2024

Módulo 3:
Bases de Datos NoSQL

Proyecto Final

JACOBO ÁLVAREZ GUTIÉRREZ

MÁSTER EN BIG DATA E INTELIGENCIA ARTIFICIAL | Universidad Complutense de Madrid (UCM)

Contenido

EJERCICIO 0

Realizar la importación del json en una colección llamada "movies".

En primer lugar, necesitamos importar los datos del archivo JSON. Para ello, creamos una database desde *"MongoDB Compass"* tal y como se muestra en la *Figura 1.*

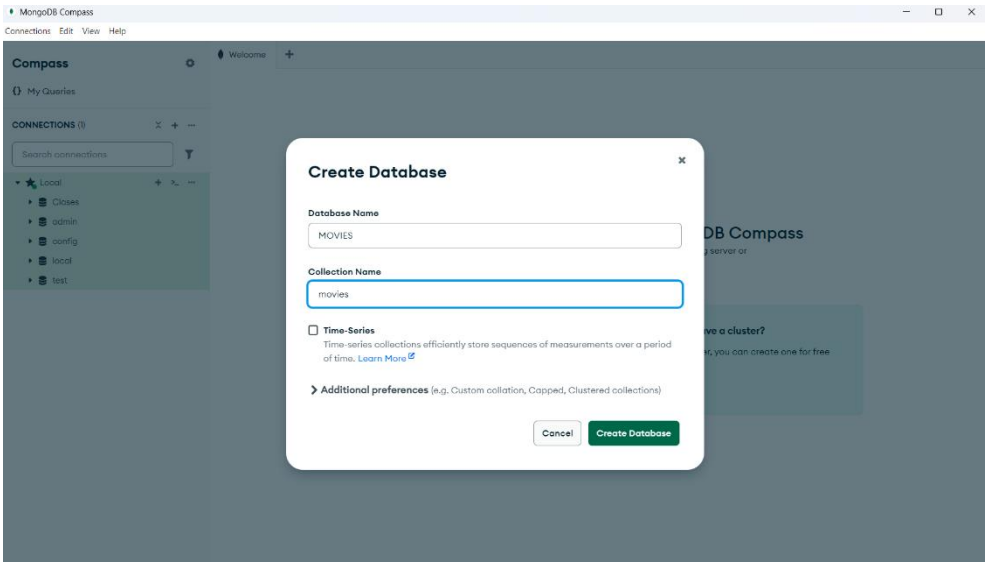


FIGURA 1. Creación de una Database para importar el conjunto de datos del archivo JSON.

Una vez creada, procedemos a importar los datos del archivo *"movies.json"*. Una vez realizado este paso esperamos un resultado como el que se muestra en la *Figura 2*.

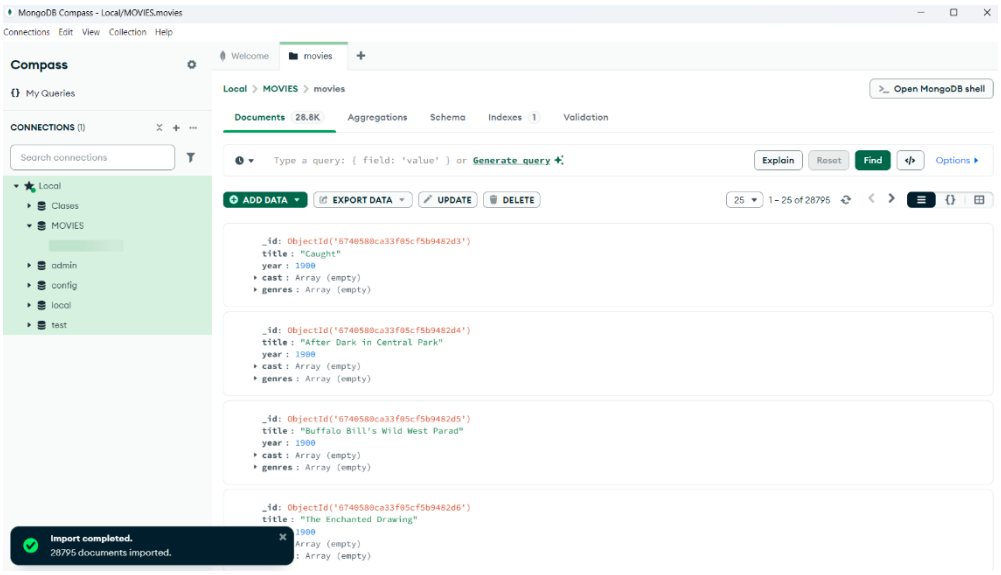


FIGURA 2. Resultado de importar el dataset en la database creada en el paso anterior.

Llegados a este punto, estamos en disposición de pasar a trabajar con el cliente "*NoSQLBooster for MongoDB"*. Bastará con emplear el comando:

```
use "MOVIES"
```

EJERCICIO 1

Analizar con find la colección.

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


```
{
    "_id" : ObjectId("6740580ca33f05cf5b9482d3"),
    "title" : "Caught",
    "year" : 1900,
    "cast" : [ ],
    "genres" : [ ]
},

/* 2 createdAt:22/11/2024 10:08:12*/
{
    "_id" : ObjectId("6740580ca33f05cf5b9482d4"),
    "title" : "After Dark in Central Park",
    "year" : 1900,
    "cast" : [ ],
    "genres" : [ ]
},
```


FIGURA 3. Salida obtenida en el ejercicio 1 tras la ejecución del script.

EJERCICIO 2

Contar cuántos documentos (películas) tiene cargado.

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

FIGURA 4. Salida obtenida en el ejercicio 2 tras la ejecución del script.

EJERCICIO 3

Insertar una película.

```
var nuevo_item = {"title": "Película de prueba", "year": "2050",
"cast": [], "genres": []}

db.movies.insertOne(nuevo_item)
```

FIGURA 5. Salida obtenida en el ejercicio 3 tras la ejecución del script.

EJERCICIO 4

Borrar la película insertada en el punto anterior (en el 3).

```
var nuevo_item = {"title": "Película de prueba", "year": "2050",
"cast": [], "genres": []}

db.movies.deleteOne(nuevo_item)
```

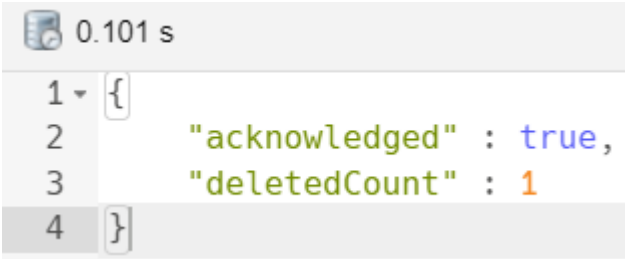


FIGURA 6. Salida obtenida en el ejercicio 4 tras la ejecución del script.

EJERCICIO 5

Contar cuántas películas tienen actores (cast) que se llaman "and". Estos nombres de actores están por ERROR.

```
var query = {"cast": "and"}

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

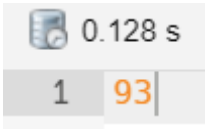


FIGURA 7. Salida obtenida en el ejercicio 5 tras la ejecución del script.

EJERCICIO 6

Actualizar los documentos cuyo actor (cast) tenga por error el valor "and" como si realmente fuera un actor. Para ello, se debe sacar únicamente este valor del array cast. Por lo tanto, no se debe eliminar ni el documento (película) ni su array cast con el resto de actores.

```
var query = {"cast": "and"}

var operation = {$pull: query}

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

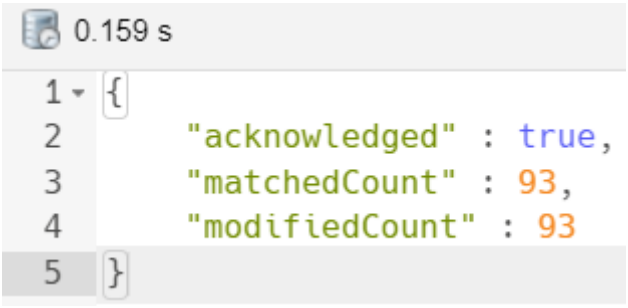


FIGURA 8. Salida obtenida en el ejercicio 6 tras la ejecución del script.

EJERCICIO 7

Contar cuantos documentos (películas) tienen el array "cast" vacío.

```
var query1 = {"cast": {$size: 0}}
var query2 = {"cast": {$exists: false}}
var queryT = {$or: [query1, query2]}
db.movies.find(queryT).count()
```

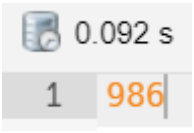


FIGURA 9. Salida obtenida en el ejercicio 7 tras la ejecución del script.

EJERCICIO 8

Actualizar TODOS los documentos (películas) que tengan el array cast vacío, añadiendo un nuevo elemento dentro del array con valor Undefined. Cuidado! El tipo de cast debe seguir siendo un array. El array debe ser así -> ["Undefined"].

```
var query1 = {"cast": {$size: 0}}
var query2 = {"cast": {$exists: false}}
var queryT = {$or: [query1, query2]}
var operation = {$set: {"cast": ["Undefined"]}}
db.movies.updateMany(queryT, operation)
```

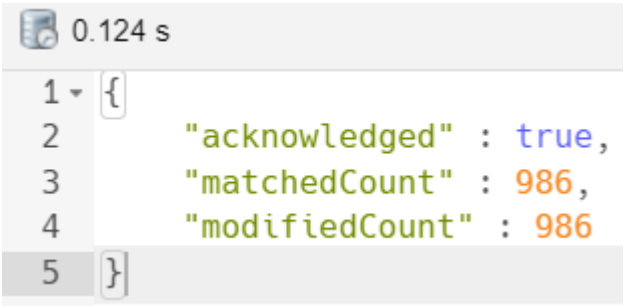


FIGURA 10. Salida obtenida en el ejercicio 8 tras la ejecución del script.

EJERCICIO 9

Contar cuántos documentos (películas) tienen el array genres vacío.

```
var query1 = {"genres": {$size: 0}}
var query2 = {"genres": {$exists: false}}
var queryT = {$or: [query1, query2]}
db.movies.find(queryT).count()
```

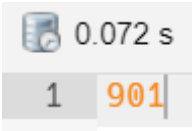


FIGURA 11. Salida obtenida en el ejercicio 9 tras la ejecución del script.

EJERCICIO 10

Actualizar TODOS los documentos (películas) que tengan el array genres vacío, añadiendo un nuevo elemento dentro del array con valor Undefined. Cuidado! El tipo de genres debe seguir siendo un array. El array debe ser así -> [“Undefined”].

```
var query1 = {"genres": {$size: 0}}
var query2 = {"genres": {$exists: false}}
var queryT = {$or: [query1, query2]}
var operation = {$set: {"genres": ["Undefined"]}}
db.movies.updateMany(queryT, operation)
db.movies.find()
```

FIGURA 12. Salida obtenida en el ejercicio 10 tras la ejecución del script.

EJERCICIO 11

Mostar el año más reciente/actual que tenemos sobre todas las películas.

```
var query = {}

var projection = {"_id": 0, "year": 1}

db.movies.find(query, projection).sort({"year": -1}).limit(1)
```

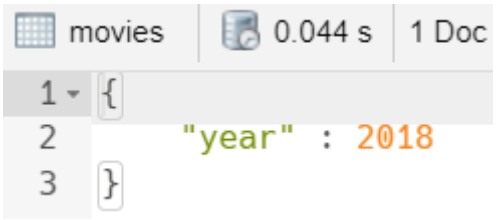


FIGURA 13. Salida obtenida en el ejercicio 11 tras la ejecución del script.

EJERCICIO 12

Contar cuántas películas han salido en los últimos 20 años. Debe hacerse desde el último año que se tienen registradas películas en la colección, mostrando el resultado total de esos años. Se debe hacer con el Framework de Agregación.

```
var query1 = {"_id": "$year", "num_peliculas": {$sum: 1}}

var fase1 = {$group: query1}

var query2 = {"_id": -1}
```

```
var fase2 = {$sort: query2}
var fase3 = {$limit: 20}
var query4 = {"_id": null, "total": {$sum: "$num_peliculas"}}
var fase4 = {$group: query4}
var etapas = [fase1, fase2, fase3, fase4]
db.movies.aggregate(etapas)
```

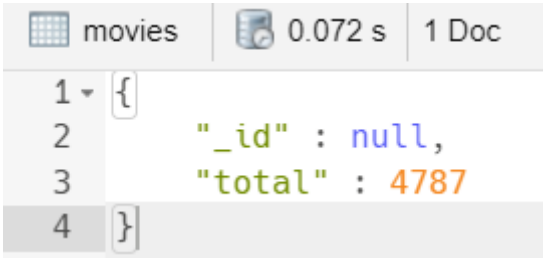


FIGURA 14. Salida obtenida en el ejercicio 12 tras la ejecución del script.

EJERCICIO 13

Contar cuántas películas han salido en la década de los 60 (del 60 al 69 incluidos). Se debe hacer con el Framework de Agregación.

```
var query1 = {"year": {$gte: 1960, $lte: 1969}}
var fase1 = {$match: query1}
var query2 = {"_id": null, "total": {$sum: 1}}
var fase2 = {$group: query2}
var etapas = [fase1, fase2]
db.movies.aggregate(etapas)
```

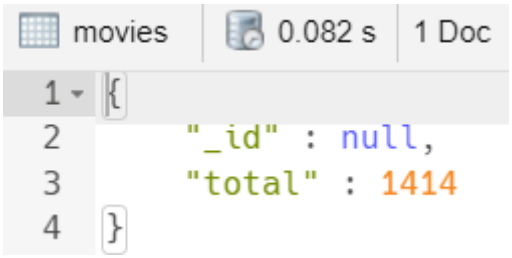


FIGURA 15. Salida obtenida en el ejercicio 13 tras la ejecución del script.

EJERCICIO 14

Mostar el año u años con más películas mostrando el número de películas de ese año. Revisar si varios años pueden compartir tener el mayor número de películas.

```
var query1 = {"_id": "$year", "num_peliculas": {$sum: 1}}
var fase1 = {$group: query1}
var fase2 = {$sort: {"num_peliculas": -1}}
var fase3 = {$limit: 1}
var etapas1 = [fase1, fase2, fase3]
var maxpelis = db.movies.aggregate(etapas1).toArray()[0].num_peliculas
var query4 = {"num_peliculas": {$eq: maxpelis}}
var fase4 = {$match: query4}
var etapas2 = [fase1, fase4]
db.movies.aggregate(etapas2)
```

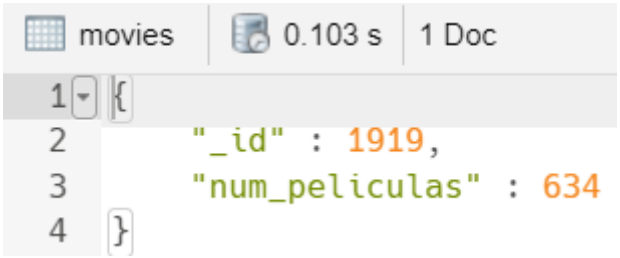


FIGURA 16. Salida obtenida en el ejercicio 14 tras la ejecución del script.

EJERCICIO 15

Mostrar el año u años con menos películas mostrando el número de películas de ese año. Revisar si varios años pueden compartir tener el menor número de películas.

```
var query1 = {"_id": "$year", "num_peliculas": {$sum: 1}}
var fase1 = {$group: query1}
var fase2 = {$sort: {"num_peliculas": 1}}
var fase3 = {$limit: 1}
var etapas1 = [fase1, fase2, fase3]
var minpelis = db.movies.aggregate(etapas1).toArray()[0].num_peliculas
```

```
var query4 = {"num_peliculas": {$eq: minpelis}}
var fase4 = {$match: query4}
var etapas2 = [fase1, fase4]
db.movies.aggregate(etapas2)
```

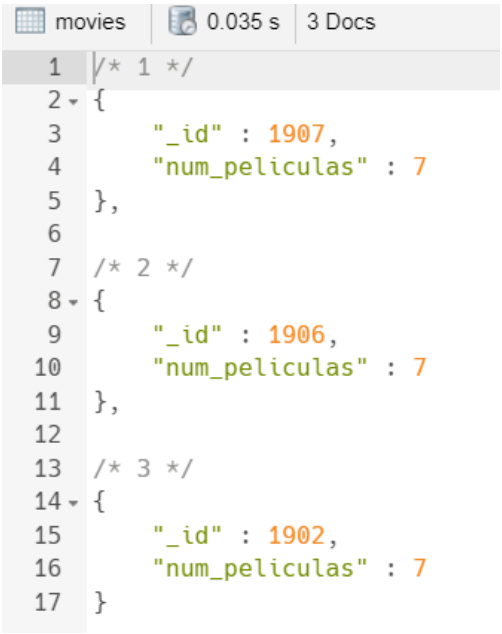


FIGURA 17. Salida obtenida en el ejercicio 15 tras la ejecución del script.

EJERCICIO 16

Guardar en una nueva colección llamada "actors" realizando la fase $unwind por actor. Después, contar cuántos documentos existen en la nueva colección.

```
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.count()
```

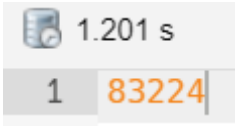


FIGURA 18. Salida obtenida en el ejercicio 16 tras la ejecución del script.

EJERCICIO 17

Sobre actors (nueva colección), mostrar la lista con los 5 actores que han participado en más películas mostrando el número de películas en las que ha participado. Importante! Se necesita filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, solo que filtramos para que no aparezcan.

```
var query1 = {"_id": "$cast", "num_peliculas": {$sum: 1}}
var fase1 = {$group: query1}
var query2 = {"_id": {$ne: "Undefined"}}
var fase2 = {$match: query2}
var fase3 = {$sort: {"num_peliculas": -1}}
var fase4 = {$limit: 5}
var etapas = [fase1, fase2, fase3, fase4]
db.actors.aggregate(etapas)
```

actors | 0.118 s | 5 Docs

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

FIGURA 19. Salida obtenida en el ejercicio 17 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización.

EJERCICIO 18

Sobre actors (nueva colección), agrupar por película y año mostrando las 5 en las que más actores hayan participado, mostrando el número total de actores.

```
var query1 = {"cast": {$ne: "Undefined"}}
var fase1 = {$match: query1}
var query2 = {"_id": {"título": "$title", "año": "$year"}, "num_actors": {$sum: 1}}
var fase2 = {$group: query2}
var fase3 = {$sort: {"num_actors": -1}}
```

```
var fase4 = {$limit: 5}

var etapas = [fase1, fase2, fase3, fase4]

db.actors.aggregate(etapas)
```

actors | 0.274 s | 5 Docs

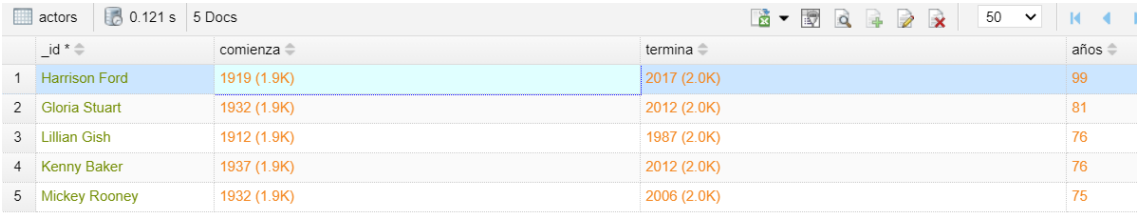
	_id		num_actors
	título	año	
1	The Twilight Saga: Breaking Dawn - Part 2	2012 (2.0K)	35
2	Anchorman 2: The Legend Continues	2013 (2.0K)	33
3	Cars 2	2011 (2.0K)	32
4	Avengers: Infinity War	2018 (2.0K)	29
5	Grown Ups 2	2013 (2.0K)	28

FIGURA 20. Salida obtenida en el ejercicio 18 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización.

EJERCICIO 19

Sobre actors (nueva colección), mostrar los 5 actores cuya carrera haya sido la más larga. Para ello, se debe mostrar cuándo empezó su carrera, cuándo finalizó y cuántos años ha trabajado. Importante! Se necesita previamente filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, solo que filtramos para que no aparezcan.

```
var query1 = {"cast": {$ne: "Undefined"}}

var fase1 = {$match: query1}

var query2 = {"_id": "$cast", "comienza": {$min: "$year"}, "termina": {$max: "$year"}}

var fase2 = {$group: query2}

var query3 = {"_id": 1, "comienza": 1, "termina": 1, "años": {$subtract: [{$add: ["$termina", 1]}, "$comienza"]}}

var fase3 = {$project: query3}

var fase4 = {$sort: {"años": -1}}

var fase5 = {$limit: 5}

var etapas = [fase1, fase2, fase3, fase4, fase5]

db.actors.aggregate(etapas)
```



actors | 0.121 s | 5 Docs

	_id *	comienza	termina	años
1	Harrison Ford	1919 (1.9K)	2017 (2.0K)	99
2	Gloria Stuart	1932 (1.9K)	2012 (2.0K)	81
3	Lillian Gish	1912 (1.9K)	1987 (2.0K)	76
4	Kenny Baker	1937 (1.9K)	2012 (2.0K)	76
5	Mickey Rooney	1932 (1.9K)	2006 (2.0K)	75

FIGURA 21. Salida obtenida en el ejercicio 19 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización.

EJERCICIO 20

Sobre actors (nueva colección), guardar en una nueva colección llamada "genres" realizando la fase $unwind por genres. Después, contar cuántos documentos existen en la nueva colección.

```
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.count()
```

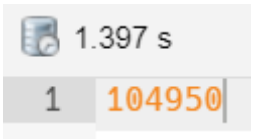


FIGURA 22. Salida obtenida en el ejercicio 20 tras la ejecución del script.

EJERCICIO 21

Sobre genres (nueva colección), mostrar los 5 documentos agrupados por "Año y Género" que más número de películas diferentes tienen mostrando el número total de películas.

```
var query1 = {"_id": {"año": "$year", "género": "$genres"}, "peliculas": {$addToSet: "$title"}}

var fase1 = {$group: query1}

var query2 = {"_id": 1, "num_películas_dif": {$size: "$peliculas"}}

var fase2 = {$project: query2}

var fase3 = {$sort: {"num_películas_dif": -1}}

var fase4 = {$limit: 5}
```

```
var etapas = [fase1, fase2, fase3, fase4]

db.genres.aggregate(etapas)
```

genres | 0.153 s | 5 Docs

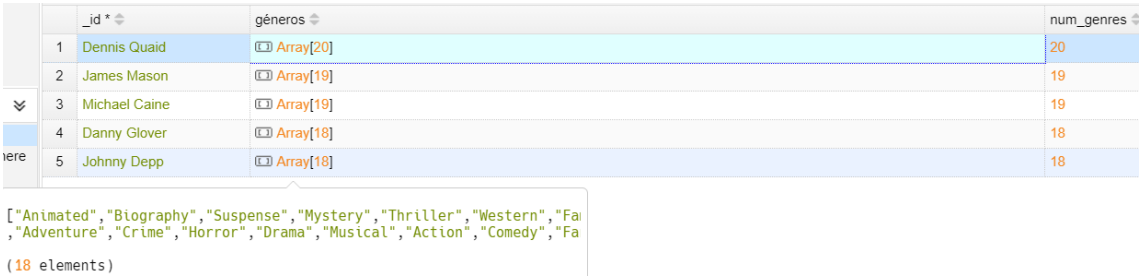
	_id		num_películas_dif
	año	género	
1	1919 (1.9K)	Drama	291
2	1925 (1.9K)	Drama	247
3	1924 (1.9K)	Drama	233
4	1919 (1.9K)	Comedy	226
5	1922 (1.9K)	Drama	209

FIGURA 23. Salida obtenida en el ejercicio 21 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización.

EJERCICIO 22

Sobre genres (nueva colección), mostrar los 5 actores y los géneros en los que han participado con más número de géneros diferentes. Se debe mostrar el número de géneros diferentes que ha interpretado. Importante! Se necesita previamente filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, solo que filtramos para que no aparezcan.

```
var query1 = {"cast": {$ne: "Undefined"}}

var fase1 = {$match: query1}

var query2 = {"_id": "$cast", "géneros": {$addToSet: "$genres"},}

var fase2 = {$group: query2}

var query3 = {"_id": 1, "num_genres": {$size: "$géneros"}, "géneros": 1}

var fase3 = {$project: query3}

var fase4 = {$sort: {"num_genres": -1}}

var fase5 = {$limit: 5}

var etapas = [fase1, fase2, fase3, fase4, fase5]

db.genres.aggregate(etapas)
```



	_id *	géneros	num_genres
1	Dennis Quaid	Array[20]	20
2	James Mason	Array[19]	19
3	Michael Caine	Array[19]	19
4	Danny Glover	Array[18]	18
5	Johnny Depp	Array[18]	18


```
["Animated","Biography","Suspense","Mystery","Thriller","Western","Fa
,"Adventure","Crime","Horror","Drama","Musical","Action","Comedy","Fa

(18 elements)
```


FIGURA 24. Salida obtenida en el ejercicio 22 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización. Además, se muestra al menos una de las listas de géneros.

EJERCICIO 23

Sobre genres (nueva colección), mostrar las 5 películas y su año correspondiente en los que más géneros diferentes han sido catalogados, mostrando esos géneros y el número de géneros que contiene.

```
var query1 = {"_id": {"título": "$title", "año": "$year"}, "géneros": {$addToSet: "$genres"}}

var fase1 = {$group: query1}

var query2 = {"num_genres": {$size: "$géneros"}, "géneros": 1}

var fase2 = {$project: query2}

var fase3 = {$sort: {"num_genres": -1}}

var fase4 = {$limit: 5}

var etapas = [fase1, fase2, fase3, fase4]

db.genres.aggregate(etapas)
```

FIGURA 25. Salida obtenida en el ejercicio 23 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización. Además, se muestra al menos una de las listas de géneros.

EJERCICIO 24 (LIBRE ELECCIÓN)

Listado de los distintos géneros que se encuentran en el dataset y la cantidad de películas asociadas a cada uno.

```
var query1 = {$and: [{"cast": {$ne: ["Undefined"]}}, {"genres": {$ne: ["Undefined"]}}]}

var fase1 = {$match: query1}

var fase2 = {$unwind: "$genres"}

var query3 = {"_id": "$genres", "num_peliculas": {$sum: 1}}

var fase3 = {$group: query3}

var fase4 = {$sort: {"num_peliculas": -1}}

var etapas = [fase1, fase2, fase3, fase4]
```

```
db.movies.aggregate(etapas)
```

movies | 0.079 s | 40 Docs

	_id *	num_peliculas
1	Drama	8688 (8.7K)
2	Comedy	7332 (7.3K)
3	Western	3010 (3.0K)
4	Crime	1497 (1.5K)
5	Musical	1163 (1.2K)
6	Romance	1145 (1.1K)
7	Horror	1131 (1.1K)
8	Action	1105 (1.1K)
9	Adventure	997
10	Thriller	864
11	Science Fiction	775
12	Mystery	627

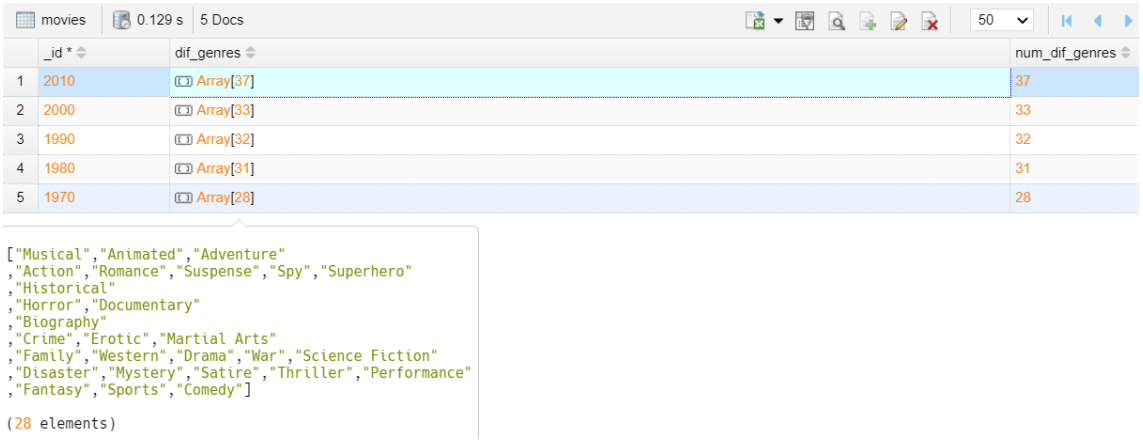
FIGURA 26. Salida obtenida en el ejercicio 24 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización, aunque no se aprecian los 40 documentos resultantes.

EJERCICIO 25(LIBRE ELECCIÓN)

Listado de todos los géneros diferentes en cada una de las últimas cinco décadas, así como el número de géneros diferentes.

```
var query1 = {$and: [{"cast": {$ne: ["Undefined"]}}, {"genres": {$ne: ["Undefined"]}}]}

var fase1 = {$match: query1}

var fase2 = {$unwind: "$genres"}

var query3 = {"década": {$subtract: ["$year", {$mod: ["$year", 10]}]}, "genres": 1}

var fase3 = {$project: query3}

var query4 = {"_id": "$década", "dif_genres": {$addToSet: "$genres"}}

var fase4 = {$group: query4}

var query5 = {"_id": 1, "num_dif_genres": {$size: "$dif_genres"}, "dif_genres": 1}

var fase5 = {$project: query5}
```

```
var fase6 = {$sort: {"_id": -1}}

var fase7 = {$limit: 5}

var etapas = [fase1, fase2, fase3, fase4, fase5, fase6, fase7]

db.movies.aggregate(etapas)
```



movies | 0.129 s | 5 Docs | 50

	_id *	dif_genres	num_dif_genres
1	2010	Array[37]	37
2	2000	Array[33]	33
3	1990	Array[32]	32
4	1980	Array[31]	31
5	1970	Array[28]	28

```
["Musical","Animated","Adventure"
,"Action","Romance","Suspense","Spy","Superhero"
,"Historical"
,"Horror","Documentary"
,"Biography"
,"Crime","Erotic","Martial Arts"
,"Family","Western","Drama","War","Science Fiction"
,"Disaster","Mystery","Satire","Thriller","Performance"
,"Fantasy","Sports","Comedy"]

(28 elements)
```

FIGURA 27. Salida obtenida en el ejercicio 25 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización. Además, se muestra al menos una de las listas de géneros.

EJERCICIO 26 (LIBRE ELECCIÓN)

Listado con los 5 géneros con mayor promedio anual de películas. Debe tenerse en cuenta el primer y último registro de la fecha en la que aparece cada género para hacer una correcta distribución de media anual.

```
var query1 = {"genres": {$ne: ["Undefined"]}}

var fase1 = {$match: query1}

var fase2 = {$unwind: "$genres"}

var query3 = {"_id": {"año": "$year", "género": "$genres"}, "películas": {$sum: 1}}

var fase3 = {$group: query3}

var query4 = {"_id": "$_id.género", "total_películas": {$sum: "$películas"},
              "primer_registro": {$min: "$_id.año"}, "último_registro": {$max: "$_id.año"}}

var fase4 = {$group: query4}

var query5 = {"_id": 1, "primer_registro": 1, "último_registro": 1,
```

```
            "avg_pelis_anuales":      {$round:      [{$divide:
["$total_películas",{$add:   [{$subtract:   ["$último_registro",
"$primer_registro"]}, 1]}]},2]}}

var fase5 = {$project: query5}

var fase6 = {$sort: {"avg_pelis_anuales": -1}}

var fase7 = {$limit: 5}

var etapas = [fase1, fase2, fase3, fase4, fase5, fase6, fase7]

db.movies.aggregate(etapas)
```

movies | 0.079 s | 5 Docs | 50

	_id *	primer_registro	último_registro	avg_pelis_anuales
1	Drama	1906 (1.9K)	2018 (2.0K)	77,38.0
2	Comedy	1900 (1.9K)	2018 (2.0K)	61,87.0
3	Western	1903 (1.9K)	2018 (2.0K)	25,97.0
4	Crime	1906 (1.9K)	2018 (2.0K)	13,27.0
5	Musical	1928 (1.9K)	2018 (2.0K)	12,81.0

FIGURA 28. Salida obtenida en el ejercicio 26 tras la ejecución del script. Se muestra en formato tabla para una mejor visualización.