

Tarea
Base de Datos NoSQL

Autor:
Christopher Rodríguez Príncipe

Universidad Complutense de Madrid

```
// Autor: Christopher Rodriguez Principe
//Fecha entrega: Noviembre 25,2021
//Programa Escrito en una MAC
```

```
//0 Importar los Datos
```

```
const contents = [
  {
    content: "/Users/christopher/Downloads/movies.json",
    collection: "movies",
    idPolicy: "drop_collection_first",
    //overwrite_with_same_id|always_insert_with_new_id|insert_with_new_id_if_id_exists|skip_documents_with_existing_id|abort_if_id_already_exists|drop_collection_first|log_errors
    //Use the transformer to customize the import result
    //transformer: (doc)=>{ //async (doc)=>{
    //  doc["importDate"]= new Date()
    //  return doc; //return null skips this doc
  }
];
```

```
mb.importContent({
  connection: "localhost",
  database: "TareaModulo2",
  fromType: "file",
  batchSize: 2000,
  contents
})
drop collection "TareaModulo2.movies"
import into "TareaModulo2.movies" start...
import into "TareaModulo2.movies" 12%, 2000 docs inserted.
import into "TareaModulo2.movies" 6%, 4000 (+2000) docs inserted.
import into "TareaModulo2.movies" 19%, 6000 (+2000) docs inserted.
import into "TareaModulo2.movies" 25%, 8000 (+2000) docs inserted.
import into "TareaModulo2.movies" 32%, 10000 (+2000) docs inserted.
import into "TareaModulo2.movies" 38%, 12000 (+2000) docs inserted.
import into "TareaModulo2.movies" 45%, 14000 (+2000) docs inserted.
import into "TareaModulo2.movies" 51%, 16000 (+2000) docs inserted.
import into "TareaModulo2.movies" 58%, 18000 (+2000) docs inserted.
import into "TareaModulo2.movies" 65%, 20000 (+2000) docs inserted.
import into "TareaModulo2.movies" 72%, 22000 (+2000) docs inserted.
import into "TareaModulo2.movies" 79%, 24000 (+2000) docs inserted.
import into "TareaModulo2.movies" 87%, 26000 (+2000) docs inserted.
import into "TareaModulo2.movies" 96%, 28000 (+2000) docs inserted.
import into "TareaModulo2.movies" 100%, 28795 (+795) docs inserted.
import into "TareaModulo2.movies" finished.

A total of 28795 document(s) have been imported into 1 collection(s).
{
  "movies": {
    "nInserted": 28795,
    "nModified": 0,
    "nSkipped": 0,
    "failed": 0
  }
}
```


//1 Analizar con find la coleccion

```
db.movies.find()
```

```
/* 1 createdAt:11/9/2021, 9:56:17 AM*/
{
  "_id" : ObjectId("618a7e0170cef11f87a342cb"),
  "title" : "Caught",
  "year" : 1900,
  "cast" : [ "Unknown" ],
  "genres" : [ "Unknown" ]
},
```

//2 Contar cuantos documentos(peliculas) tien cargado

```
db.movies.find().count()
```

 0.008 s

1 28795


//3 Insertar una pelicula

```
var add_movie = {"title":
"AAA", "year":1998, "cast":["Anna", "Bob"], "genres":[]}
db.movies.insertOne(add_movie)
```

```
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61931e19462b8c87605f0f14")
}
```

//4 Borrar la pelicula insertada en el punto anterior


```
var add_movie = {"title":
"AAA", "year":1998, "cast":["Anna", "Bob"], "genres":[]}
db.movies.deleteMany(add_movie)
```

 0.064 s

```
1 {
2   "acknowledged" : true,
3   "deletedCount" : 1
4 }
```

//5 Contar cuantos "cast" se llaman "and"

```
db.movies.find({cast:"and"}).count()
```

 0.030 s


1 93

//6 Sacar el Valor "and" del array de "cast"

```
var query = {}
```

```
var operacion= { $pull: { "cast": "and" } }
```


```
db.movies.updateMany(query,operacion)
```

 0.146 s

```
1 {
2   "acknowledged" : true,
3   "matchedCount" : 28795,
4   "modifiedCount" : 93
5 }
```

//7 Contar Peliculas cuyo array de "cast" es Null

```
db.movies.find({cast: []}).count()
```

 0.035 s

1 986

//8 Actualizar todos los documentos que no tengan actores en el array con "Unknown"

```
var query= {cast: []}
```

```
var push = {$push:{"cast":"Unknown"}}
```


```
db.movies.updateMany(query,push)
```

```
db.movies.find()
```

```
1 /* 1 createdAt:11/9/2021, 9:56:17 AM*/
2 {
3   "_id" : ObjectId("618a7e0170cef11f87a342cb"),
4   "title" : "Caught",
5   "year" : 1900,
6   "cast" : [ "Unknown" ],
7   "genres" : [ "Unknown" ]
8 },
9
10 /* 2 createdAt:11/9/2021, 9:56:17 AM*/
```

//9 Contar cuantos documentos no tienen género

```
db.movies.find({genres: []}).count()
```

	0.030 s
1	901



//10 Actualizar todos los documentos que no tengan actores en el array con "Unknown"

```
var query= {genres: []}  
var push = {$push:{"genres":"Unknown"}}  
db.movies.updateMany(query,push)  
db.movies.find()
```

1	/* 1 createdAt:11/9/2021, 9:56:17 AM*/
2	{
3	"_id" : ObjectId("618a7e0170cef11f87a342cb"),
4	"title" : "Caught",
5	"year" : 1900,
6	"cast" : ["Unknown"],
7	"genres" : ["Unknown"]
8	},
9	
10	/* 2 createdAt:11/9/2021, 9:56:17 AM*/

//11 Mostrar el año más reciente que tenemos sobre todas las películas

```
var orderyear= { "year" : -1 }  
db.movies.find({}, {year:1, _id:0}).sort(orderyear).limit(1)
```

	movies		0.057 s	1 Doc
1	{			
2	"year" : 2018			
3	}			

//12 Contar cuantas películas han salido en los últimos 20 años.

```
var query1 = { _id: null, maxYear: { $max: { $add : ["$year", -  
20]} } }, items : { $push : {year: "$year", title : "$title"} } }  
var fase1 = {$group: query1}  
var fase2 = {$unwind : "$items"}  
var fase3 = {$replaceWith : {$mergeObjects: [ {maxYear:  
"$maxYear"}, "$items"]} }  
var fase4 = {$match:{$expr:{$gt:["$year", "$maxYear"]}}}  
var query5 = { _id: null, count: { $sum: 1 } }  
var fase5 = {$group:query5}
```

```
var pipe = [fase1,fase2,fase3,fase4,fase5]
db.movies.aggregate(pipe)
```

movies		0.030 s	1 Doc
1	{		
2	"_id" : null,		
3	"total" : 4787		
4	}		

//13 Contar películas que salieron durante los 60.

```
var filteryear = { "$match":{"year":{"$gte:1960,$lte:1969}} } }
var group = { $group:{_id:null,"total":{"$sum:1}} }
var years = [filteryear,group]
db.movies.aggregate(years)
```

movies		0.026 s	1 Doc
1	{		
2	"_id" : null,		
3	"total" : 1414		
4	}		

//14 Mostrar el año con más películas.

```
var orderpeli= { "peliculas" : -1 }
var group = {$group:{_id:"$year", peliculas:{$sum:1}}}
var pipeline = [group]
db.movies.aggregate(pipeline).sort(orderpeli).limit(1)
```

movies		0.042 s	1 Doc
1	{		
2	"_id" : 1919,		
3	"peliculas" : 634		
4	}		

//15 Mostrar el año con menos películas.


```
var orderpeli= { "peliculas" : 1 }
var group = {$group:{_id:"$year", peliculas:{$sum:1}}}
var pipeline = [group]
```

```
db.movies.aggregate(pipeline).sort(orderpeli).limit(3)
```

	_id	peliculas
1	1907	7
2	1902	7
3	1906	7

//16 Guardar nueva colección llamada "actos" y contar cuantos documentos existen

```
var fase1 = { $unwind: "$cast" }
var query2 = { "_id": 0 }
var fase2 = { $project: query2 }
var fase3 = { $out: "actors" }
var etapas = [ fase1, fase2, fase3 ]
db.movies.aggregate( etapas )
db.actors.find().count()
```

 0.734 s
1 83224

//17 Lista con los 5 actores que han participado en más películas

```
var query1 = {"cast":{"$ne:"Unknown"}}
var fase1 = { $match: query1 }
var query2 = { "_id": "$cast", "peliculas": { $sum: 1 } }
var fase2 = { $group: query2 }
var query3 = { "peliculas": -1 }
var fase3 = { $sort: query3 }
var etapas = [ fase1,fase2,fase3 ]
db.actors.aggregate( etapas ).limit(5)
```

	_id	peliculas
1	Harold Lloyd	190
2	Hoot Gibson	142
3	John Wayne	136
4	Charles Starrett	116
5	Bebe Daniels	103

//18 Agrupar por Title y Año y mostrar las peliculas que más actores han participado

```
var query1 = { "cast": { $ne: "Unknown" } }
var fase1 = { $match: query1 }
var query2 = { _id:
{ title: "$title", year: "$year", "cuenta": { $sum: 1 } }
var fase2 = { $group: query2 }
var query3 = { "cuenta": -1 }
var fase3 = { $sort: query3 }
var etapas = [fase1, fase2, fase3]
db.actors.aggregate( etapas ).limit(5)
```

Key	Value
▶ (1) { title : "The Twilight Saga: Break	{ cuenta : 35 }
▶ (2) { title : "Anchorman 2: The Leger	{ cuenta : 33 }
▶ (3) { title : "Cars 2", year : 2011 }	{ cuenta : 32 }
▶ (4) { title : "Avengers: Infinity War", y	{ cuenta : 29 }
▶ (5) { title : "Grown Ups 2", year : 201	{ cuenta : 28 }

//19 Mostrar 5 actores cuya carrera haya sido la más larga

```
var query1 = { "cast": { $ne: "Unknown" } }
var fase1 = { $match: query1 }
var query2 = { " _id": "$cast" , "comienza": { $min: "$year" },
"termina": { $max: "$year" } }
fase2= { $group: query2 }
var fase3 = { $addFields:
{ anos: { $subtract: [ "$termina", "$comienza" ] } } }
var query4 = { "anos": -1 }
var fase4 = { $sort: query4 }
var etapas = [fase1, fase2, fase3, fase4]
db.actors.aggregate( etapas ).limit(5)
```


Key	Value
▶ (1) Harrison Ford	{ comienza : 1919, termina : 2017, anos : 98 } (4 fields)
▶ (2) Gloria Stuart	{ comienza : 1932, termina : 2012, anos : 80 } (4 fields)
▶ (3) Lillian Gish	{ comienza : 1912, termina : 1987, anos : 75 } (4 fields)
▶ (4) Kenny Baker	{ comienza : 1937, termina : 2012, anos : 75 } (4 fields)
▶ (5) Mickey Rooney	{ comienza : 1932, termina : 2006, anos : 74 } (4 fields)

//20 Contar cuantos documentos hay en la coleccion "genres"

```
var fase1 = { $unwind: "$genres" }
var query2 = { "_id": 0 }
var fase2 = { $project: query2 }
var fase3 = { $out: "genres" }
var etapas = [ fase1, fase2, fase3 ]
db.actors.aggregate( etapas )
db.genres.find().count()
```

1 104950

//21 mostrar los 5 documentos agrupados por "Año y Género" que mas numero de peliculas diferentes tienen.

```
var query1 = { _id: { "year": "$year", "genres": "$genres" },
peliculas: { $addToSet: "$title" } }
var fase1 = { $group: query1 }
var etapa2 = { "_id": "$_id", peliculas: { "$size": "$peliculas" } }
var fase2 = { $project: etapa2 }
var fase3 = { $sort: { "peliculas": -1 } }
var pipe = [ fase1, fase2, fase3 ]
db.genres.aggregate( pipe ).limit(5)
```

	_id		peliculas
genres		year	
1 Drama		1919	291
2 Drama		1925	247
3 Drama		1924	233
4 Comedy		1919	226
5 Drama		1922	209

//22 mostrar 5 actores y los generos en los que participaron con más numeros diferentes.

```
var genres = "$genres"
var movies = { "TotalGeneros": -1 }
var query1 = {"cast":{"$ne:"Unknown"}}
var fase1 = { $match: query1 }
var query2 = { _id: { "cast": "$cast" },  generos:{$addToSet:
genres} }
var fase2 = {$group:query2}
var fase3 = { $addFields: {numgeneros: { $size: ["$generos"] } } }
}
var query4 = {"numgeneros":-1}
var fase4 = {$sort:query4}
var pipeline = [fase1, fase2,fase3,fase4]
db.genres.aggregate(pipeline).limit(5)
```

```
1  /* 1 */
2  {
3    "_id" : {
4      "cast" : "Dennis Quaid"
5    },
6    "generos" : [
7      "Drama",
8      "Comedy",
9      "Animated",
10     "Thriller",
11     "Satire",
12     "Horror",
13     "Disaster",
14     "Adventure",
15     "Action",
16     "Dance",
17     "Romance",
18     "Western",
19     "Crime",
20     "Biography",
21     "Fantasy",
22     "Family",
23     "Science Fiction",
24     "Musical",
25     "Sports",
26     "Suspense"
27   ],
28   "numgeneros" : 20
29 },
```

//23 Mostrar 5 películas en los que más géneros diferentes han sido catalogados.

```
var docu = "$genres"
var query1 = { "genres": { $ne: "Unknown" } }
var fase1 = {$match: query1}
var query2 = { _id: { title: "$title", year:"$year"},
genres:{$addToSet: docu} }
var fase2 = {$group:query2}
var fase3 = { $addFields: {numgeneros: { $size: ["$genres"] } } }
}
var query4 = {"numgeneros":-1}
var fase4 = {$sort:query4}
var pipeline = [fase1, fase2,fase3,fase4]
db.genres.aggregate(pipeline).limit(5)
```

```
1 /* 1 */
2 {
3   "_id" : {
4     "title" : "American Made",
5     "year" : 2017
6   },
7   "generos" : [
8     "Comedy",
9     "Crime",
10    "Action",
11    "Drama",
12    "Thriller",
13    "Historical",
14    "Biography"
15  ],
16   "numgeneros" : 7
17 },
```

// 24 Lista de Peliculas que salieron en el 2018, Convertir la lista en Mayuscula y ordear de manera ascendente.

```
var query1 = { "year": 2018 }
var fase1 = { $match: query1 }
var query2 = { "_id": "$title" }
var fase2 = { $group: query2 }
var fase3 = { $project: { _id: { $toUpper: "$_id" } } }
var fase4 = { $sort: { "_id": 1 } }
pipe = [fase1, fase2, fase3, fase4]
db.genres.aggregate(pipe)
```

	_id ↕
1	12 STRONG
2	7 DAYS IN ENTEBBE
3	A FANTASTIC WOMAN
4	A PRAYER BEFORE DAWN
5	A PRIVATE WAR
6	A QUIET PLACE
7	A SIMPLE FAVOR
8	A STAR IS BORN


// 25 Lista de actores que han participado de una pelicula con genero que termine con la letra "t".

```
query1 = { "genres" : { $regex: /t$/ } }
fase1 = { $match: query1 }
var query2 = { "_id": 0, "title": 0, "year": 0 }
var fase2 = { $project: query2 }
var query3 = { "cast": { $ne: "Unknown" } }
var fase3 = { $match: query3 }
pipe = [fase1, fase2, fase3]
db.genres.aggregate(pipe)
```

Key ↕	Value 📄
▶ (1)	{ cast : "Ching Ling Foo", genres : "Short" }
▶ (2)	{ cast : "J. Stuart Blackton", genres : "Short" }
▶ (3)	{ cast : "Florence Lawrence", genres : "Short" }
▶ (4)	{ cast : "J. Stuart Blackton", genres : "Short" }
▶ (5)	{ cast : "Henry B. Walthall", genres : "Short" }
▶ (6)	{ cast : "Anna Rosemond", genres : "Short" }
▶ (7)	{ cast : "George Barnes", genres : "Short" }

//26 Crear lista concatenando el titulo de la pelicula y los actores que participan en ella + convertirlo en mayuscula todo.

```
var query1 = { "cast": { $ne: "Unknown" } }
var fase1 = {$match: query1}
var concatenar = { $concat: [ "$title", " --> ", "$cast" ] }
var toUp = { $toUpper: concatenar }
var fase2 = { $project: { _id: 0, "Lista" : toUp } }
var pipe = [ fase1, fase2 ]
db.genres.aggregate( pipe )
```

Lista 	
1	FEEDING SEA LIONS --> PAUL BOYTON
2	THE WONDER, CHING LING FOO --> CHING LING FOO
3	LOVE'S PENALTY --> HOPE HAMPTON
4	LOVE'S PENALTY --> PERCY MARMONT
5	LOVE'S PENALTY --> VIRGINIA VALLI
6	THE LUCKY DOG --> STAN LAUREL
7	THE LUCKY DOG --> OLIVER HARDY
8	THE LURE OF YOUTH --> CLEO MADISON
9	THE LURE OF YOUTH --> GARETH HUGHES
10	JIM THE PENMAN --> LIONEL BARRYMORE
11	JIM THE PENMAN --> LIONEL BARRYMORE
12	A MAN OF STONE --> CONWAY TEARLE