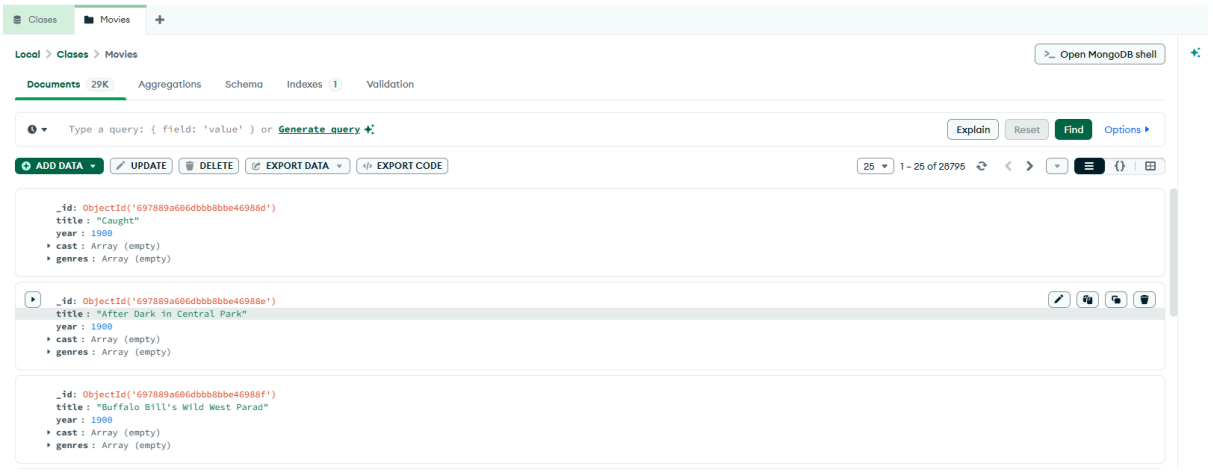
TAREA FINAL NoSQL CARLOS LARA MARIN 77767539P

Ejercicios a resolver:

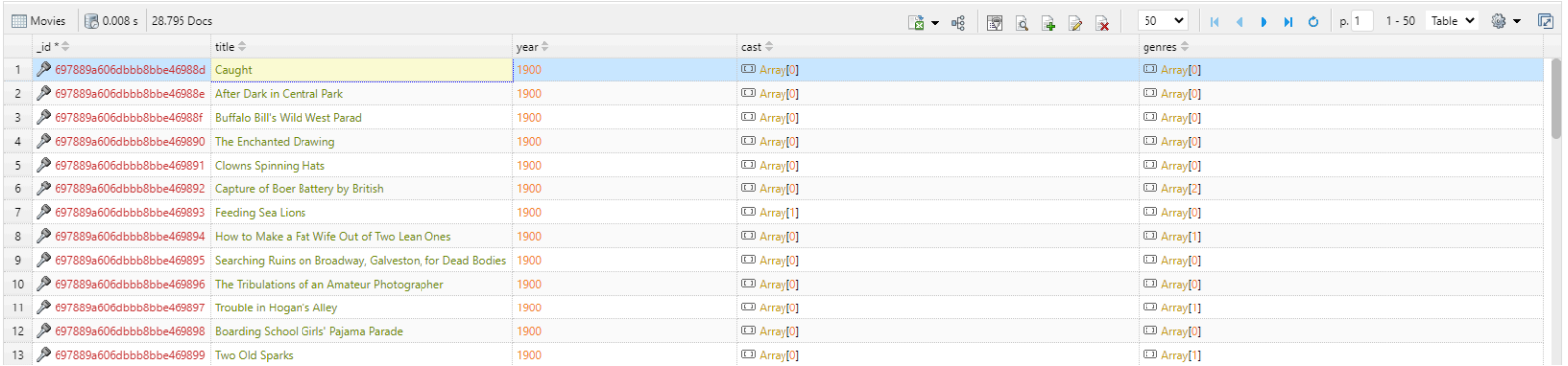
0. Realizar la importación del json en una colección llamada “movies”.

Hemos importado todo el datasets de Movies, el cual contiene 28.795 documentos.



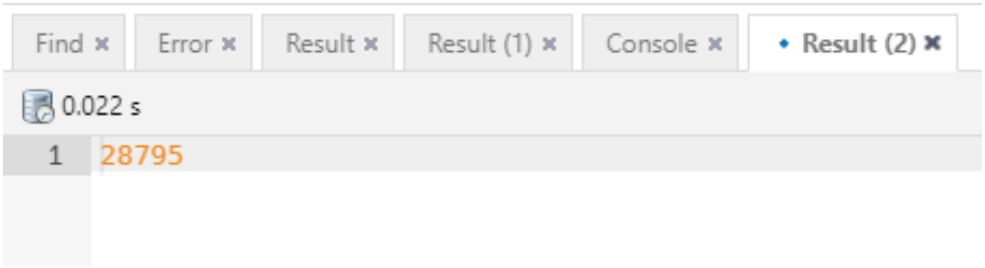
1. Analizar con find la colección.

db.Movies.find()



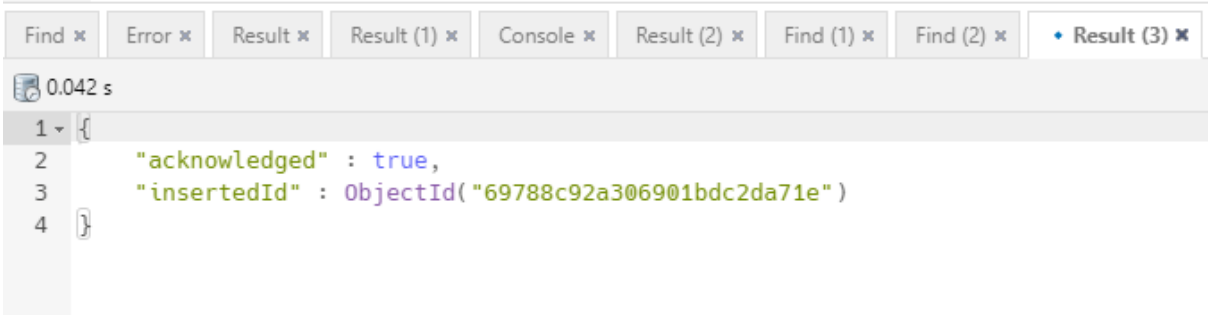
2. Contar cuántos documentos (películas) tiene cargado.

db.Movies.count()



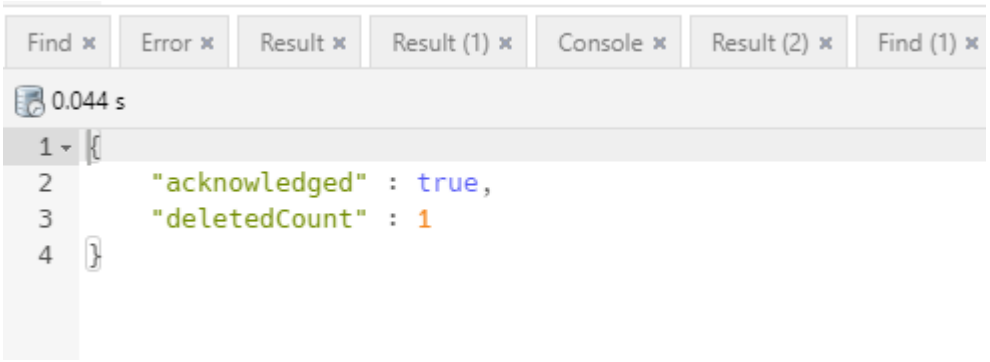
3. Insertar una película.

```
db.Movies.insertOne({
    title: "Interstellar"
    year: 2014,
    cast: ["Matthew McConaughey", "Anne Hathaway", "Jessica Chastain"],
    genres: ["Science Fiction", "Drama"]
})
```



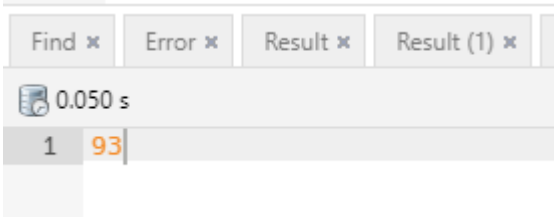
4. Borrar la película insertada en el punto anterior (en el 3).

db.Movies.deleteOne({ title: "Interestellar"})



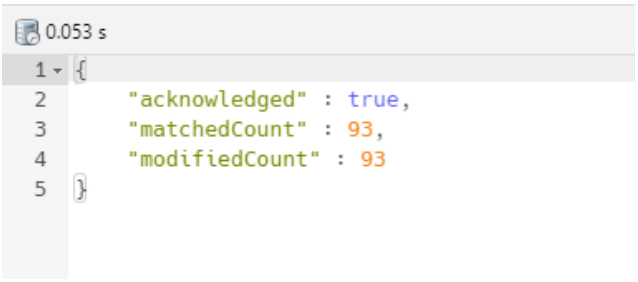
5. Contar cuantas películas tienen actores (cast) que se llaman “and”. Estos nombres de actores están por ERROR.

db.Movies.count({cast: "and"})



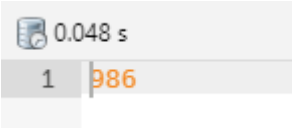
6. Actualizar los documentos cuyo actor (cast) tenga por error el valor “and” como si realmente fuera un actor. Para ello, se debe sacar únicamente ese valor del array cast. Por lo tanto, no se debe eliminar ni el documento (película) ni su array cast con el resto de actores.

db.Movies.updateMany(
{ cast: "and"},
{ $pull: { cast: "and"} }
)



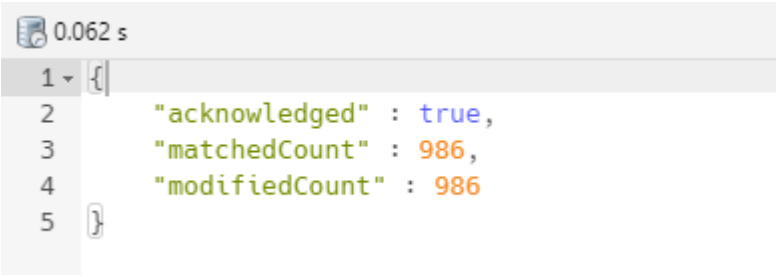
7. Contar cuantos documentos (películas) tienen el array ‘cast’ vacío.

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



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"].

```
db.Movies.updateMany(
   { cast: [] },
   {$push: { cast: "Undefined"}}
   )
```

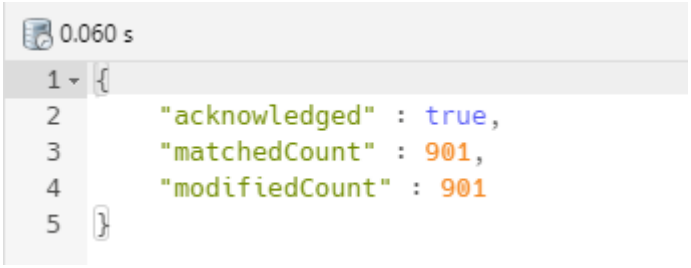


9. Contar cuantos documentos (películas) tienen el array genres vacío.

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

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"].

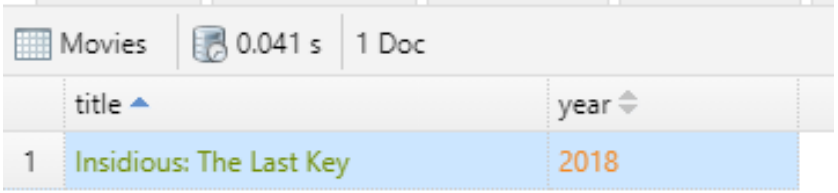
```
db.Movies.updateMany(
   { genres: [] },
   {$push: { genres: "Undefined"}}
   )
```



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

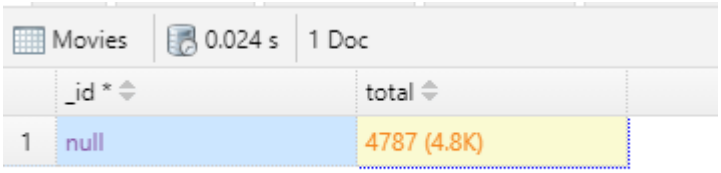
db.Movies.find().sort({year: -1}).limit(1)

db.Movies.find({}, {_id:0,title:1, year: 1}).sort({year: -1}).limit(1)



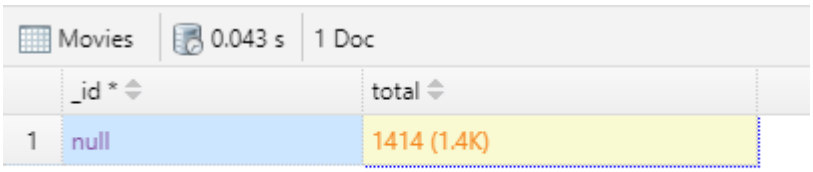
12. Contar cuántas películas han salido en los últimos 20 años. Debe hacerse desde el último año que se tienen registradas películas en la colección, no desde la fecha actual, mostrando el resultado total de esos años, no el resultado por año. Revisar que no se cuentan 21 años. Se debe hacer con el Framework de Agregación.

```
var fase1 = { $match: { year: { $gte: 1999, $lte: 2018 } } }
var fase2 = { $group: { _id: null, total: { $sum: 1 } } }

db.Movies.aggregate([fase1, fase2])
```



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 fase1 = { $match: { year: { $gte: 1960, $lte: 1969 } } }
var fase2 = { $group: { _id: null, total: { $sum: 1 } } }

db.Movies.aggregate([fase1, fase2])
```



14. Mostrar 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 fase1 = {$group: {_id: "$year", pelis: { $sum:1 } } }
var fase2 = {$sort: {pelis: -1 } }
var fase3 = {$limit: 1 }

db.Movies.aggregate([fase1, fase2, fase3])
```

Movies | 0.041 s | 1 Doc

	_id *	pelis
1	1919	634

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.

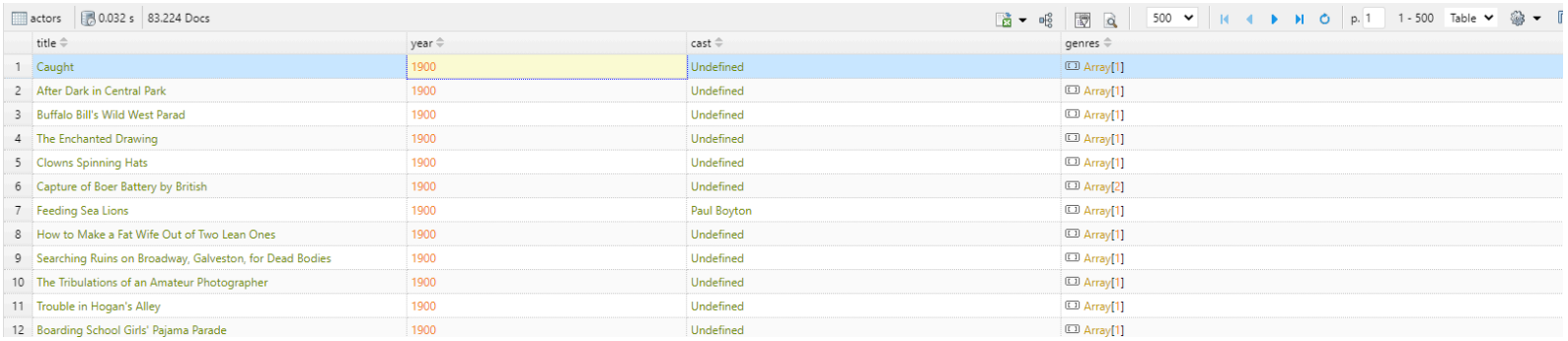
```
var fase1 = {$group: {_id: "$year", pelis: { $sum:1 } } }
var fase2 = {$sort: {pelis: 1 } }
var fase3 = {$limit: 3 }

db.Movies.aggregate([fase1, fase2, fase3])
```

Movies | 0.040 s | 3 Docs

	_id *	pelis
1	1907	7
2	1902	7
3	1906	7

16. Guardar en nueva colección llamada “actors” realizando la fase $unwind por cast, no se debe agrupar de nuevo y debe contener todas las claves (title,year,cast,genres). Después, contar cuantos documentos existen en la nueva colección.

```
var fase1 = { $unwind: "$cast" }
var fase2 = { $project: { _id: 0, title: 1, year: 1, cast: 1, genres: 1 } }
var fase3 = { $out: "actors" }

db.Movies.aggregate([fase1, fase2, fase3])

db.actors.find({}, { _id: 0 })

db.actors.count()
```



actors | 0.032 s | 83.224 Docs

500 | p. 1 | 1 - 500 | Table

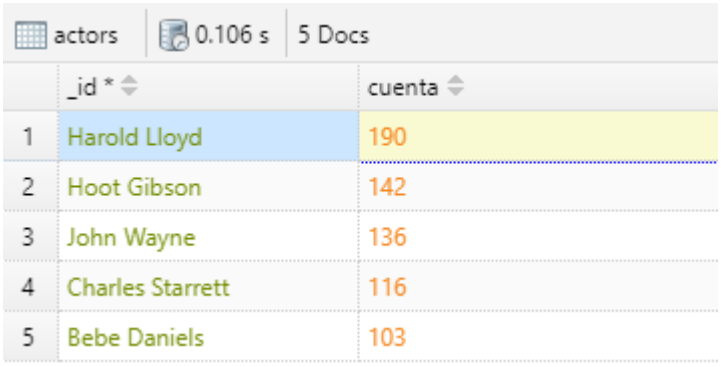
	title	year	cast	genres
1	Caught	1900	Undefined	Array[1]
2	After Dark in Central Park	1900	Undefined	Array[1]
3	Buffalo Bill's Wild West Parad	1900	Undefined	Array[1]
4	The Enchanted Drawing	1900	Undefined	Array[1]
5	Clowns Spinning Hats	1900	Undefined	Array[1]
6	Capture of Boer Battery by British	1900	Undefined	Array[2]
7	Feeding Sea Lions	1900	Paul Boyton	Array[1]
8	How to Make a Fat Wife Out of Two Lean Ones	1900	Undefined	Array[1]
9	Searching Ruins on Broadway, Galveston, for Dead Bodies	1900	Undefined	Array[1]
10	The Tribulations of an Amateur Photographer	1900	Undefined	Array[1]
11	Trouble in Hogan's Alley	1900	Undefined	Array[1]
12	Boarding School Girls' Pajama Parade	1900	Undefined	Array[1]

0.024 s

1 83224

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 previamente filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, sólo que filtramos para que no aparezcan.

```
var fase1 = { $match: { cast: { $ne: "Undefined" } } }
var fase2 = { $group: { _id: "$cast", cuenta: { $sum: 1 } } }
var fase3 = { $sort: { cuenta: -1 } }
var fase4 = { $limit: 5 }

db.actors.aggregate([fase1, fase2, fase3, fase4])
```

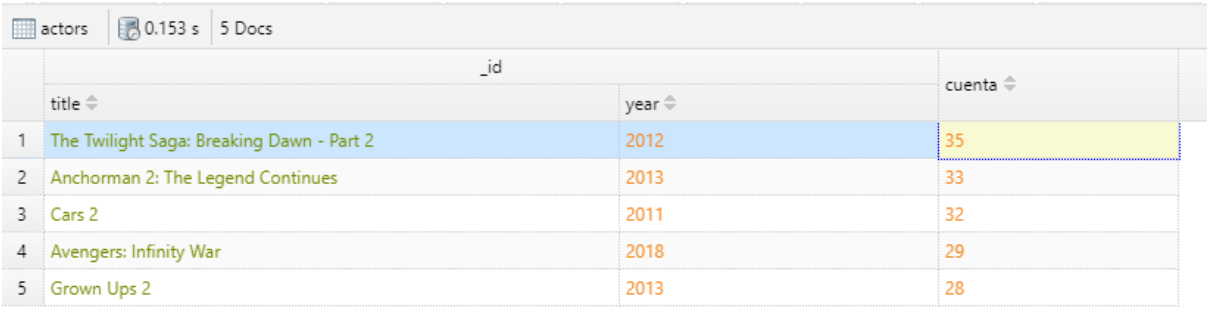


actors | 0.106 s | 5 Docs

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

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 fase1 = { $group: {_id: {title: "$title", year: "$year"}, cuenta: {$sum: 1}}}
var fase2 = { $sort: { cuenta: -1 } }
var fase3 = { $limit: 5 }

db.actors.aggregate([fase1, fase2, fase3])
```

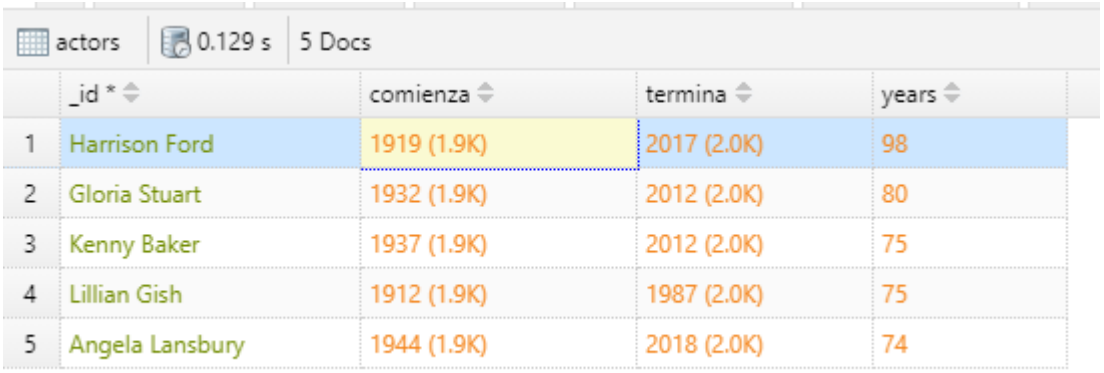


actors | 0.153 s | 5 Docs

	_id		cuenta
	title	year	
1	The Twilight Saga: Breaking Dawn - Part 2	2012	35
2	Anchorman 2: The Legend Continues	2013	33
3	Cars 2	2011	32
4	Avengers: Infinity War	2018	29
5	Grown Ups 2	2013	28

19. Sobre actors (nueva colección), mostrar los 5 actores cuya carrera haya sido la más larga. Para ello, se debe mostrar cuándo comenzó 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, sólo que filtramos para que no aparezcan.

```
var fase1 = {$match: { cast: { $ne: "Undefined" } } }
var fase2 = {$group: {_id: "$cast", comienza: { $min: "$year"}, termina: { $max: "$year"} } }
var fase3 = {$project: {_id: 1, comienza: 1, termina: 1, years: { $subtract: ["$termina", "$comienza"]}}}
var fase4 = { $sort: { years: -1 } }
var fase5 = { $limit: 5 }

db.actors.aggregate([fase1, fase2, fase3, fase4, fase5])
```



actors | 0.129 s | 5 Docs

	_id *	comienza	termina	years
1	Harrison Ford	1919 (1.9K)	2017 (2.0K)	98
2	Gloria Stuart	1932 (1.9K)	2012 (2.0K)	80
3	Kenny Baker	1937 (1.9K)	2012 (2.0K)	75
4	Lillian Gish	1912 (1.9K)	1987 (2.0K)	75
5	Angela Lansbury	1944 (1.9K)	2018 (2.0K)	74

*Hay un error en la base de datos que data películas de Harrison Ford dando un valor de duración de su carrera un tanto irreal. Había pensado en limitar la edad mínima a 1930 por ejemplo pero entonces estaría quitando información a otros actores por lo que no sería lógico.

20. Sobre actors (nueva colección), Guardar en nueva colección llamada "genres" realizando la fase $unwind por genres, no se debe agrupar de nuevo y debe contener todas las claves (title,year,cast,genres). Después, contar cuantos documentos existen en la nueva colección.

```
var fase1 = { $unwind: "$genres" }
var fase2 = { $project: { _id: 0, title: 1, year: 1, cast: 1, genres: 1 } }
var fase3 = { $out: "genres" }

db.actors.aggregate([fase1, fase2, fase3])

db.genres.find({}, { _id: 0 })

db.genres.count()
```

genres | 0.028 s | 104.950 Docs

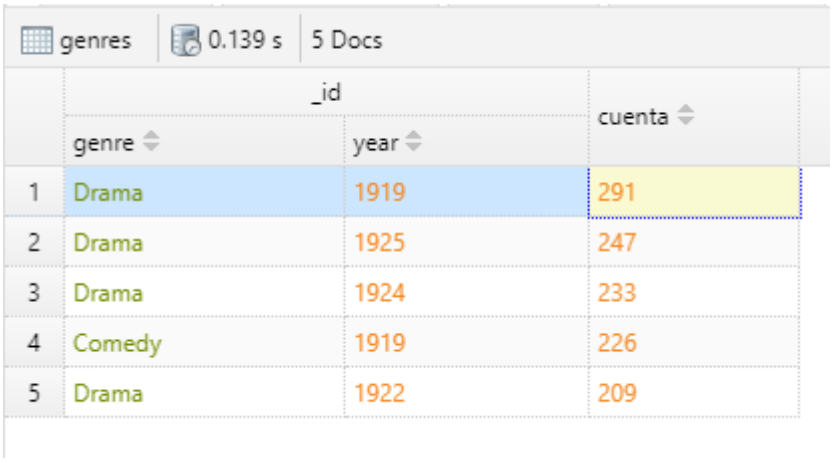
	title	year	cast	genres
1	Caught	1900	Undefined	Undefined
2	After Dark in Central Park	1900	Undefined	Undefined
3	Buffalo Bill's Wild West Parad	1900	Undefined	Undefined
4	The Enchanted Drawing	1900	Undefined	Undefined
5	Clowns Spinning Hats	1900	Undefined	Undefined
6	Capture of Boer Battery by British	1900	Undefined	Short
7	Capture of Boer Battery by British	1900	Undefined	Documentary
8	Feeding Sea Lions	1900	Paul Boyton	Undefined
9	How to Make a Fat Wife Out of Two Lean Ones	1900	Undefined	Comedy
10	Searching Ruins on Broadway, Galveston, for Dead Bodies	1900	Undefined	Undefined
11	The Tribulations of an Amateur Photographer	1900	Undefined	Undefined
12	Trouble in Hogan's Alley	1900	Undefined	Comedy

0.019 s

1	104950

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. Importante! Se necesita previamente filtrar para descartar aquellos genres llamados "Undefined". Aclarar que no se eliminan de la colección, sólo que filtramos para que no aparezcan.

```
var fase1 = { $match: { genres: { $ne: "Undefined" } } }
var fase2 = {$group: {_id: { genre: "$genres", year: "$year" }, titulos: { $addToSet: "$title" }}}
var fase3 = { $project: { _id: 1, cuenta: { $size: "$titulos" } } }
var fase4 = { $sort: { cuenta: -1 } }
var fase5 = { $limit: 5 }

db.genres.aggregate([fase1, fase2, fase3, fase4, fase5])
```



genres | 0.139 s | 5 Docs

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

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 tanto el número de géneros diferentes que ha interpretado como los diferentes géneros además del nombre del actor. Importante! Se necesita previamente filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, sólo que filtramos para que no aparezcan.

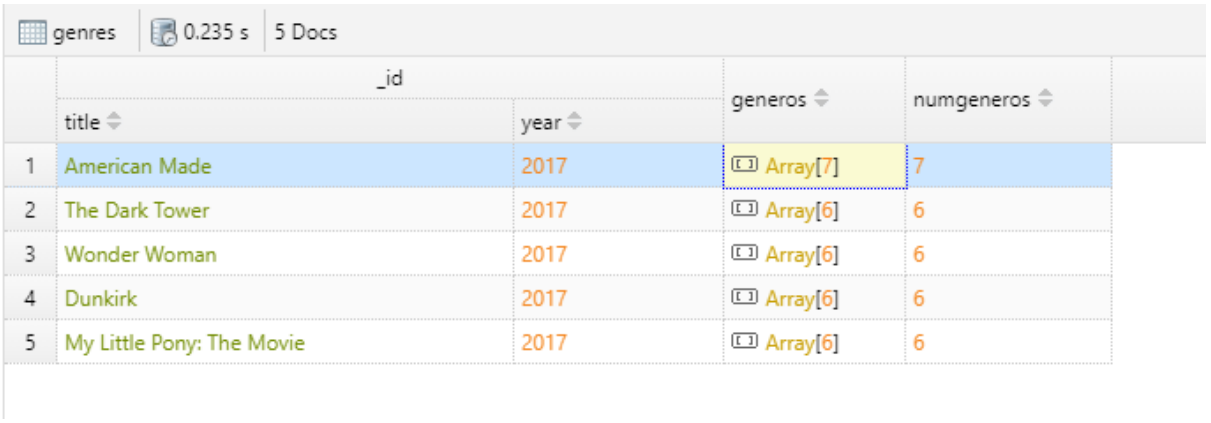
```
var fase1 = { $match: { cast: { $ne: "Undefined" }, genres: { $ne: "Undefined" } } }
var fase2 = { $group: {_id: "$cast", generos: { $addToSet: "$genres" }}}
var fase3 = { $project: {_id: 1, numgeneros: { $size: "$generos" }, generos: 1}}
var fase4 = { $sort: { numgeneros: -1 } }
var fase5 = { $limit: 5 }

db.genres.aggregate([fase1, fase2, fase3, fase4, fase5])
```

genres | 0.182 s | 5 Docs

	_id *	generos	numgeneros
1	Dennis Quaid	Array[20]	20
2	Helen Mirren	Array[18]	18
3	James Coburn	Array[18]	18
4	Colin Farrell	Array[18]	18
5	Gene Hackman	Array[18]	18

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. Importante! Se necesita previamente filtrar para descartar aquellos genres llamados "Undefined". Aclarar que no se eliminan de la colección, sólo que filtramos para que no aparezcan.

```
var fase1 = { $match: { genres: { $ne: "Undefined" } } }
var fase2 = {$group: {_id: { title: "$title", year: "$year" },generos: { $addToSet: "$genres" } } }
var fase3 = {$project: {_id: 1, generos: 1, numgeneros: { $size: "$generos" } } }
var fase4 = { $sort: { numgeneros: -1 } }
var fase5 = { $limit: 5 }

db.genres.aggregate([fase1, fase2, fase3, fase4, fase5])
```



genres | 0.235 s | 5 Docs

	_id		generos	numgeneros
	title	year		
1	American Made	2017	Array[7]	7
2	The Dark Tower	2017	Array[6]	6
3	Wonder Woman	2017	Array[6]	6
4	Dunkirk	2017	Array[6]	6
5	My Little Pony: The Movie	2017	Array[6]	6

24. Género más dominante por cada época.

```
var fase1= {$match:{genres:{$ne:"Undefined"}}}
var fase2= {$addFields:{decada:{$subtract:["$year",{$mod:["$year",10]}]}}}
var fase3=
{$group:{_id:{decada:"$decada",genre:"$genres"},pelis:{$addToSet:{title:"$title",year:
"$year"}}}}
var fase4= {$project:{_id:1,cuenta:{$size:"$pelis"}}}
var fase5= {$sort:{"_id.decada":1,cuenta:-1}}
var fase6= {$group:{_id:"$_id.decada",top:{$first:"$$ROOT"}}}
var
fase7={$project:{_id:0,decada:"$_id",genero_top:"$top._id.genre",num_peliculas:"$to
p.cuenta"}}
var fase8= {$sort:{decada:-1}}


db.genres.aggregate([fase1,fase2,fase3,fase4,fase5,fase6,fase7, fase8])
```

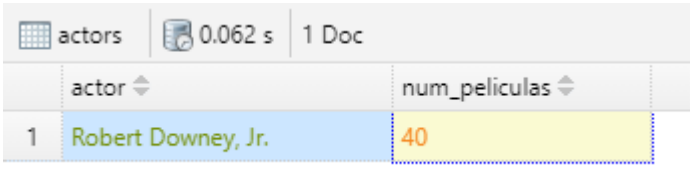
genres | 0.338 s | 12 Docs

	decada	genero_top	num_peliculas
1	2010 (2.0K)	Drama	647
2	2000 (2.0K)	Comedy	767
3	1990 (2.0K)	Drama	837
4	1980 (2.0K)	Comedy	603
5	1970 (2.0K)	Drama	417
6	1960 (2.0K)	Drama	407
7	1950 (1.9K)	Western	698
8	1940 (1.9K)	Comedy	918
9	1930 (1.9K)	Drama	1627 (1.6K)
10	1920 (1.9K)	Drama	1519 (1.5K)
11	1910 (1.9K)	Drama	587
12	1900 (1.9K)	Short	18

25. Número de películas totales de Robert Downey, Jr. y sus generos mas representativos a lo largo de su carrera.

Número total de peliculas:

```
var fase1={$match:{"cast": "Robert Downey, Jr."}}
var fase2={$group:{_id:"$cast",pelis:{$addToSet:{title:"$title",year:"$year"}}}}
var fase3={$project:{_id:0,actor:"$_id",num_peliculas:{$size:"$pelis"}}}

db.actors.aggregate([fase1,fase2,fase3])
```



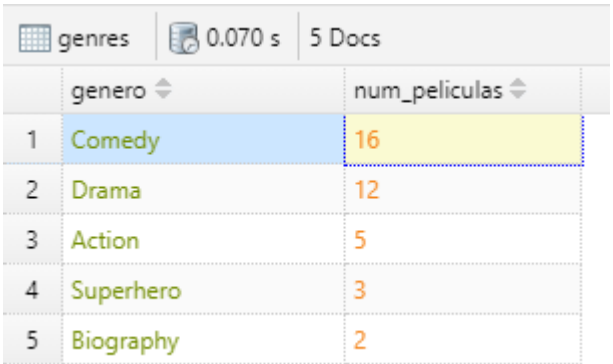
actors | 0.062 s | 1 Doc

	actor	num_peliculas
1	Robert Downey, Jr.	40

5 géneros más representativos de la carrera de Robert:

.

```
var fase1={$match:{cast:"Robert Downey, Jr.",genres:{$ne:"Undefined"}}}
var fase2={$group:{_id:"$genres",pelis:{$addToSet:{title:"$title",year:"$year"}}}}
var fase3={$project:{_id:0,genero:"$_id",num_peliculas:{$size:"$pelis"}}}
var fase4={$sort:{num_peliculas:-1}}
var fase5={$limit:5}

db.genres.aggregate([fase1,fase2,fase3,fase4,fase5])
```



genres | 0.070 s | 5 Docs

	genero	num_peliculas
1	Comedy	16
2	Drama	12
3	Action	5
4	Superhero	3
5	Biography	2

26. Evolución del cine Science Fiction por décadas (1900-2010)

```
var fase1={$match:{genres:"Science Fiction"}}
var fase2={$addFields:{decada:{$subtract:["$year",{$mod:["$year",10]}]}}}
var fase3={$group:{_id:"$decada",pelis:{$addToSet:{title:"$title",year:"$year"}}}}
var fase4={$project:{_id:0,decada:"$_id",num_peliculas:{$size:"$pelis"}}}
var fase5={$sort:{decada:1}}

db.genres.aggregate([fase1,fase2,fase3,fase4,fase5])
```

genres | 0.098 s | 10 Docs

	decada	num_peliculas
1	1920 (1.9K)	2
2	1930 (1.9K)	16
3	1940 (1.9K)	15
4	1950 (1.9K)	141
5	1960 (2.0K)	64
6	1970 (2.0K)	58
7	1980 (2.0K)	124
8	1990 (2.0K)	118
9	2000 (2.0K)	118
10	2010 (2.0K)	124