

# Práctica

Para resolver los siguientes ejercicios puede resultar útil este machete: [MongoDB Cheat Sheet](#).

Vamos a usar la base de datos **sample\_restaurants** que se encuentra en el cluster del curso. Conectarse a la URI **mongodb+srv://dh:1234@cluster0.qpm27.mongodb.net/** usando MongoDB Compass; las credenciales son de **solo lectura**. Una vez adentro, desplegar Mongo Shell y ejecutar

```
> use sample_restaurants
```

😊 Todo listo para empezar.



# Calentamiento

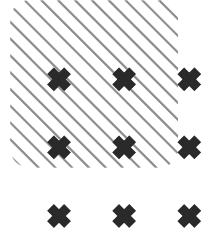
Estas preguntas pueden responderse utilizando la interfaz gráfica de Compass.

1. ¿Cuántas colecciones tiene la base de datos?
2. ¿Cuántos documentos en cada colección? ¿Cuánto pesa cada colección?
3. ¿Cuántos índices en cada colección? ¿Cuánto espacio ocupan los índices de cada colección?
4. Traer un documento de ejemplo de cada colección. `db.collection.find(...).pretty()` nos da un formato más legible.
5. Para cada colección, listar los campos a nivel raíz (ignorar campos dentro de documentos anidados) y sus tipos de datos.



Usando Mongo Shell. Colección **restaurants**:

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



# NoSQL

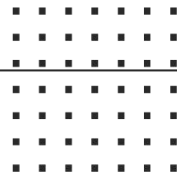
1. Traer 3 restaurantes que hayan recibido al menos una calificación de **grade 'A'** con **score** mayor a 50. Una misma calificación debe cumplir con ambas condiciones simultáneamente; investigar el operador [elemMatch](#).
2. ¿A cuántos documentos les faltan las coordenadas geográficas? En otras palabras, revisar si el tamaño de **address.coord** es 0 y contar.
3. Devolver **name**, **borough**, **cuisine** y **grades** para los primeros 3 restaurantes; de cada documento **solo la última calificación**. Ver el operador [slice](#).



# Desafiantes

1. ¿Cuál es top 3 de tipos de cocina (**cuisine**) que podemos encontrar entre los datos? *Googlear* "mongodb group by field, count it and sort it". Ver etapa [limit](#) del *pipeline* de agregación.
2. ¿Cuáles son los barrios más desarrollados gastronómicamente? Calcular el promedio (**\$avg**) de puntaje (**grades.score**) por barrio; considerando restaurantes que tengan más de tres reseñas; ordenar barrios con mejor puntaje arriba. **Ayuda:**
  - a. [match](#) es una etapa que filtra documentos según una condición, similar a **db.orders.find(<condición>)**.
  - b. Parece necesario deconstruir las listas **grades** para producir un documento por cada puntaje utilizando la etapa [unwind](#).
3. Una persona con ganas de comer está en longitud -73.93414657 y latitud 40.82302903, ¿qué opciones tiene en 500 metros a la redonda? Consultar [geospatial tutorial](#).

# Soluciones





1.

```
db.restaurants.findOne(  
  {},  
  { restaurant_id: 1, name: 1, borough: 1, cuisine: 1, _id: 0 }  
)
```

2.

```
db.restaurants.find(  
  { name: /Bake/ },  
  { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 }  
)limit(3)
```

3.

```
db.restaurants.count(  
  { cuisine: { $in: [ "Chinese", "Thai" ] }, borough: "Bronx" }  
)
```

# NoSQL

1.

```
db.restaurants.find(  
  { grades: { $elemMatch: { grade: "A", score: { $gt: 50 } } }  
}).limit(3)
```

2.

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

3.

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