

1) Usando Google Maps encuentra las coordenadas terrestres de “UCM Biblioteca María Zambrano”, y completa su longitud y latitud (los valores que aparecen después de la @ en la url, pero dados la vuelta), las instrucciones de inserción siguiente:

```
db.lugares.drop()
db.lugares.insert({nombre:"Informática-UCM", tipo:"Facultad", pais:"España",
  posicion: {type:"Point", coordinates:[-3.7353797,40.450305]}})
db.lugares.insert({nombre:"Metropolitano", tipo:"metro", pais:"España",
  posicion: {type:"Point", coordinates:[-3.7202654,40.4465915]}})
db.lugares.insert({nombre:"Ciudad Universitaria", tipo:"metro", pais:"España",
  posicion: {type:"Point", coordinates:[-3.7289768,40.4435602]}})
db.lugares.insert({nombre:"Pabellón de Plata", tipo:"templo", pais:"Japón",
  posicion: {type:"Point", coordinates:[135.7982058,35.0270213]}})
db.lugares.insert({nombre:"María Zambrano", tipo:"biblioteca", pais:"España",
  posicion: {type:"Point", coordinates:[???long,???lat]}})
```

Y crear un índice tipo 2d esférico sobre el campo “posicion”

Escribir en la solución:

- Comando insert del lugar del que se han buscado las coordenadas (solo de ese)
- Instrucción para crear el índice y respuesta del shell

Solución

```
db.lugares.insert({ nombre:"María Zambrano", tipo:"biblioteca", pais:"España",
  posicion: {type:"Point", coordinates:[ -3.735358, 40.449153]}})
```

```
db.lugares.createIndex({posicion:"2dsphere"})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

2.-

Solución

```
db.lugares.find( { 'posicion' : { $near : {
  $geometry: { type:'Point',
  coordinates: [-3.7251149,40.4381963]},
  $minDistance: 1500 } } }, { nombre:1, tipo:1, _id:0});
```

3)

use pract2d

```
db.plano.drop()
```

```
for (var i=0; i<500; i++) { for (var j=0; j<=i; j++) { db.plano.insert({desc:"("+i+", "+j+")", punto:[i,j]});}}
```

Solución

Creación de índice:

```
db.plano.createIndex({punto:1},"ipunto")
db.plano.createIndex({desc:1},"idesc")
```

Comprobar tamaño: db.plano.stats()

¿Cuál de los dos índices tiene mayor tamaño? ipunto

¿Por qué? (una frase debe bastar): porque es un índice multikey, donde tiene que crear una entrada para cada valor del array.

4) Solución:

```
query: db.plano.find({"punto.0":{"$gt:480}}).count()
```

5)

Solución

Utiliza el índice (Sí/no): No

¿Cuál? Ninguno, porque el único que puede tener acceso es a ipunto, pero al ser un array crea una entrada por cada componente y no tiene forma de distinguir la posición 0.

6) Solución

```
db.plano.find( {punto: {
```

```
$geoWithin:
```

```
{ $polygon: [ [ 0 , 0 ], [ 3 , 0 ], [ 3 , 3 ], [0,3] ] } }
```

```
} )
```

o

```
db.plano.find( {punto: {
```

```
$geoWithin: { $box: [ [ 0 , 0 ],
```

```
[ 3 , 3 ] ] } }
```

```
} )
```

7) La consulta anterior no utiliza ningún índice. Escribe en la solución qué índice crearías para mejorar la eficiencia.

Solución

```
db.plano.createIndex({ punto: "2d" })
```

8) ¿Se puede crear el índice? ¿Por qué?

Solución:

No, porque una de contiene coordenadas con formato incorrecto

9)

Solución

Sí, por ejemplo: `db.plano.createIndex({punto:1,desc:1})`

10) Queremos encontrar la distancia aproximada a la que se encuentra la biblioteca María Zambrano del Museo de América, con un error de más/menos 100 metros. Escribir un fragmento de código que nos indique esa distancia

Solución

```
db.lugares.aggregate([
  {
    $geoNear: {
      near: { type: "Point", coordinates: [-3.7251149,40.4381963] },
      distanceField: "dist.calculated",
      query: { "nombre": "María Zambrano" },
      spherical: true
    }
  }
]).pretty()

distancia = 100;
found = false;
while (!found) {
  d = db.lugares.find( { 'posicion' : { $near : {
    $geometry: { type:'Point',
    coordinates: [-3.7251149,40.4381963]},
    $maxDistance: distancia } } }, { nombre:1,_id:0}).toArray();
  // ver si está a esta distancia
  for (i=0; i<d.length && !found;i++)
    if (d[i]['nombre']=='María Zambrano')
      found=true;
  // si no está, probar 100 metros más allá
  if (!found)
    distancia +=100;
}
print("Distancia: "+distancia);
```

```
db.lugares.aggregate([
```

```
{
  $geoNear: {
    near: { type: "Point", coordinates: [ -3.724155 , 40.438323 ] },
    distanceField: "lejos",
    query: { nombre: "María Zambrano" },
    spherical: true
  }
},
{
  $project: {
    lejos:1,
    _id:0
  }
}
})
```