Práctica 7 – NoSQL

Preparación:
0. Iniciar el servidor (Desde un terminal: mongod -dbpath datos)
1. Bajar del campus el fichero tweet.json
2. Desde un terminal de linux teclear
mongoimport --db test --collection tweet --file tweet.json

Importará 17834 tweets

1) Crear un índice textual sobre el campo text de tweet, y una consulta para conocer el número de palabras que incluyen la palabra USA. Deben salir 93

Solución
db.tweet.createIndex({text:"text"})
db.tweet.count({$text:{$search:"USA"}})

2) Para buscar textos que incluyan alguna palabra de una lista, simplemente se separan las palabras por un espacio al hacer la búsqueda textual. Probar a contar el número de tweets que incluyan o bien "USA" o bien "WIN". Deben salir 637 tweets.

Solución
db.tweet.count({$text:{$search:"USA WIN"}})

3) Para excluir una palabra de la búsqueda, se precede esta palabra por el símbolo -.
Contar el número de tweets que incluyen WIN pero no USA. Deben salir 544

Solución
db.tweet.count({$text:{$search:"-USA WIN"}})

4) Una opción interesante de la búsqueda textual es la posibilidad de que se nos muestre el resultado por orden de parecido. Se puede utilizar tanto en la proyección "generando" una nueva clave con el score o parecido, como en la opción sort, en ambos casos con la sintaxis:

```
{ <projectedFieldName>: { $meta: "textScore" } }
```

Copia en la solución una consulta que muestre el tweet que contiene la palabra *lie* con máximo score.

La salida debe ser de la forma:

{ "text" : ". @HillaryClinton Even Lies About Lying About Her Lies https://t.co/ltqzv8qQV2", "score" : 1.1666666666666665 }

Nota: ver https://docs.mongodb.com/manual/reference/operator/projection/meta/#proj._S_meta para más detalles

Solución:

```
db.tweet.find({$text:{$search:"lie"}},{text:1,_id:0,score:{$meta:"
textScore"}}).sort({score:{$meta:"textScore"}}).limit(1)
```

5) Los tweets que son negativos para Clinton se indentifican por hlabel:-1. Los negativos para Trump, por tlabel:-1.

Escribir dos consultas para determiner

a) La proporción de tweets negativos para Clinton que utilizan la palabra 'lie' (para calcular la proporción dividir el total de tweets con hlabel:-1 y palabra lie entre el total de tweets con hlabel:-1)
b) La proporción de tweets negativos para Trump que utilizan la palabra 'lie'

Solución

```
db.tweet.find({$text:{$search:"lie"}, hlabel:-1}).count()/db.tweet.find({hlabel:-1}).count()
0.012271259418729818
db.tweet.find({$text:{$search:"lie"}, tlabel:-1}).count()/db.tweet.find({tlabel:-1}).count()
0.006979062811565304
```

6) Queremos conocer el usuario (user.screen) que más veces ha utilizado la palabra *lie*, ordenado por el número de veces que ha utilizado la palabra

Salida esperada:

```
{ "_id" : "AppSame", "count" : 15 }
{ "_id" : "USAneedsTRUMP", "count" : 10 }
{ "_id" : "CarmineZozzora", "count" : 9 }
{ "_id" : "patrioticpepe", "count" : 9 }
…
```

Solución

```
db.tweet.aggregate([ {$match:{$text:{$search:"lie"}}},
                     {$group:{_id:"$user.screen", count:{$sum:1}}},
                     {$sort:{count:-1}} ])
```

7) [2 puntos] Queremos hacer consultas de la forma:

```
db.tweet.find({"tlabel":1},
              {"created_at":1,"tlabel":1,_id:0}).sort({created_at:-1})
```

de forma que se consulte la cantidad más pequeña de documentos posible (a costa de aumentar las claves de índice consultadas). Escribir

a) Escribir el tiempo en milisegundos, el total de claves y el total de documentos examinados por esta consulta sin índices.
b) Una instrucción para crear un índice que permita lograr el objetivo.
c) Escribir el número de claves y documentos que se consultan, antes de crear el índice y después de hacerlo.

Solución

a)

```
"executionTimeMillis" : 63,
"totalKeysExamined" : 0,
"totalDocsExamined" : 17834,
```

b) db.tweet.createIndex({tlabel:1,created_at:1})

c)

```
"executionTimeMillis" : 17,
"totalKeysExamined" : 5381,
"totalDocsExamined" : 0,
```

10) Consideramos el siguiente código a ejecutar en la shell de mongo:

```
db.prueba.drop()
db.prueba.insert({a:1,b:1})
db.prueba.insert({a:2,b:1})
db.prueba.insert({a:3,b:1})
db.prueba.createIndex({a:1},'ia')
db.prueba.createIndex({b:1},'ib')
```

y la consulta

```
db.prueba.find({a:2,b:1})
```

comprobamos, consternados, que esta consulta utiliza, el índice 'ia', pero nosotros, por alguna oscura razón, queremos que use el índice 'ib'.

Queremos forzar a que la consulta utilize el índice 'ib', en lugar del 'ia' pero sin modificar la consulta, ni usar sort, ni crear ni borrar índices, ni nada similar.

Hint: buscar en la documentación de mongo, es algo que no hemos visto en clase y que permite forzar el uso de un índice

Solución

```
db.prueba.find({a:2,b:1}).hint('ib')
```