

Practica mongo

Ejecuciones en Shell para importar el archivo que representa la base de datos de colecciones

Linux:

```
mongoimport --db practicaMongo --collection cPractica <"/gitRepo/uc3m-BigData/mongoPractica/data.json"
```

Windows

```
mongoimport --db practicaMongo --collection cPractica <"D:\DEV\mongo3.4.23\practica3\uc3m-bigData\mongoPractica\data.json"
```

Start mongo

```
D:\DEV\mongo3.4.23\bin\mongod.exe --dbpath D:\DEV\mongo3.4.23\data
```

```
D:\mongo3.4.23\bin\mongod.exe --dbpath D:\mongo3.4.23\data
```

```
2019-12-17T16:57:43.910+0100    connected to: mongod://localhost/
```

```
2019-12-17T16:57:44.915+0100    4230 document(s) imported successfully. 0 document(s) failed to import.
```

Posterior se abre un Shell de mongo

```
>use practicaMongo
```

```
> show collections
```

```
cPractica
```

```
mytest
```

```
>
```

```
> db.cPractica.count()
```

```
4230
```

```
>
```

Solucion de ejercicios de mongo

1. ExM: clientes con dos apellidos con contrato fuera de España (aportar nombre, apellido y país).

Descripción de la solución: Se ejecuta la función find sobre la colección que contiene la data. Dentro de la función hay condiciones de búsqueda sobre el apellido en el cual se restringe a un array de 2 posiciones, y sobre el país en el cual se restringe que sea diferente de España. Posterior se validan los atributos de proyección: nombre, apellido y país.

Código:

```
db.cPractica.find(
  { "Client.Surname": { $size:2 } , "contract.country": { $ne: "España" } },
  {"_id":0,"Client.Name":1,"Client.Surname":1,"contract.country":1}
)
```

Pruebas:

```
{ "Client" : { "Name" : "Victor Enrique", "Surname" : [ "Apaza", "Okuma" ] }, "contract" : { "country" : "Jordan" } }
{ "Client" : { "Name" : "Maria del Altagracia", "Surname" : [ "Fuentes", "Castellanos" ] }, "contract" : { "country" : "Saint Lucia" } }
{ "Client" : { "Name" : "Jaime", "Surname" : [ "Aliaga", "Oporto" ] }, "contract" : { "country" : "Macedonia" } }
{ "Client" : { "Name" : "M. Carmela", "Surname" : [ "Maranon", "Solis" ] }, "contract" : { "country" : "British Indian Ocean Territory" } }
{ "Client" : { "Name" : "Ignacio", "Surname" : [ "Guillen", "Perez" ] }, "contract" : { "country" : "Japan" } }
{ "Client" : { "Name" : "Atila", "Surname" : [ "Ruiz", "Gomez" ] }, "contract" : { "country" : "United Arab Emirates" } }
{ "Client" : { "Name" : "Amaya", "Surname" : [ "Calero", "Garcia" ] }, "contract" : { "country" : "Lebanon" } }
{ "Client" : { "Name" : "Monica", "Surname" : [ "Hernandez", "Malpartida" ] }, "contract" : { "country" : "British Indian Ocean Territory" } }
{ "Client" : { "Name" : "Daniel David", "Surname" : [ "Diaz", "Illescas" ] }, "contract" : { "country" : "Uruguay" } }
{ "Client" : { "Name" : "Maria de la Fortuna", "Surname" : [ "Martinez", "Lopez" ] }, "contract" : { "country" : "Gibraltar" } }
{ "Client" : { "Name" : "Iban", "Surname" : [ "Olivas", "Mendieta" ] }, "contract" : { "country" : "Suriname" } }
{ "Client" : { "Name" : "Matias", "Surname" : [ "Prince", "Solis" ] }, "contract" : { "country" : "Malawi" } }
{ "Client" : { "Name" : "Chavela", "Surname" : [ "Tuesta", "Martinez" ] }, "contract" : { "country" : "Brazil" } }
{ "Client" : { "Name" : "Daisy", "Surname" : [ "Supo", "Escajadillo" ] }, "contract" : { "country" : "Philippines" } }
{ "Client" : { "Name" : "Armando", "Surname" : [ "Ventocilla", "Copa" ] }, "contract" : { "country" : "Sri Lanka" } }
{ "Client" : { "Name" : "Alberto", "Surname" : [ "Camayo", "Prudencio" ] }, "contract" : { "country" : "Liechtenstein" } }
{ "Client" : { "Name" : "Manuel Oscar", "Surname" : [ "Huapaya", "Alatrisme-Gomez" ] }, "contract" : { "country" : "Pitcairn" } }
{ "Client" : { "Name" : "Victor", "Surname" : [ "Ruiz", "Bocanegra" ] }, "contract" : { "country" : "Panama" } }
{ "Client" : { "Name" : "Silvia", "Surname" : [ "Montesinos", "Escobar" ] }, "contract" : { "country" : "Belgium" } }
{ "Client" : { "Name" : "Leandro", "Surname" : [ "Ortiz", "Ferro" ] }, "contract" : { "country" : "Zimbabwe" } }
Type "it" for more
>
```

2. User-Ganga: facturación de clientes que no han consumido nada en un mes (DNI y total).

Descripción de la solución: Se crea una variable "currentDateMinusAMonth" que guarde la fecha actual menos un mes, y posterior se hace la búsqueda sobre charge date donde la fecha sea menor a currentDateMinusAMonth

Código:

```
var currentDateMinusAMonth = new Date();
currentDateMinusAMonth.setMonth(currentDateMinusAMonth.getMonth() - 1);

db.cPractica.find(
  {"charge date": {$lte: currentDateMinusAMonth.toString()}}, {"Client.DNI":1,"TOTAL":1}
);
```

Pruebas:

```
{ "_id" : "SI23361/12/1/2016", "TOTAL" : 16.65, "Client" : { "DNI" : "96607380C" } }
{ "_id" : "JO30887/12/1/2016", "TOTAL" : 13.8, "Client" : { "DNI" : "61475943W" } }
{ "_id" : "FM23440/03/1/2016", "TOTAL" : 21.6, "Client" : { "DNI" : "84001081K" } }
{ "_id" : "SK15987/03/1/2016", "TOTAL" : 15.7, "Client" : { "DNI" : "05131232R" } }
{ "_id" : "CN38838/10/1/2016", "TOTAL" : 15.7, "Client" : { "DNI" : "31220800W" } }
{ "_id" : "IQ38436/12/1/2016", "TOTAL" : 17.6, "Client" : { "DNI" : "66312279J" } }
{ "_id" : "CA31157/13/1/2016", "TOTAL" : 17.6, "Client" : { "DNI" : "16304184J" } }
```

```
{ "_id" : "AW16095/07/1/2016", "TOTAL" : 19.5, "Client" : { "DNI" : "90458603V" } }
{ "_id" : "MH32588/01/1/2016", "TOTAL" : 16.65, "Client" : { "DNI" : "34983718M" } }
{ "_id" : "TR16117/14/1/2016", "TOTAL" : 16.65, "Client" : { "DNI" : "71292980H" } }
{ "_id" : "LC39936/10/1/2016", "TOTAL" : 17.6, "Client" : { "DNI" : "68534833P" } }
{ "_id" : "MK23180/11/1/2016", "TOTAL" : 16.65, "Client" : { "DNI" : "83383980X" } }
{ "_id" : "IO17305/12/1/2016", "TOTAL" : 16.65, "Client" : { "DNI" : "86057826Z" } }
{ "_id" : "PT23432/01/1/2016", "TOTAL" : 15.7, "Client" : { "DNI" : "93566516D" } }
{ "_id" : "MS32374/09/1/2016", "TOTAL" : 21.6, "Client" : { "DNI" : "59427180X" } }
{ "_id" : "NI17222/07/1/2016", "TOTAL" : 15.7, "Client" : { "DNI" : "58280466F" } }
{ "_id" : "ES38816/09/1/2016", "TOTAL" : 13.8, "Client" : { "DNI" : "53887554A" } }
{ "_id" : "AQ25477/08/1/2016", "TOTAL" : 17.6, "Client" : { "DNI" : "01581449S" } }
{ "_id" : "RO31046/04/1/2016", "TOTAL" : 13.8, "Client" : { "DNI" : "54641232V" } }
{ "_id" : "MT17374/05/1/2016", "TOTAL" : 16.65, "Client" : { "DNI" : "58359966L" } }
Type "it" for more
```

3. Secuela parece: DNI y títulos de películas que hayan visto los clientes que han visualizado películas cuyos títulos contienen guarismos pero no al principio ni final del título.

Descripción de la solución: Se llama la función aggregate haciendo unwind para obtener Movies y poder filtrarlo con el regex sobre el titulo

Regex:

```
Indica el inicio con al menos una letra del alfabeto: ^([a-z]|[A-Z])+
Indica al menos un numero posterior:[0-9]+
Indica al final con al menos una letra del alfabeto: [a-z]|[A-Z])+$
```

posterior se hace la proyección del DNI y los títulos

```
db.cPractica.aggregate([
  {
    $unwind: '$Movies'
  },
  {
    $match: {"Movies.Title": new RegExp('^([a-z]|[A-Z])+[0-9]+([a-z]|[A-Z])+$')}
  },
  {
    $project: {
      _id: 0,
      DNI: '$Client.DNI',
      Movies: '$Movies.Title'
    }
  }
]);
```

Pruebas:

```
{ "DNI" : "52362765E", "Movies" : "Thr3e" }
{ "DNI" : "34440819K", "Movies" : "Se7en" }
{ "DNI" : "69880671R", "Movies" : "Thr3e" }
{ "DNI" : "91899912P", "Movies" : "Thr3e" }
{ "DNI" : "94901575X", "Movies" : "Thr3e" }
{ "DNI" : "79227726V", "Movies" : "Thr3e" }
```

```
{ "DNI" : "61293350Y", "Movies" : "Se7en" }
{ "DNI" : "91346198L", "Movies" : "Se7en" }
{ "DNI" : "16724613W", "Movies" : "Se7en" }
{ "DNI" : "72543050S", "Movies" : "Se7en" }
{ "DNI" : "24403094W", "Movies" : "Thr3e" }
>
```

4. ZappersLand: (resolver con GROUP) para cada país, porcentaje de visualizaciones que han sido interrumpidas antes del 5% de visualización, y porcentaje de visualizaciones completas (para éstas, se considera como completa toda visualización que alcance el 95%, ya que el resto pueden ser rótulos de crédito).

Descripción de la solución: Se facilita parte del código para superar ciertos obstáculos iniciales en el manejo de JavaScript. Observa que la consulta afecta a películas y series, y por eso han de tenerse en cuenta ambos atributos (primero uno y luego otro). Además, los atributos Movies y Series son opcionales, y por ello hay que asegurarse de que existan antes de utilizarlos (de otro modo, dará error al procesar el primer documento que no contenga uno de ellos). Por último, observa los contratiempos que puede generar utilizar atributos cuyos nombres contengan espacios en lugar de CamelCasedVariableNames. En este caso, Javascript admite la notación de punto y la notación corchete, y es gracias a una combinación de ambas que conseguimos superar el obstáculo.

Código:

```
db.cPractica.group( {
  key:{"contract.country":1},
  cond: {"contract.country":{"$exists:1}},
  initial:{card:0, low:0, high:0},
  reduce: function(doc,res) {
    if (typeof doc.Series !== "undefined" && doc.Series !== null)
    {
      for (epi of doc.Series){
        pct=epi["Viewing PCT"];
        if(pct < 5){
          res.low++;
        }
        if(pct > 95){
          res.high++;
        }
        res.card++;
      }
    }
  };
},
finalize: function(res) {
  res.media_low=(res.low/res.card)*100 + ' %';
  res.media_high=(res.high/res.card)*100 + ' %';
}
});
```

Pruebas: No copio todos los valores que me regresa la consulta, solo unos pocos:

```
{
  "contract.country" : "Saudi Arabia",
  "card" : 91,
  "low" : 15,
  "high" : 70,
  "media_low" : "16.483516483516482 %",
  "media_high" : "76.92307692307693 %"
},
{
  "contract.country" : "Solomon Islands",
  "card" : 117,
  "low" : 15,
  "high" : 97,
  "media_low" : "12.82051282051282 %",
  "media_high" : "82.90598290598291 %"
},
{
  "contract.country" : "Madagascar",
  "card" : 61,
  "low" : 7,
  "high" : 49,
  "media_low" : "11.475409836065573 %",
  "media_high" : "80.32786885245902 %"
}
```

5. Género-Favorito: (resolver con map-reduce) para cada género cinematográfico, cantidad de visionados de películas de ese género y porcentaje medio de visionado de esas películas.

Descripción de la solución:

Map: Se valida que exista el atributo Movies y Genres, posterior se emite el valor (ej: (Action,1)) y se hace conteo para calcular los porcentajes

Reduce: Se suman las cantidades por genero (ej: (Action,1) + (Action,1) = (Action,2)

Finalize: Se calculan los porcentajes de visionado

Código:

```
var mapGen = function() {
  if (typeof this.Movies !== "undefined" && this.Movies !== null)
    for (var i = 0, len = this.Movies.length; i < len; i++) {
      var detalle = this.Movies[i].Details;
      if (typeof detalle.Genres !== "undefined" && detalle.Genres !== null)
        for (var j = 0, len2 = detalle.Genres.length; j < len2; j++) {
          emit(detalle.Genres[j],1);
          count++;
        }
    }
}
```

```

}
var redGen = function(key, cant) {
  return Array.sum(cant);
};

db.cPractica.mapReduce(mapGen, redGen,
  {
    out:"favGenres",
    scope:{count:0},
    finalize:function(key, total){
      var aux=total*100/count;
      return {occurrences:total, pct: aux};
    }
  }
);

db.favGenres.find();

```

Pruebas:

```

> db.favGenres.find();
{ "_id" : "Action", "value" : { "occurrences" : 3889, "pct" : 7.947926672252764 } }
{ "_id" : "Adventure", "value" : { "occurrences" : 3109, "pct" : 6.353845210602685 } }
{ "_id" : "Animation", "value" : { "occurrences" : 835, "pct" : 1.7064846416382253 } }
{ "_id" : "Biography", "value" : { "occurrences" : 966, "pct" : 1.9742085794281743 } }
{ "_id" : "Comedy", "value" : { "occurrences" : 6298, "pct" : 12.871185955733583 } }
{ "_id" : "Crime", "value" : { "occurrences" : 3047, "pct" : 6.22713617134332 } }
{ "_id" : "Documentary", "value" : { "occurrences" : 384, "pct" : 0.7847785657354234 } }
{ "_id" : "Drama", "value" : { "occurrences" : 8727, "pct" : 17.835319122846457 } }
{ "_id" : "Family", "value" : { "occurrences" : 1872, "pct" : 3.825795507960189 } }
{ "_id" : "Fantasy", "value" : { "occurrences" : 2023, "pct" : 4.134393329382191 } }
{ "_id" : "Film-Noir", "value" : { "occurrences" : 28, "pct" : 0.057223437084874616 } }
{ "_id" : "Game-Show", "value" : { "occurrences" : 3, "pct" : 0.006131082544807995 } }
{ "_id" : "History", "value" : { "occurrences" : 692, "pct" : 1.4142363736690442 } }
{ "_id" : "Horror", "value" : { "occurrences" : 1924, "pct" : 3.9320676054035273 } }
{ "_id" : "Music", "value" : { "occurrences" : 722, "pct" : 1.475547199117124 } }
{ "_id" : "Musical", "value" : { "occurrences" : 440, "pct" : 0.8992254399051726 } }
{ "_id" : "Mystery", "value" : { "occurrences" : 1792, "pct" : 3.6622999734319754 } }
{ "_id" : "News", "value" : { "occurrences" : 6, "pct" : 0.01226216508961599 } }
{ "_id" : "Reality-TV", "value" : { "occurrences" : 6, "pct" : 0.01226216508961599 } }
{ "_id" : "Romance", "value" : { "occurrences" : 3645, "pct" : 7.449265291941714 } }
Type "it" for more

```

- Serial-Lovers: clientes que han visto al menos una temporada completa de una serie. Se proporcionará el DNI, y un listado con las series (título de la serie y lista de temporadas completas que ha visto).

Descripción de la solución: Se hace una operación de agregación sobre la colección en la cual se hace un unwind para extraer las Series y se filtra que la temporada sea mayor a 1 (Se asume que las temporadas anteriores ya fueron vistas por el cliente y que Series.Season corresponde a la temporada actual, es decir, si Series.Season == 3 entonces ya vio 2 temporadas). El resultado se guarda en otra colección en la que posteriormente se hace un aggregate para hacer un distinct sobre el DNI y la Serie vista.

Código:

```
db.cPractica.aggregate([
  {$project:{_id:0}},
  {$unwind:"$Series"},
  {$match:{"Series.Season":{$gt:1}}},
  {$project:{"Client.DNI":1, "Series.Title":1, TemporadasCompletas : {$subtract: ["$Series.Season" , 1]} }},
  {$out:"SerialLovers"}
]);

db.SerialLovers.aggregate([{"$group": { "_id": { Client: "$Client", Series: "$Series", TemporadasCompletas:"$TemporadasCompletas"} }}}]);
```

Pruebas:

```
{ "_id" : { "Client" : { "DNI" : "15625505H" }, "Series" : { "Title" : "The Sopranos" }, "TemporadasCompletas" : 2 } }
{ "_id" : { "Client" : { "DNI" : "15625505H" }, "Series" : { "Title" : "Saturday Night Live" }, "TemporadasCompletas" : 20 } }
{ "_id" : { "Client" : { "DNI" : "45016401L" }, "Series" : { "Title" : "Numb3rs" }, "TemporadasCompletas" : 4 } }
{ "_id" : { "Client" : { "DNI" : "45016401L" }, "Series" : { "Title" : "Knight Rider" }, "TemporadasCompletas" : 1 } }
{ "_id" : { "Client" : { "DNI" : "45016401L" }, "Series" : { "Title" : "Dr Who" }, "TemporadasCompletas" : 34 } }
{ "_id" : { "Client" : { "DNI" : "17620216P" }, "Series" : { "Title" : "Monk" }, "TemporadasCompletas" : 1 } }
{ "_id" : { "Client" : { "DNI" : "17620216P" }, "Series" : { "Title" : "Mad Men" }, "TemporadasCompletas" : 3 } }
{ "_id" : { "Client" : { "DNI" : "17620216P" }, "Series" : { "Title" : "Cheers" }, "TemporadasCompletas" : 3 } }
{ "_id" : { "Client" : { "DNI" : "09803499W" }, "Series" : { "Title" : "How I Met Your Mother" }, "TemporadasCompletas" : 8 } }
{ "_id" : { "Client" : { "DNI" : "09803499W" }, "Series" : { "Title" : "Cheers" }, "TemporadasCompletas" : 1 } }
{ "_id" : { "Client" : { "DNI" : "89281978A" }, "Series" : { "Title" : "Star Trek" }, "TemporadasCompletas" : 2 } }
{ "_id" : { "Client" : { "DNI" : "89281978A" }, "Series" : { "Title" : "Perry Mason" }, "TemporadasCompletas" : 4 } }
{ "_id" : { "Client" : { "DNI" : "89281978A" }, "Series" : { "Title" : "Hill Street Blues" }, "TemporadasCompletas" : 1 } }
{ "_id" : { "Client" : { "DNI" : "32068286F" }, "Series" : { "Title" : "Poirot" }, "TemporadasCompletas" : 6 } }
{ "_id" : { "Client" : { "DNI" : "32068286F" }, "Series" : { "Title" : "Murder, She Wrote" }, "TemporadasCompletas" : 10 } }
{ "_id" : { "Client" : { "DNI" : "32068286F" }, "Series" : { "Title" : "Game of Thrones" }, "TemporadasCompletas" : 2 } }
{ "_id" : { "Client" : { "DNI" : "32068286F" }, "Series" : { "Title" : "Bonanza" }, "TemporadasCompletas" : 8 } }
{ "_id" : { "Client" : { "DNI" : "12549029E" }, "Series" : { "Title" : "Two and a Half Men" }, "TemporadasCompletas" : 5 } }
{ "_id" : { "Client" : { "DNI" : "86681439M" }, "Series" : { "Title" : "Tell Me How It Happened" }, "TemporadasCompletas" : 15 } }
{ "_id" : { "Client" : { "DNI" : "20354798M" }, "Series" : { "Title" : "Cheers" }, "TemporadasCompletas" : 10 } }
Type "it" for more
>
```

7. Keyword: “bodrio”: 10 palabras clave que están en el decil superior de palabras clave que corresponden a películas abandonadas (se considera abandonada una película que no llega al 95% de visionado; si un cliente ha visualizado varias veces la misma película, se considera abandonada si ninguna de esas veces llega al 95%). Se requiere el decil en porcentaje de abandonos (con respecto al total de visualizaciones para cada palabra clave; si un cliente visualiza varias veces la misma película computa como una visualización). De ese conjunto de palabras clave, se requiere las 10 que presenten mayor visualización media (computando sólo películas abandonadas). Proporcionar la palabra clave, porcentaje de abandonos, y porcentaje medio de visualización.

Descripción de la solución: Se hace un mapReduce para obtener las palabras más usadas sobre las películas abandonadas. En el map se hace emit de cada palabra para título de película que ha sido abandonada. En el reduce se agrupan las cantidades por palabra. Posteriormente se hace sort descendente y se obtiene el top de las 10 palabras más usadas por película abandonada.

Código:

```
var mapGenBodrio = function() {
  if (typeof this.Movies !== "undefined" && this.Movies !== null)
    for (var i = 0, len = this.Movies.length; i < len; i++) {
      if(this.Movies[i]["Viewing PCT"] < 95){
        var title = this.Movies[i].Title;
        var res = title.split(" ");
        res.forEach(function(element){
          if(element.length > 2 && element !== "The" && element !== "the" && element !== "and" && element !== "for"){
            emit(element,1);
            count++;
          }
        });
      }
    }
}

var redGenBodrio = function(key, cant) {
  return Array.sum(cant);
};

db.cPractica.mapReduce(mapGenBodrio,redGenBodrio,
{
  out:"pelisBodrio",
  scope:{count:0}
});

db.pelisBodrio.find().sort({value: -1}).limit(10)
```

Pruebas:

```
> db.pelisBodrio.find().sort({value: -1}).limit(10)
{ "_id" : "Man", "value" : 39 }
{ "_id" : "Love", "value" : 36 }
{ "_id" : "Last", "value" : 33 }
{ "_id" : "Dead", "value" : 31 }
```



```
{ "_id" : "House", "value" : 31 }  
{ "_id" : "Movie", "value" : 31 }  
{ "_id" : "with", "value" : 28 }  
{ "_id" : "You", "value" : 27 }  
{ "_id" : "Time", "value" : 23 }  
{ "_id" : "American", "value" : 22 }  
>
```