

## Práctica 5: Agregación (II)

0. Iniciar el servidor (mongod -dbpath datos)
1. Bajar del campus los ficheros minitweet.json y miniuser.json
2. Desde un terminal de linux teclear  
mongoimport --db test --collection user --file user.json  
mongoimport --db test --collection tweet --file tweet.json

### Nota:

- No olvidar poner los nombres
- Copiar en solucion.txt solo consultas.

En esta práctica vamos a utilizar las operaciones de agregación [\\$group](#) y [\\$sort](#).  
En particular, queremos consultas para:

- 1) Usando la colección tweet, consideramos para aquellos tweets que tengan el campo opinion=1 y correspondan al usuario (user.screen) “patrioticpepe”. La salida de la consulta será de la forma

```
{ .... "h" : -17, "t" : 29, "count" : 70 }
```

h: la suma de la clave hlabel en todos los tweets considerados

t: la suma de la clave tlabel en todos los tweets considerados

count: número de tweets considerados

Solución.

```
db.tweet.aggregate([
  {
    • "$match":{
      • "user.screen":"patrioticpepe",
      • "opinion":1
    }
  },
  {
    • "$group":{
      • "_id":null,
      • "h":{
        • "$sum":"$hlabel"
      },
      • "t":{
        • "$sum":"$tlabel"
      },
      • "count":{
        • "$sum":1
      }
    }
  }
])
```

1)

2) Nombre de la persona más mencionada (mentions en tweet). Debe ser HillaryClinton con 4832 menciones.

Solución

```
db.tweet.aggregate([
  {"$unwind":"$mentions" },
  {"$group":{"_id":"$mentions", "count":{"$sum":1}} },
  {"$sort":{"count":-1}},
  {"$limit":1}
])
```

3) En user, queremos saber el número de seguidores por intervalos. En particular nos interesan los intervalos

[0,1000), [1000,5000), [5000,10000), [10000,100000), [100000, infinito)

Nos deben salir los valores 5,6,5,56,69, respectivamente

Solución

```
db.user.aggregate( [ { $bucket: { groupBy: "$followers", boundaries: [ 0, 1000, 5000, 10000, 100000 ], default: "Muchos" } } ] )
```

4) En user, queremos solo ver el \_id y el screen\_name, ambos formando un array para la clave user. Debe ser algo así:

```
{ "user" : [ "265554862", "JosephMorgan" ] }
{ "user" : [ "721446222549147648", "FacMagnaAmerica" ] }
{ "user" : [ "1075681177", "2ALAW" ] }
```

Solución:

```
db.tweet.aggregate([{$project:{user:["$user._id","$user.screen"],_id:0}}])
```

5) Para cada usuario (documento de la colección user) queremos obtener

- Su screen\_name
- El número de seguidores
- El total de tweets que tiene en la colección tweet

q

Una posible salida:

Una posible salida:

```
{ "screen_name" : "JosephMorgan", "followers" : 2248939, "cuantos" : 1 }
{ "screen_name" : "FacMagnaAmerica", "followers" : 12877, "cuantos" : 55 }
{ "screen_name" : "2ALAW", "followers" : 35360, "cuantos" : 86 }
```

.....

Solución:

```
db.user.aggregate([{$lookup: {
    from:"tweet",
    localField:"screen_name",
    foreignField:"user.screen",
    as:"tweets"
  }
},
{$project: {
    screen_name:1,
    followers:1,
    cuantos:{$size:"$tweets"},
    _id:0
  }
}
])
```