Universidad Internacional de La Rioja

Facultad de Ingeniería y Tecnología

Máster Universitario en Análisis y Visualización de Datos Masivos / Visual Analitics & Big Data

Laboratorio: uno de MongoDB

Actividad de estudio presentado por:	Juan David Escobar Escobar
Tipo de trabajo:	Laboratorio
Modalidad:	Individual
Director/a:	Marlon Cardenas Bonett
Fecha:	Enero 2022

Índice de contenidos

Table of Contents

Índice de tablas

1. Carga de datos

1.1. Creación base de datos “miselania”

```
1  use miscelanea
2  db.movie.insert({"name":"tutorials point"}) #Para que aparezca
   en el listado show db.
```

1.2. Creación de colecciones books y companies

```
3  db.createCollection("books")
4  db.createCollection("companies")
```

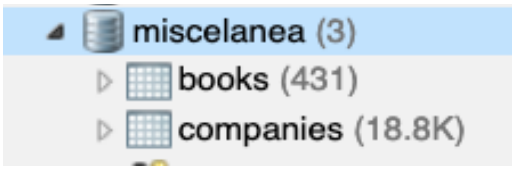
1.3. Creación de colecciones books y companies

```
5  db.createCollection("books")
6  db.createCollection("companies")
```

1.4. Importar datos desde archivos en formato json

```
7  mongoimport --verbose --db miscelanea --collection books --
   file
   "/Users/juandavidescobarescobar/Documents/Unir/Materias/BD Big
   Data/Actividad 2/visa01_lab (1)/act-2-books.json"
8  mongoimport --verbose --db miscelanea --collection companies
   --file
   "/Users/juandavidescobarescobar/Documents/Unir/Materias/BD Big
   Data/Actividad 2/visa01_lab (1)/act-2-companies.json"
```

Ilustración 1. BD miselanias y colecciones books y companies.



2. Exploración de colecciones

2.1. Identifica todas las distintas categorías (categories) de la colección books

```
9  db.books.find({}, {"categories":1, "_id":0}).pretty()
```

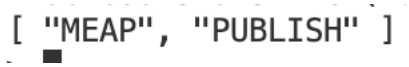
Ilustración 2. Resultado pregunta 2-a.

```
{ "categories" : [ "Software Engineering" ] }
{ "categories" : [ "Web Development" ] }
{ "categories" : [ "Internet" ] }
```

2.2. Identifica los distintos estados (status) de la colección books.

```
10 db.books.distinct("status").pretty()
```

Ilustración 3. Resultado pregunta 2-b.



2.3. Describe brevemente qué arroja la siguiente consulta

```
11 db.getCollection('books').find({longDescription: {$gte: "A",
   $lt: "B"}}, {title: 1, longDescription: 1}).
```

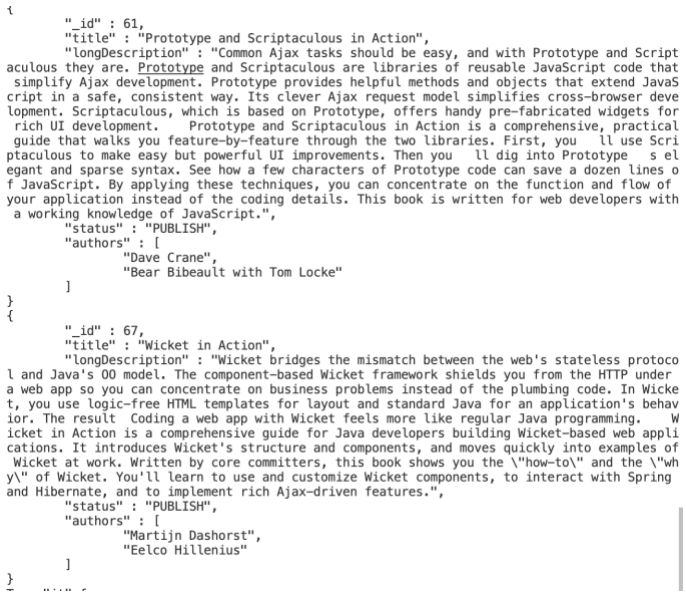
La consulta retorna 36 documentos que cumplen las condiciones de que el texto del campo o key longDescription sea mayor o igual a la letra “A” y que sea menor a la letra “B”, solo se muestran los valores para el title y longDescription.

2.4. Utiliza la condición de la consulta anterior para recuperar aquellos libros que posean exactamente 2 autores y que estén publicados. Muestra solo los campos: title, longDescription, status y authors

Se identifican 88 documentos que cumplen estas condiciones, a continuación la consulta:

```
12 db.getCollection('books').find({status: "PUBLISH", authors:
   {$size: 2}}, {title: 1, longDescription:1, status:1,
   authors:1})
```

Ilustración 4. Resultado pregunta 2-d.



2.5. Describe brevemente qué ocurre si a la consulta del punto anterior, le añades al final la siguiente instrucción: .toArray()

El método find() retorna un cursor para imprimir el resultado en documentos, objetos o json, al invocar la función .toArray(), los documentos se almacenan en un objeto tipo Array[], separados por comas.

2.6. Qué ocurre si también le añades lo siguiente

```
13 db.getCollection('books').find({status: "PUBLISH", authors:
   {$size: 2}}, {title: 1, longDescription:1, status:1,
   authors:1}).toArray().forEach(function(valor, indice,
   array){print("Titulo: " + valor.title + "Autho 1: " +
   valor.authors[0] + " Author 2: " + valor.authors[1] + "
   Registro No. " + indice);})
```

Al ejecutar la consulta, estamos haciendo uso del método forEach() de la clase Array de Javascript, dicho método recibe como argumento una función anónima de Javascript, la cual a su vez se le definen 4 argumentos (valor, índice y array), el array es el objeto o colección que va a recorrer el forEach, el valor será una variable que cambiara en cada iteración de la lista y el índice indicara la posición iterada de cada elemento del Array, por ultimo se utiliza la función print() de JS, para imprimir los valores de una manera legible y elegante.

El organizar los datos en una estructura tipo Array nos da orden, también nos brinda la posibilidad de manipularlos de una mejor manera, al recorrerlos datos mediante una función de iteración y utilizar una función anónima, nos da la posibilidad de crear el código de Javascript que nosotros queramos definir, por ejemplo, podríamos crear columnas calculadas, o generar resultados a partir de condiciones, etc.

3. Consulta la colección 1 (books)

3.1. ¿Cuál es el tamaño de la colección (en bytes)?

Resultado 517474 bytes, 0.5175 MB

```
14 db.books.dataSize()
```

Ilustración 5. Resultado pregunta 3-a.

```
> use books
switched to db books
> db.book.dataSize()
517474
```

3.2. ¿Cuántos libros tiene la colección?

Resultado 431

```
15 db.books.find({}).count()
```

Ilustración 6. Resultado pregunta 3-b

```
> db.book.find({}).pretty(), db.book.find({}).count()
431
```

3.3. ¿Cuántos libros tienen 200 o más páginas?

Resultado 264

```
16 db.books.find( { pageCount: { $gte: 200 } }).count()
```

Ilustración 7. Resultado pregunta 3-c

```
> db.book.find( { pageCount: { $gte: 200 } }).count()
264
> db.book.find( { pageCount: { $gte: 200 } }, {_id: 1, title:1, pageCount:1} )
{ "_id" : 1, "title" : "Unlocking Android", "pageCount" : 416 }
{ "_id" : 2, "title" : "Android in Action, Second Edition", "pageCount" : 592 }
```

3.4. ¿Cuántos libros tienen entre 300 y 600 páginas?

Resultado 215

```
17 db.books.find({pageCount: {$gte: 300, $lte: 600} }, {_id: 1,
   title:1, pageCount:1}).count()
```

Ilustración 8. Resultado pregunta 3-d

```
> db.book.find({pageCount: {$gte: 300, $lte: 600} }, {_id: 1, title:1, pageCount:1} ).count()
215
> db.book.find({pageCount: {$gte: 300, $lte: 600} }, {_id: 1, title:1, pageCount:1} ).pretty()
{ "_id" : 1, "title" : "Unlocking Android", "pageCount" : 416 }
{
        "_id" : 2,
        "title" : "Android in Action, Second Edition",
        "pageCount" : 592
}
{ "_id" : 4, "title" : "Flex 3 in Action", "pageCount" : 576 }
```

3.5. ¿Cuántos libros tienen 0 páginas y cuántos no?

Respuesta 215

```
18 db.books.find({pageCount: {$gte: 300, $lte: 600} }, {_id: 1,
   title:1, pageCount:1}).count()
```

Ilustración 9. Resultado pregunta 3-d

```
> db.book.find({pageCount: {$gte: 300, $lte: 600} }, {_id: 1, title:1, pageCount:1} ).count()
215
> db.book.find({pageCount: {$gte: 300, $lte: 600} }, {_id: 1, title:1, pageCount:1} ).pretty()
{ "_id" : 1, "title" : "Unlocking Android", "pageCount" : 416 }
{
        "_id" : 2,
        "title" : "Android in Action, Second Edition",
        "pageCount" : 592
}
{ "_id" : 4, "title" : "Flex 3 in Action", "pageCount" : 576 }
```

3.6. ¿ Cuántos libros tienen 0 páginas y cuántos no?

Libros con cero paginas 166, libros con mas de cero paginas 265

```
19 db.books.find({pageCount:{$eq:0} }, {_id: 1, title:1,
   pageCount:1}).count() // 23. PREGUNTA 3-e libros tienen 0
   páginas y cuántos.
20 db.books.find({pageCount:{$gt:0} }).count() // 24. PREGUNTA
   3-e Libros que tienen 0 páginas.
```

Ilustración 10. Resultado pregunta 3-e

```
> db.book.find({pageCount:{$eq:0} }, {_id: 1, title:1, pageCount:1}).count()
166
> db.book.find({pageCount:{$gt:0} }).count()
265
```

3.7. ¿Cuántos libros han sido publicados y cuántos no?

Libros publicados 363, libros no publicados 68

```
21 db.books.find({status: {$ne:"PUBLISH"}}, {_id: 1, title:1,
   status:1}).count() // 25. PREGUNTA 3-f Libros no publicados.
22 db.books.find({status: {$eq:"PUBLISH"}}, {_id: 1, title:1,
   status:1}).count() // 26. PREGUNTA 3-f Libros publicados.
```

Ilustración 11. Resultado pregunta 3-f

```
> db.book.find({status: {$ne:"PUBLISH"}}, {_id: 1, title:1, status:1}).count()
68
> db.book.find({status: {$eq:"PUBLISH"}}, {_id: 1, title:1, status:1}).count()
363
```

4. Consulta la colección 2 (companies)

4.1. ¿Cuál es el tamaño de la colección (en bytes)?

Resultado 72236994 bytes, 72.23 MB

```
23 db.companies.dataSize()
```

Ilustración 12. Resultado pregunta 4-a.

```
> db.companies.dataSize()
72236994
```

4.2. ¿Cuántas compañías tiene la colección?

Resultado 18801

```
24 db.companies.find({}).count()
```

Ilustración 13. Resultado pregunta 4-b.

```
> db.companies.find({}).count()
18801
```

4.3. ¿Cuántas compañías se fundaron en los años 1996, 1997, 2001 y 2005 respectivamente?

- Compañías fundadas en 1996 = 216
- Compañías fundadas en 1997 = 200
- Compañías fundadas en 2001 = 464
- Compañías fundadas en 2005 = 961

```
25 db.companies.aggregate([
26     {$match: { $or:[{founded_year:1996},
27                     {founded_year:1997},
28                     {founded_year:2001},
29                     {founded_year:2005}
30                     ]
31               }
32     },
33     {$group : {_id:"$founded_year", count:{$sum:1}}},
34     {$sort: {_id:1}},
35     {
36       $project: {
37             "founded_year": "$_id",
38             count: 1,
39             "_id": 0
40       }
41     }
42 ])
```

Ilustración 14. Resultado pregunta 4-c.

```
{ "count" : 216, "founded_year" : 1996 }
{ "count" : 200, "founded_year" : 1997 }
{ "count" : 464, "founded_year" : 2001 }
{ "count" : 961, "founded_year" : 2005 }
```

4.4. Lista las compañías que se dedican a “web” o “mobile” y recupera: nombre, descripción, número de empleados, email, año, mes y día de su fundación.

En total se obtienen 4805 compañías de dicadas a las categorías “web” o “mobile”, se imprime una muestra de la consulta obtenida:

```
43 db.companies.find(
44       {$or: [{category_code:"web"},{category_code:"mobile"}]},
45       {
46             name:1,
47             description:1,
48             number_of_employees:1,
49             email_address:1,
50             founded_year:1,
51             founded_month:1,
52             founded_day:1
53       }
54 ).pretty().count()
```

Ilustración 15. Resultado pregunta 4-d.


```
{
        "_id" : ObjectId("52cdef7c4bab8bd675297d8a"),
        "name" : "Wetpaint",
        "number_of_employees" : 47,
        "founded_year" : 2005,
        "founded_month" : 10,
        "founded_day" : 17,
        "email_address" : "info@wetpaint.com",
        "description" : "Technology Platform Company"
}
{
        "_id" : ObjectId("52cdef7c4bab8bd675297d90"),
        "name" : "Postini",
        "number_of_employees" : null,
        "founded_year" : 1999,
        "founded_month" : 6,
        "founded_day" : 2,
        "email_address" : "",
        "description" : null
}
```


4.5. Lista las compañías que se dedican a videojuegos y muéstralas en orden descendente según el año en que fueron fundadas

En total se obtienen 4805 compañías de dicadas a las categorías “web” o “mobile”, se imprime una muestra de la consulta obtenida:

```
55 db.companies.find(
56                     {category_code: "games_video"},
57                     {_id:1,      name:1,      category_code:1,
   founded_year:1}
58                 ).sort({ "founded_year": -1}).pretty().count()
```

Ilustración 16. Resultado pregunta 4-e.


```
{
        "_id" : ObjectId("52cdef7e4bab8bd67529b1ef"),
        "name" : "Fliggo",
        "category_code" : "games_video",
        "founded_year" : 2012
}
{
        "_id" : ObjectId("52cdef7c4bab8bd67529831a"),
        "name" : "Social Gaming Network",
        "category_code" : "games_video",
        "founded_year" : 2011
}
```


4.6. ¿Cuántas compañías tienen 600 o más empleados?

En total se obtienen 303, se imprime una muestra de la consulta obtenida:

```
59 db.companies.find({number_of_employees: {$gte:600} },
60                   {_id:1,                                  name:1,
   number_of_employees:1}).pretty().count()
```

Ilustración 17. Resultado pregunta 4-f.

```
{
        "_id" : ObjectId("52cdef7c4bab8bd675297d8b"),
        "name" : "AdventNet",
        "number_of_employees" : 600
}
{
        "_id" : ObjectId("52cdef7c4bab8bd675297d8c"),
        "name" : "Zoho",
        "number_of_employees" : 1600
}
```

4.7. Recupera el nombre, la URL, el usuario de Twitter y el número de empleados de las compañías fundadas entre los años 2001 y 2005 incluidos, que cuenten con 500 o más empleados y que se dediquen a los videojuegos o a la música. ¿Alguna empresa se dedica a videojuegos y a la música a la vez?

No se encuentran documentos disponibles para empresas que se dedican a la música y los videojuegos a la vez. En total se obtienen 2 resultados, se imprime una muestra de la consulta obtenida:

```
61 db.companies.find({
62                     $and:[
63                             {founded_year: {$gte: 2001, $lte:
   2005}},
64                             {number_of_employees: {$gte: 500}}
65                         ],
66                     $or:[
67                             {category_code: "games_video" },
68                             {category_code: "music" }
69                         ]
70                  },
71                  {
72                     _id:1,
73                     name:1,
74                     category_code:1,
75                     homepage_url:1,
76                     twitter_username:1,
77                     number_of_employees:1
78                 }).pretty().count() // 33. PREGUNTA 4-g
```

```
79 db.companies.find({$and:[
80                         {category_code: "games_video" },
81                         {category_code: "music" }
82                   ]},
83                   {
84                                          _id:1,
```

```
85                                                  name:1,
86                                                  category_code:1,
87                                                  homepage_url:1,
88                                                  twitter_username:1,
89
   number_of_employees:1
90                                          }).pretty().count() // 34.
   PREGUNTA 4-g
```

Ilustración 18. Resultado pregunta 4-g.

```
{
        "_id" : ObjectId("52cdef7e4bab8bd67529a8b4"),
        "name" : "GREE",
        "homepage_url" : "http://www.gree-corp.com",
        "twitter_username" : "gree_corp",
        "category_code" : "games_video",
        "number_of_employees" : 700
}
{
        "_id" : ObjectId("52cdef7f4bab8bd67529c178"),
        "name" : "Bigpoint",
        "homepage_url" : "http://www.bigpoint.com",
        "twitter_username" : "Bigpoint",
        "category_code" : "games_video",
        "number_of_employees" : 500
}
```

4.8. Lista las empresas que cuentan con única y exclusivamente 2 oficinas en la ciudad de San Francisco.

En total se obtienen 2 resultados, se imprime una muestra de la consulta obtenida:

```
91 db.companies.find( {offices: {$size: 2},
92                     $and:[
93                                {"offices.0.city":          "San
   Francisco"},
94                                {"offices.1.city":          "San
   Francisco"}
95                          ]},{ _id:1, offices:1}).pretty()
```

Ilustración 19. Resultado pregunta 4-h.


```
...                                                                    }).pretty()
{
        "_id" : ObjectId("52cdef7d4bab8bd6752990cc"),
        "offices" : [
                {
                        "description" : "Corporate Office",
                        "address1" : "2 Harrison Street",
                        "address2" : "2nd Floor",
                        "zip_code" : "94105",
                        "city" : "San Francisco",
                        "state_code" : "CA",
                        "country_code" : "USA",
                        "latitude" : 37.7895248,
                        "longitude" : -122.3886569
                },
                {
                        "description" : "Data Center",
                        "address1" : "360 Spear Street",
                        "address2" : "2nd Floor",
                        "zip_code" : "94105",
                        "city" : "San Francisco",
                        "state_code" : "CA",
                        "country_code" : "USA",
                        "latitude" : 37.78932,
                        "longitude" : -122.390058
                }
        ]
}
{
        "_id" : ObjectId("52cdef7d4bab8bd675299c64"),
        "offices" : [
                {
                        "description" : "Mailing address",
                        "address1" : "PO BOX 77464",
                        "address2" : "",
                        "zip_code" : "CA94107-0464",
                        "city" : "San Francisco",
                        "state_code" : "CA",
                        "country_code" : "USA",
                        "latitude" : null,
                        "longitude" : null
                },
                {
                        "description" : "Offices",
                        "address1" : "PariSoMa co-working space",
                        "address2" : "1436 Howard Street",
                        "zip_code" : "94103",
                        "city" : "San Francisco",
                        "state_code" : "CA",
                        "country_code" : "USA",
                        "latitude" : null,
                        "longitude" : null
                }
        ]
} _
```


4.9. Lista el nombre, el mes y día de adquisición de las empresas de videojuegos que hayan sido adquiridas en el año 2007 por un precio igual o superior a los 10 millones de dólares y que tengan oficinas en la ciudad de Culver City.

En total se obtienen 1 resultados, se imprime una muestra de la consulta obtenida:

```
96 db.getCollection('companies').find({ $and:[
97                     {category_code: "games_video"},
98                     {"acquisition.acquired_year": 2007},
99                     {"acquisition.price_amount":        {$gte:
   10000000} }
100                     ]
101             },
102             {
103                 name:1,
104                 "acquisition.acquired_month":1,
105                 "acquisition.acquired_day":1,
106                 "acquisition.price_currency_code":1,
107                 "acquisition.acquiring_company.name":1,
108                 // acquisition:1,
109                 offices: 1
```

```
110
111             }).toArray().forEach(function(valor,       indice,
   array){
112
113                 array_offices = valor.offices;
114                 current_name = valor.name;
115                 current_acquired_month                          =
   valor.acquisition.acquired_month.toString()
116                 current_acquired_day                            =
   valor.acquisition.acquired_day.toString()
117                 current_acquisition_price_currency_code
   =valor.acquisition.price_currency_code.toString()
118                 current_acquiring_company                       =
   valor.acquisition.acquiring_company.name.toString()
119
120                 is_culver_city = false;
121                 var       city_to_filter       =       "Culver
   City".toUpperCase().trim();
122
123
124                 for  (var  i  =  0;  i  <  array_offices.length;
   ++i) {
125
126                     current_city                                =
   array_offices[i].city.toUpperCase().trim()
127
128                     if( city_to_filter === current_city ){
129
130
131                         str_result    =    "nombre:    "    +
   current_name + ", mes: " + current_acquired_month  + ", dia: "
   + current_acquired_day + ", ciudad: " + current_city + "\n";
132
133                         print(str_result)
134
135                     }
136                 }
137
138                 // current_index = indice + 1
139                 // array_len = array.length
140
141                 // if( current_index == array_len ){
142                 //     print(current_index, array_len)
143                 //     print(str_result)
144                 // }
145             });
```

Ilustración 20. Resultado pregunta 4-i.

```
nombre: Flektor, mes: 5, dia: 30, ciudad: CULVER CITY
```

El ejercicio exploración de mongo me ha permitido identificar las principales funciones de la base de datos de MongoDB, publico en el siguiente repositorio de GitHub los documentos asociados a la actividad.