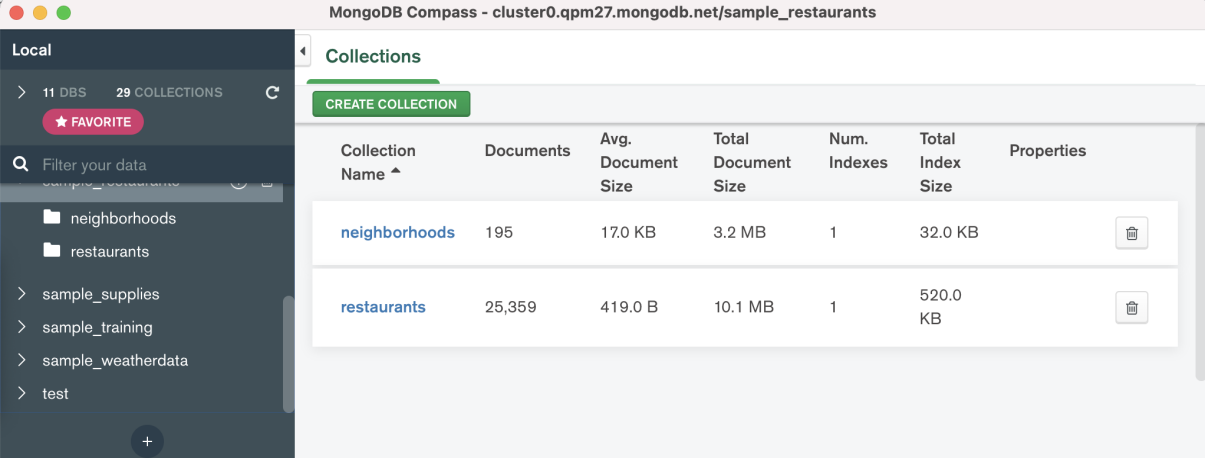


## Preguntas

- 1.¿Cuántas colecciones tiene la base de datos?.
- 1.29.
- 2.¿Cuántos documentos hay en cada colección?. ¿Cuánto pesa cada colección?.
- 2.Listo los de 1 colección.



MongoDB Compass - cluster0.qpm27.mongodb.net/sample\_restaurants

Local

11 DBS 29 COLLECTIONS

★ FAVORITE

Filter your data



sample\_restaurants

- neighborhoods
- restaurants
- sample\_supplies
- sample\_training
- sample\_weatherdata
- test

+

Collections

CREATE COLLECTION

Collection Name ^	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
neighborhoods	195	17.0 KB	3.2 MB	1	32.0 KB	
restaurants	25,359	419.0 B	10.1 MB	1	520.0 KB	

## Comandos:

show dbs

show collections (de la base actual donde apunta)

```
> show dbs
< sample_airbnb      54.7 MB
   sample_analytics  9.61 MB
   sample_geospatial 1.31 MB
   sample_mflix      43.8 MB
   sample_restaurants 6.43 MB
   sample_supplies    1.12 MB
   sample_training    48.2 MB
   sample_weatherdata 2.85 MB
   test              41 kB
   admin              336 kB
   local              2.3 GB
```

3. ¿Cuántos índices tiene cada colección? ¿Cuánto espacio ocupan los índices de cada colección?.

3.

```
db.restaurants.getIndexes()
```

```
Response: [ { v: 2, key: { _id: 1 }, name: '_id_' } ]
```

4. Traer un documento de ejemplo de cada colección. `db.collection.find(...).pretty()` nos da un formato más legible.

```
4. db.restaurants.findOne({})
```

5. Para cada colección, listar los campos a nivel raíz (ignorar campos dentro de documentos anidados) y sus tipos de datos.

5. Pongo uno de ejemplo de

```
find({}, {_id:1,address:1,borough:1,cuisine:1,grades:1,name:1,restaurant_id:1}).
```

```
{ _id: ObjectId("5eb3d668b31de5d588f4293d"),  
  address:  
    { building: '759',  
      coord: [ -73.9925306, 40.7309346 ],  
      street: 'Broadway',  
      zipcode: '10003' },  
  borough: 'Manhattan',  
  cuisine: 'Delicatessen',  
  grades:  
    [ { date: 2014-01-21T00:00:00.000Z, grade: 'A', score: 12 },  
      { date: 2013-01-04T00:00:00.000Z, grade: 'A', score: 11 },  
      { date: 2012-06-07T00:00:00.000Z, grade: 'A', score: 6 },  
      { date: 2012-01-17T00:00:00.000Z, grade: 'A', score: 8 } ],  
  name: 'Bully\'S Deli',  
  restaurant_id: '40361708' }
```

## Ejercicio SQL 1



### Ejercicio 1: SQL

Usando Mongo Shell. Colección **restaurantes** se requiere:

1. Devolver **restaurant\_id**, **nombre**, **barrio** y **tipo\_cocina** pero excluyendo **\_id** para un documento (el primero).
2. Devolver **restaurant\_id**, **nombre**, **barrio** y **tipo\_cocina** para los primeros 3 restaurantes que contengan 'Bake' en alguna parte de su nombre.
3. Contar los restaurantes de comida (**tipo\_cocina**) china (*Chinese*) o tailandesa (*Thai*) del barrio (**barrio**) Bronx. Consultar [or versus in](#).

```
1.db.restaurants.find({}, {restaurant_id:1,name:1,borough:1,cuisine:1})
2.db.restaurants.find({name:{$regex:/Bake/}}, {restaurant_id:1,name:1,borough:1,cuisine:1})
3.db.restaurants.find({cuisine:{$in:["Chinese","Thai"]},borough:"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 [elemMatch](#).
2. ¿A cuántos documentos les faltan las coordenadas geográficas? En otras palabras, revisar si el tamaño de **direccion.coord** es 0 y contar.
3. Devolver **nombre**, **barrio**, **tipo\_cocina** y **grados** para los primeros 3 restaurantes; de cada documento **solo la última calificación**. Ver el operador [slice](#).

1.

`db.restaurants.find({'$and':[{"grades.grade":"'A'"}, {"grades.score":{"$gt":20}}]})` - Este trae con que algún grade sea A y haya un puntaje mayor a 20

`db.restaurants.find({'grades':{'elemMatch':{'grade':"A",score:{$gt:20}}}})`

2.`db.restaurants.count({"address.coord":{"$size":0}})`

3.`db.restaurants.find({}, {name:1,borough:1,cuisine:1,grades:{$slice:-1}}).limit(3)`