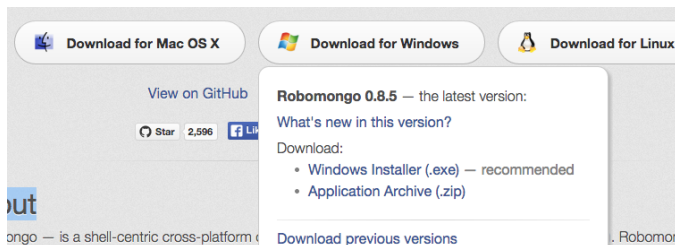


# SQL – NOSQL

## 1. INSTALAÇÃO E CONFIGURAÇÃO

Vamos começar por fazer download do Robomongo para Windows em <http://www.robomongo.org/>



Escolher versão **Application Archive(.zip)**

De seguida iniciamos o Robomongo através do executável que se encontra dentro do arquivo.

Vamos seleccionar **Connection** para abrir o menu **MongoDB Connections** onde vamos seleccionar **Create** de forma a criar uma ligação ao servidor remoto MongoDB.

Em **Connection Settings** introduzimos o endereço do servidor:

Address - bdad.fe.up.pt

Port - 27017

e os parâmetros de autenticação em **Authentication**:

Database - twitter

User Name - bdad

Password - 2017

De seguida clicar em **Test**. Em caso de sucesso podemos guardar a ligação.

No menu **MongoDB Connections** seleccionar a ligação que acabamos de criar e clicar em **Connect**.

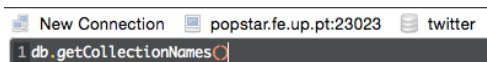
Vamos então seleccionar o Explorer dentro do menu **View**. Deverá aparecer a base de dados twitter no **explorer** do lado esquerdo. Clicar em twitter com o botão direito do rato e abrir **Open Shell**.

A partir deste momento vamos escrever instruções de pesquisa na linha de comando. Podemos executar as instruções clicando em play ou então através do atalho ctrl+enter.

## 2. TUTORIAL

Vamos obter o nome das coleções existentes na base de dados utilizando o método `getCollectionNames`.

Escreve a seguinte instrução na linha de comandos e executa (play ou ctrl+enter):



```
1 db.getCollectionNames()
```

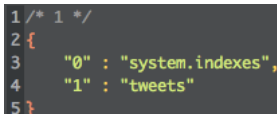
Por omissão os resultados são exibidos no modo de árvore:

Key	Value	Type
▼ (1)	{ 2 fields }	Object
0	system.indexes	String
1	tweets	String

Os modos de exibição de resultados podem ser seleccionados no canto superior da linha de comandos (árvore, tabela e texto).



De forma a facilitar a leitura devemos seleccionar sempre o modo de leitura de texto (ícone mais à direita):



```
1 /* 1 */
2 {
3   "0" : "system.indexes",
4   "1" : "tweets"
5 }
```

De forma a retornar um cursor para todos os tweets na coleção utilizamos a função `find()`.



```
1 db.tweets.find()
```

O objetivo é obter o número total de documentos (tweets) na coleção. Para tal faz-se:

```
1 db.tweets.find().count()
2
3
```

0.266 sec.

```
1 51428
2
```

Estamos interessados na collection tweets, desta forma sempre que criarmos instruções teremos que selecionar essa collection escrevendo db.tweets. Vamos então ver como é um tweet:

```
1 db.tweets.findOne()
```

0.029 sec.

```
1 /* 1 */
2 {
3   "_id" : ObjectId("5567669ae9e57318a849ea0c"),
4   "favorited" : false,
5   "entities" : {
6     "user_mentions" : [],
7     "hashtags" : [],
8     "urls" : []
9   },
10  "contributors" : null,
11  "truncated" : false,
12  "text" : "eu preciso de terminar de fazer a minha tabela, está muito",
13  "created_at" : "Thu Sep 02 18:11:23 +0000 2010",
14  "retweeted" : false,
15  "coordinates" : null,
16  "source" : "web",
17  "in_reply_to_status_id" : null,
18  "in_reply_to_screen_name" : null,
19  "user" : {
20    "follow_request_sent" : null,
21    "profile_use_background_image" : true,
22    "id" : 5.35078e+07,
23    "verified" : false,
24    "profile_sidebar_fill_color" : "768575",
25    "profile_text_color" : "25b8c2",
26    "followers_count" : 102,
27    "protected" : false,
28    "location" : "",
29    "profile_background_color" : "081114",
30    "listed_count" : 0,
31    "utc_offset" : -10800,
32    "statuses_count" : 3504,
33    "description" : "só os loucos sabem (:)",
34    "friends_count" : 73,
35    "profile_link_color" : "eb55b6",
36    "profile_image_url" : "http://a2.twimg.com/profile_images/103641",
37    "notifications" : null,
38    "show_all_inline_media" : false,
39    "geo_enabled" : false,
```

```

40     "profile_background_image_url" : "http://a1.twimg.com/profile_bac
41     "name" : "Beatriz Helena Cunha",
42     "lang" : "en",
43     "profile_background_tile" : true,
44     "favorites_count" : 1,
45     "screen_name" : "Bia_cunha1",
46     "url" : "http://http://www.orkut.com.br/Main#Profile?uid=14332958",
47     "created_at" : "Fri Jul 03 21:44:05 +0000 2009",
48     "contributors_enabled" : false,
49     "time_zone" : "Brasilia",
50     "profile_sidebar_border_color" : "1c9dbd",
51     "following" : null
52   },
53   "place" : null,
54   "retweet_count" : null,
55   "geo" : null,
56   "id" : NumberLong(22819396900),
57   "in_reply_to_user_id" : null
58 }

```

Como podemos verificar, um tweet possui uma estrutura relativamente complexa. A lista de atributos da raiz do documento é composta pelos seguintes atributos:

*favorited; retweet\_count; in\_reply\_to\_user\_id; contributors; truncated; text; created\_at; retweeted; coordinates; entities; in\_reply\_to\_status\_id; in\_reply\_to\_screen\_name; source; place; \_id; geo; id; user*

Podemos efetuar pesquisas condicionadas pelos valores dos atributos de cada documento sejam strings, inteiros ou arrays.

Para definir igualdade (=) utilizamos a estrutura chave valor dos documentos {atributo:valor}.

Vamos testar:

```
1 db.tweets.findOne({'source':'web'})
```

As projeções permitem-nos retornar apenas os atributos e os respetivos valores que nos interessam.

Neste caso será:

```

1 db.tweets.findOne({'source':'web'},{'source':1})
0.139 sec.
1 /* 1 */
2 {
3   "_id" : ObjectId("5567669ae9e57318a849ea0c"),
4   "source" : "web"
5 }

```

O documento tweet possui alguns atributos com documentos 'nested', isto é, atributos cujo valor são outros documentos, como é o caso dos atributos entities, user e location. Neste caso sempre que quisermos pesquisar por valores de atributos dos documentos 'nested' utilizamos a notação '.' como por exemplo, {'user.name':'John'}.

Vamos experimentar:

```
1 db.tweets.findOne({'user.name': 'John'})
```

O MongoDB oferece-nos uma série de operadores de comparação:

<code>\$gt</code>	Matches values that are greater than a specified value.
<code>\$gte</code>	Matches values that are greater than or equal to a specified value.
<code>\$lt</code>	Matches values that are less than a specified value.
<code>\$lte</code>	Matches values that are less than or equal to a specified value.
<code>\$ne</code>	Matches all values that are not equal to a specified value.
<code>\$in</code>	Matches any of the values specified in an array.
<code>\$nin</code>	Matches none of the values specified in an array.

A sintaxe do MongoDB obriga-nos a utilizar os operadores como valores de um atributo entre chavetas, como por exemplo `{ 'idade': { '$gt': 18 } }`

Vamos encontrar um tweet com hashtags (entities.hashtags é um array) o que corresponde a pesquisar por um tweet cujo array hashtags não seja vazio.

```
1 db.tweets.findOne( { 'entities.hashtags': { '$ne': [ ] } } )
0.154 sec.
1 /* 1 */
2 {
3   "_id" : ObjectId("5567669ae9e57318a849ea0f"),
4   "favorited" : false,
5   "entities" : {
6     "user_mentions" : [],
7     "hashtags" : [
8       {
9         "indices" : [
10          17,
11          33
12        ],
13        "text" : "BoardwalkEmpire"
14      }
15    ],
16    "urls" : []
17  },
```

E se quisermos saber o número total de tweets com hashtags?

```
1 db.tweets.find( { 'entities.hashtags': { '$ne': [ ] } }).count()
0.078 sec.
1 6558
```

Agora que conhecemos a estrutura de uma hashtag podemos pesquisar por uma hashtag com valor específico:

```
1 db.tweets.findOne( { 'entities.hashtags.text': 'swagg' } )
```

O MongoDB tem funcionalidades de pesquisa de texto em atributos cujo valor é uma string através da função '\$regex'. Vamos então encontrar um tweet que fale sobre futebol:

```
1 db.tweets.findOne({'text':{'$regex':'futebol'}})
2
0.035 sec.
1 /* 1 */
2 {
3   "_id" : ObjectId("5567669ae9e57318a849ec5f"),
4   "favorited" : false,
5   "entities" : {
6     "user_mentions" : [],
7     "hashtags" : [],
8     "urls" : []
9   },
10  "contributors" : null,
11  "truncated" : false,
12  "text" : "Time do Flamengo tá pior que meu futebol",
13  "created_at" : "Thu Sep 02 18:12:37 +0000 2010",
14  "retweeted" : false,
15  "coordinates" : null,
```

As funções de ordenação são muito simples e podemos selecionar qual o atributo pelo qual queremos ordenar os resultados utilizando a função .sort({'atributo':1}) para ordenar por ordem crescente ou -1 para decrescente.

No entanto por questões de memória temos sempre que limitar o número de resultados aos quais iremos aplicar a ordenação. (Nota: podemos aplicar à coleção toda mas para tal deveríamos guardar o cursor numa variável local).

Vamos então ver qual o utilizador com maior número de *followers* nos primeiros 100 *tweets* aleatórios:

```
1 db.tweets.find().limit(100).sort({'user.followers_count':-1})
2
tweets 0.346 sec.
29 "user" : {
30   "follow_request_sent" : null,
31   "profile_use_background_image" : true,
32   "id" : 5.76887e+06,
33   "verified" : true,
34   "profile_sidebar_fill_color" : "efe6ce",
35   "profile_text_color" : "996633",
36   "followers_count" : 854199,
```

O operador *distinct* é um *wrapper* para as funções de agregação do MongoDB como group by, e permite-nos obter todos os valores distintos de um dado atributo.

Vamos então ver os nomes únicos de utilizadores que existem na collection tweets:

```
1 db.tweets.distinct('user.name')  
  
0.996 sec.  
  
1 /* 1 */  
2 {  
3   "0" : "Beatriz Helena Cunha",  
4   "1" : "Medscape Pathology",  
5   "2" : "Travis Siebrass",  
6   "3" : "Emma Smith",  
7   "4" : "Lia afriani chaniago",  
8   "5" : "ピヨ山ピヨ彦",  
9   "6" : "kaitlinn spears",  
10  "7" : "ペリーさん",  
11  "8" : "Nitin",  
12  "9" : "marisa alfiani",  
13  "10" : "brittany =]",  
14  "11" : "Ariadna",  
15  "12" : "Catherine Mullane",  
16  "13" : "aundrea",
```

### 3. EXERCÍCIOS

Escreve instruções de pesquisa que respondam às seguintes questões:

1. Quantos tweets têm como source o twitter versão 'web'?
2. Quantos tweets têm a hashtag 'javascript'?
3. Quais são as hashtags mais populares? (utilize a função `aggregate()` )
4. Quantos tweets têm utilizadores com mais de 100 followers?
5. Ordena 100 tweets aleatórios pelo número de amigos dos utilizadores por ordem decrescente.
6. Quantos tweets falam do Cristiano Ronaldo?
7. Quantos tweets têm utilizadores com timezone 'Lisbon' ?
8. Encontra timezones distintas na coleção.
9. Qual é a timezone com mais tweets? (utilize a função `aggregate()` )
10. Um tweet com menções a outros utilizadores .
11. Quantos tweets têm menções a outros utilizadores?
12. Quantos tweets têm 3 menções a outros utilizadores?
13. Quantos tweets têm 3 menções a outros utilizadores e 2 hashtags?
14. Existe um utilizador com mais de 7 tweets na coleção. Qual o seu screen\_name? (utilize a função `aggregate()` )
15. Qual é o tweet com maior número de hashtags? (utilize a função `aggregate()` )