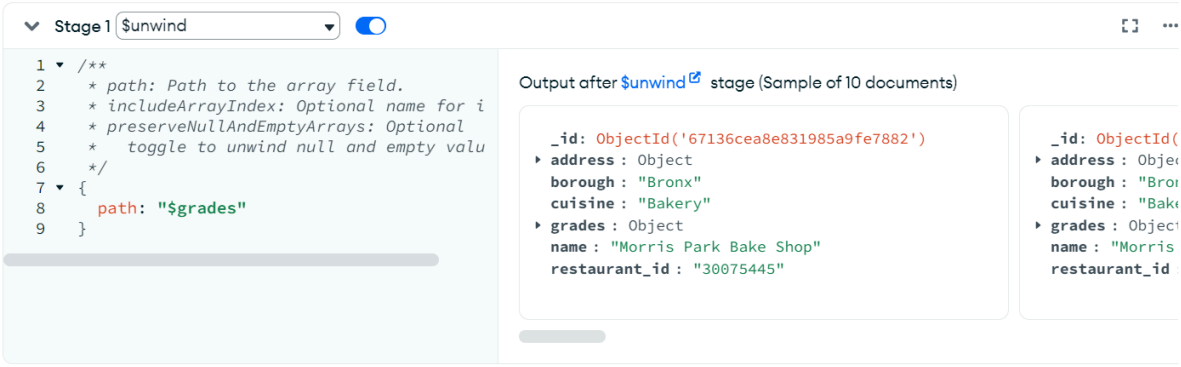
```
> db.Hosteleria.find(
    { cuisine: "Portuguese", borough: { $eq: "Queens" } },
    { name: 1, _id: 0 }
  )
< {
    name: 'Mateus Restaurant'
  }
  {
    name: 'A. Churrasqueira Restaurant'
  }
  {
    name: 'O Lavrador Restaurant'
  }
  {
    name: 'South Jamaica Portuguese Sporting Club'
  }
```

- d. Mostrar el nombre y la dirección de los restaurantes que tengan una valoración mayor de 90 puntos.

Para mostrar el nombre y la dirección de los restaurantes con `score` > 90 hay que hacer una agregación.

Para hacer una agregación vamos al campo de 'Agregations' y vamos creando nuevos 'Stages' hasta llegar al resultado deseado.

Lo primero será usar `$unwind` para desenrollar los `$grades`.

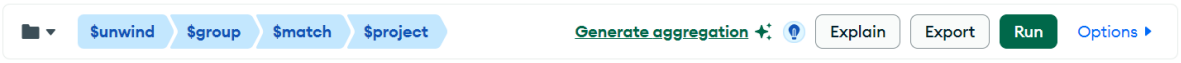


Luego agrupamos por nombre y suma con `$group`.

Luego filtramos las sumas mayores a 90 con `$match` .

Y por último usamos `$project` para mostrar solo el nombre y la dirección.

Una vez hemos terminado, ejecutamos la sentencia.



Y tras la ejecución, salen los resultados de la agregación.

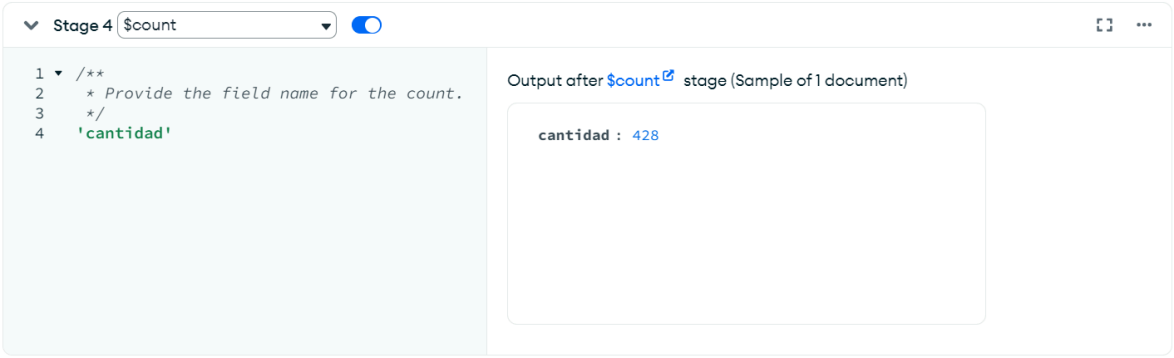

```
db.getCollection('Hosteleria').aggregate(
  [
    { $unwind: { path: '$grades' } },
    {
      $group: {
        _id: '$_id',
        totalScore: { $sum: '$grades.score' },
        name: { $first: '$name' },
        address: { $first: '$address' }
      }
    },
    { $match: { totalScore: { $gt: 90 } } },
    { $project: { _id: 0, name: 1, address: 1 } }
  ],
  { maxTimeMS: 60000, allowDiskUse: true }
);
```

- e. ¿Cuántos restaurantes de cocina española tienen una puntuación menor a 50 puntos?

Para esto hacemos otra agregación. Hacemos el primer paso y en el segundo cogemos el valor de 'cuisine' en vez de 'address'.

Luego en el `$match` necesitamos las 'score' menores a 50 y que 'cuisine' sea 'Spanish'.

Por último hacemos un `$count` para contar todos los resultados.



Ejecutamos la agregación.


```
db.getCollection('Hosteleria').aggregate(
  [
    { $unwind: { path: '$grades' } },
```

```
    {
      $group: {
        _id: '$_id',
        totalScore: { $sum: '$grades.score' },
        name: { $first: '$name' },
        cuisine: { $first: '$cuisine' }
      }
    },
    {
      $match: {
        totalScore: { $lt: 50 },
        cuisine: { $eq: 'Spanish' }
      }
    },
    { $count: 'cantidad' }
  ],
  { maxTimeMS: 60000, allowDiskUse: true }
);
```

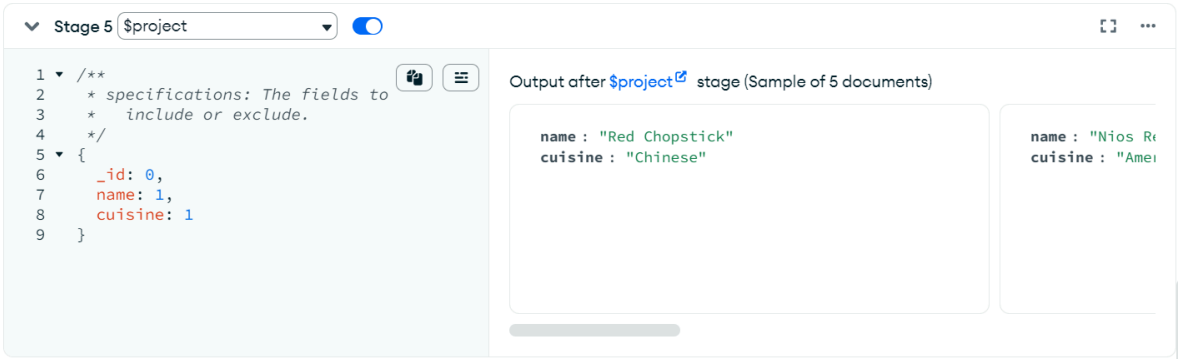
- f. Mostrar el nombre y el tipo de cocina de los cinco restaurantes con mayor puntuación.

Para hacer esta agregación dejamos las 2 etapas de la anterior y empezamos a agregar nuevas etapas acordes al enunciado.

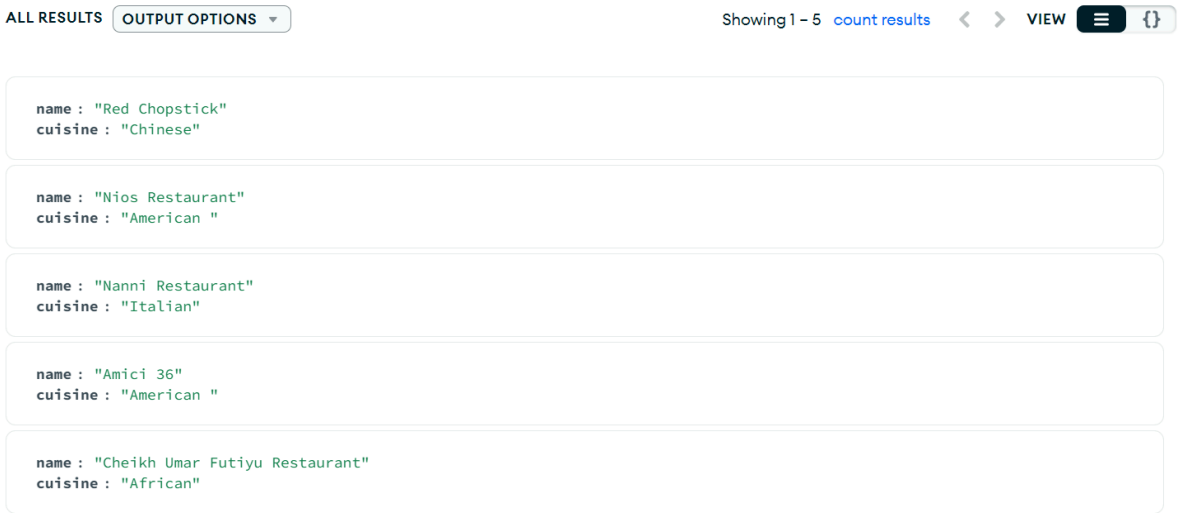
Ordenamos los resultados con `$sort` en orden del mayor al menor.

Luego usamos `$limit` para sacar los 5 primeros.

Y por último hacemos `$project` para mostrar los resultados.




```
db.getCollection('Hosteleria').aggregate(
  [
    { $unwind: { path: '$grades' } },
    {
      $group: {
        _id: '$_id',
        totalScore: { $sum: '$grades.score' },
        name: { $first: '$name' },
        cuisine: { $first: '$cuisine' }
      }
    },
    { $sort: { totalScore: -1 } },
    { $limit: 5 },
    { $project: { _id: 0, name: 1, cuisine: 1 } }
  ],
  { maxTimeMS: 60000, allowDiskUse: true }
);
```



- g. Mostrar el nombre y el tipo de cocina de los cinco restaurantes con menor puntuación

Simplemente cambiamos el `$sort` para que sea del menor al mayor.

```
db.getCollection('Hosteleria').aggregate(
  [
    { $unwind: { path: '$grades' } },
```

```
    {
      $group: {
        _id: '$_id',
        totalScore: { $sum: '$grades.score' },
        name: { $first: '$name' },
        cuisine: { $first: '$cuisine' }
      }
    },
    { $sort: { totalScore: 1 } },
    { $limit: 5 },
    { $project: { _id: 0, name: 1, cuisine: 1 } }
  ],
  { maxTimeMS: 60000, allowDiskUse: true }
);
```