Práctica - Bases de datos no relacionales - ™ MongoDB

Para empezar

1) **¿Cuántas colecciones tiene la base de datos?**
   La base de datos cuenta con una sola colección: **restaurantes.json**

2) **¿Cuántos documentos hay en cada colección? ¿Cuánto pesa cada colección?**
   La colección de restaurantes cuenta con **25359** documentos y tiene un peso de **25.4k**.

3) **¿Cuántos índices en cada colección? ¿Cuánto espacio ocupan los índices de cada colección?**
   Para la colección de restaurantes hay un índice **_id** y pesa **4.1kb**

4) **Trayendo un documento con pretty()**
   Para esta colección no hay cambios en usar el pretty o sin pretty, se notará cambios en colecciones más complejas.

```
{_id: ObjectId("5eb3d668b31de5d588f4294f"),
  direccion:
   { edificio: '625',
     coord: [ -73.990494, 40.7569545 ],
     calle: '8 Avenue',
     codigo_postal: '10018' },
  barrio: 'Manhattan',
  tipo_cocina: 'American',
  grados:
   [ { date: 2014-06-09T00:00:00.000Z, grado: 'A', puntaje: 12 },
     { date: 2014-01-10T00:00:00.000Z, grado: 'A', puntaje: 9 },
     { date: 2012-12-07T00:00:00.000Z, grado: 'A', puntaje: 4 },
     { date: 2011-12-13T00:00:00.000Z, grado: 'A', puntaje: 9 },
     { date: 2011-09-09T00:00:00.000Z, grado: 'A', puntaje: 13 } ],
  nombre: 'Cafe Metro',
  restaurante_id: '40363298'}
```

5) **Listando campos al nivel raíz, ignorando los documentos anidados**

```
db.restaurantes.find({}, {"direccion": 0, "grados":0})
```

Ejercicio 1: SQL

1. Devolver restaurante_id, nombre, barrio y tipo_cocina pero excluyendo _id para un documento (el primero).

```
db.restaurantes.findOne({}, {
      "restaurante_id": 1,
      "nombre": 1,
      "barrio":1,
      "tipo_cocina":1,
      "_id": 0
})
```

2. Devolver restaurante_id, nombre, barrio y tipo_cocina para los primeros 3 restaurantes que contengan 'Bake' en alguna parte de su nombre.

```
db.restaurantes.find({"nombre": /Bake/i},{
      "restaurante_id": 1,
      "nombre": 1,
      "barrio":1,
      "tipo_cocina":1,
      "_id": 0
}).limit(3)
```

3. Contar los restaurantes de comida (tipo_cocina) china (*Chinese*) o tailandesa (*Thai*) del barrio (barrio) Bronx.

   Opción 1: - resultado: 325

```
db.restaurantes.countDocuments({
      $or:[
         {"tipo_cocina": "Chinese"},
         {"tipo_cocina": "Thai"}
      ],
      "barrio": "Bronx"
})
```

Opción 2: - resultado: 325

```
db.restaurantes.countDocuments({
      "tipo_cocina": {
            $in:["Chinese", "Thai"]
      },
      "barrio": "Bronx"
})
```

Ejercicio 2: NoSQL

1. Traer 3 restaurantes que hayan recibido al menos una calificación de grado 'A' con puntaje mayor a 20. Una misma calificación debe cumplir con ambas condiciones simultáneamente; investigar el operador

```
db.restaurantes.find({
      "grados": {
            $elemMatch: {
                  "grado": "A",
                  "puntaje": {
                        $gte: 20
                  }
            }
      }
}).limit(3)
```

2. ¿A cuántos documentos les faltan las coordenadas geográficas? En otras palabras, revisar si el tamaño de direccion.coord es 0 y contar.

```
db.restaurantes.countDocuments({
      "direccion.coord":{
            $size: 0
      }
})
```

3. Devolver nombre, barrio, tipo_cocina y grados para los primeros 3 restaurantes; de cada documento **solo la última calificación**

```
db.restaurantes.find({}, {
      "nombre": 1,
      "barrio": 1,
      "tipo_cocina": 1,
      "grados": {
            $slice: -1
      }
}).limit(3)
```