Ejemplos talleres mongoDB y MapReduce

Los siguientes fragmentos de código fuente son una guía complementaria que pretende apoyar el desarrollo de los talleres mongoDB y MapReduce.

Carga de datos

Una vez descargado el archivo data.json puede importarlo escribiendo el siguiente código en la consola.

```
mongoimport --db diplomado --collection restaurantes
--drop --file data.json
```

Tenga en cuenta que este comando funciona en la consola no en la shell de mongo, adicionalmente el archivo data.json debe encontrarse en la misma carpeta donde se ejecute este comando o usted debe especificar la ruta completa del archivo. Si se encuentra utilizando un docker recuerde que debe copiar el archivo data.json al sistema de archivos del contenedor antes de ejecutar el comando, puede realizar esto utilizando la instrucción docker cp.

Para probar la correcta carga de los datos ingrese a la shell de mongoDB escribiendo en la consola la orden mongo y presionando enter.

```
mongo
```

dentro del shell de mongo escriba el siguiente comando

```
show dbs
```

deberá ver un listado con los nombres de las bases de datos, y deberá observar con el nombre "diplomado". Antes de poder ejecutar las consultas debe ubicarse dentro de la base de datos "diplomado" para ello debe utilizar el siguiente comando.

```
use diplomado
```

Dentro de la base de datos diplomado puede utilizar el comando:

```
show collections
```

Una vez ejecutado este comando deberan observar en la lista de colecciones la colección llamada "restaurantes".

mongoDB

Las siguientes 3 consultas le serán de ayuda para solucionar el taller de mongoDB.

1. La siguiente consulta trae todos los restaurantes de la base de datos con sus campos name, restaurant_id y zipcode.

```
db.restaurantes.find({}, {restaurant_id:1, name:1, "address.zipcode":1,
_id:0})
```

2. La siguiente consulta filtra aquellos restaurantes que tienen un score mayor que 80 y menor que 90.

```
db.restaurantes.find({grades:{$elemMatch: {$and: [{score:{$gt: 80}},
{score:{$lt:90}}]}}})
```

3. La siguiente consulta recupera aquellos restaurantes que tienen una latitud menor a -95.754168.

```
db.restaurantes.find({"address.coord.0":{$lt: -95.754168}})
```

4. La siguiente consulta filtra aquellos restaurantes que no son de cocina American ni Chinese ni cuyo nombre inicia por Wil.

```
db.restaurantes.find({$or: [{$and: [{"cuisine": {$not: /American*/}},
{"cuisine": {$not: /Chinese*/}}]}, {"name":/Wil*/}]})
```

MapReduce

Los siguientes ejemplos de funciones mapReduce le pueden servir de guía (Debe complementar y escribir los ejemplos para resolver el taller no copie y pegue el presente código pues no resuleve los puntos del taller,solo son ejemplos guía, adicional debido a los saltos de línea al pegarlos pueden producirse errores, escriba las funciones y pruebelas usted mismo).

1. El siguiente código MapReduce cuenta el número de restaurnates cuya latitud se encuentra entre los valores -75 y -74.

```
// map function
var mapFunction = function(){if(this.address.coord[0] > -75 &&
this.address.coord[0] < -74){emit(1,1);};};
// reduce function
var reduceFunction = function(key, value){return Array.sum(value);};
// mapreduce
db.restaurantes.mapReduce(mapFunction, reduceFunction, {out: "count"})
```

2. El siguiente código calcula la calificación media (promedio) por cada restaurante y los guarda en una coleción.

```
//map function
var mapFunction = function(){var sum = 0; for(var i=0; i <
this.grades.length; i++){sum += this.grades[i].score;}; var avg = sum /
this.grades.length; emit(this.restaurant_id, avg);};
// reduce function
var reduceFunction = function(key, value){return Array.sum(value);};
// MapReduce function
db.restaurantes.mapReduce(mapFunction, reduceFunction, {out:"avg"})
```