

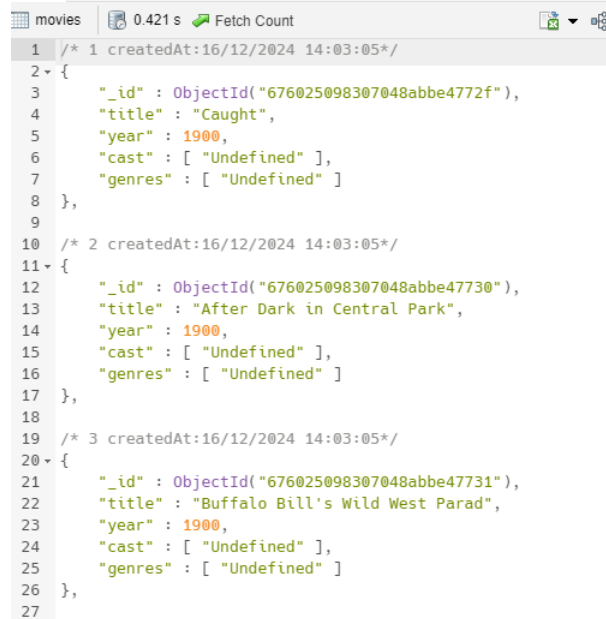
MÓDULO 3: BASES DE DATOS NO SQL



Alumno: Pablo Pérez Calvo

1. Analizar con find la colección.

```
1 // 1.  
2 db.movies.find()
```

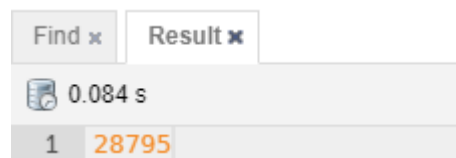


```
1 /* 1 createdAt:16/12/2024 14:03:05*/  
2 {  
3   "_id" : ObjectId("676025098307048abbe4772f"),  
4   "title" : "Caught",  
5   "year" : 1900,  
6   "cast" : [ "Undefined" ],  
7   "genres" : [ "Undefined" ]  
8 },  
9  
10 /* 2 createdAt:16/12/2024 14:03:05*/  
11 {  
12   "_id" : ObjectId("676025098307048abbe47730"),  
13   "title" : "After Dark in Central Park",  
14   "year" : 1900,  
15   "cast" : [ "Undefined" ],  
16   "genres" : [ "Undefined" ]  
17 },  
18  
19 /* 3 createdAt:16/12/2024 14:03:05*/  
20 {  
21   "_id" : ObjectId("676025098307048abbe47731"),  
22   "title" : "Buffalo Bill's Wild West Parad",  
23   "year" : 1900,  
24   "cast" : [ "Undefined" ],  
25   "genres" : [ "Undefined" ]  
26 },  
27
```

Figura 1: Resultado 1

2. Contar cuántos documentos (películas) tiene cargado.

```
1 // 2.  
2 db.movies.countDocuments()
```



```
Find x Result x  
0.084 s  
1 28795
```

Figura 2: Resultado 2

3. Insertar una película.

```
1 // 3.  
2 db.movies.insertOne({title: "Once pares de botas", year: 1954, cast  
  : [], genres: []})
```



Figura 3: Resultado 3

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

```
1 // 4.  
2 db.movies.deleteOne({title: "Once pares de botas"})
```



Figura 4: Resultado 4

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

```
1 // 5.  
2 db.movies.find({cast: "and"}).count()
```

Find x	Result x
0.456 s	
1	93

Figura 5: Resultado 5

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.

```
1 // 6.
2 db.movies.updateMany({cast: "and"}, { $pull: { cast: "and" } })
```

0.424 s	
1	{
2	"acknowledged" : true,
3	"matchedCount" : 93,
4	"modifiedCount" : 93
5	}

Figura 6: Resultado 6

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

```
1 // 7.
2 db.movies.countDocuments({cast: {$size: 0 }})
```

Result x	Result (1) x
0.245 s	
1	986

Figura 7: Resultado 7

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”].

```
1 // 8.  
2 db.movies.updateMany({cast: {$size: 0 }}, { $set: {cast: ["  
    Undefined"]}}})
```



Figura 8: Resultado 8

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

```
1 // 9.  
2 db.movies.countDocuments({genres: {$size: 0 }})
```

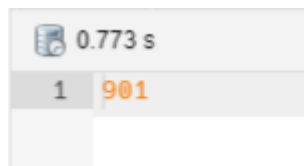


Figura 9: Resultado 9

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”].

```
1 // 10.  
2 db.movies.updateMany({genres: {$size: 0 }}, { $set: {genres: ["  
    Undefined"]}}})
```

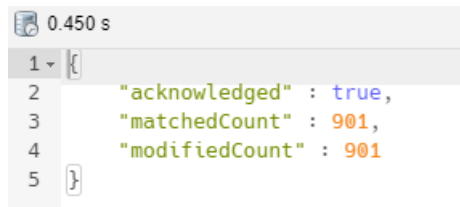


Figura 10: Resultado 10

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

```
1 // 11.
2 db.movies.find({}, { _id: 0, year: 1 }).sort({ year: -1 }).limit(1)
```

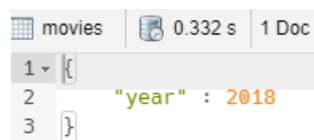


Figura 11: Resultado 11

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.

```
1 // 12.
2 var maxYear = db.movies.find({}, { _id: 0, year: 1 }).sort({ year:
3   -1 }).limit(1).toArray()[0].year
4 var query1 = {year: {$gte: maxYear-19}}
5 var fase1 = {$match: query1}
6 var query2 = {_id: null, title: {$sum: 1}}
7 var fase2 = {$group: query2}
8 var etapas = [fase1, fase2]
9 db.movies.aggregate( etapas)
```

movies	1.224 s	1 Doc
1	{	
2	"_id" : null,	
3	"title" : 4787	
4	}	

Figura 12: Resultado 12

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.

```

1 // 13.
2 var query1 = {year: {$gte: 1960, $lte:1969}}
3 var fase1 = {$match: query1}
4 var query2 = {_id: null, title: {$sum: 1}}
5 var fase2 = {$group: query2}
6 var etapas = [fase1, fase2]
7 db.movies.aggregate( etapas)

```

1	{	
2	"_id" : null,	
3	"title" : 1414	
4	}	

Figura 13: Resultado 13

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.

```

1 // 14.
2 var query1 = {_id: "$year", pelis: {$sum: 1}}
3 var fase1 = {$group: query1}
4 var query2 = {"pelis":-1}
5 var fase2 = {$sort: query2}
6 var fase3 = {$limit:1}
7 var etapas = [fase1, fase2, fase3]
8 db.movies.aggregate( etapas)

```

```

1 {
2   "_id" : 1919,
3   "pelis" : 634
4 }

```

Figura 14: Resultado 14

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.

```

1 // 15.
2 var query1 = {_id: "$year", pelis: {$sum: 1}}
3 var fase1 = {$group: query1}
4 var query2 = {"pelis":1}
5 var fase2 = {$sort: query2}
6 var fase3 = {$limit:3}
7 var etapas = [fase1, fase2, fase3]
8 db.movies.aggregate(etapas)

```

```

movies 0.441 s 3 Docs
1 /* 1 */
2 {
3   "_id" : 1906,
4   "pelis" : 7
5 },
6
7 /* 2 */
8 {
9   "_id" : 1907,
10  "pelis" : 7
11 },
12
13 /* 3 */
14 {
15   "_id" : 1902,
16   "pelis" : 7
17 }

```

Figura 15: Resultado 15

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

```
1 // 16.
2 var fase1 = {$unwind: "$cast"}
3 var query2 = {"_id": 0}
4 var fase2 = {$project: query2}
5 var fase3 = { $out: "actors"}
6 var etapas = [fase1,fase2, fase3]
7 db.movies.aggregate(etapas)
8
9 db.actors.countDocuments()
```

Aggregate x	Aggregate (1) x	Aggregate (2) x	Aggregat
2.517 s			
1	83224		

Figura 16: Resultado 16

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.

```
1 // 17.
2 var query1 = {cast: {$ne: "Undefined"}}
3 var fase1 = {$match: query1}
4 var query2 = { _id: "$cast",  cuenta: {$sum: 1}}
5 var fase2 = {$group: query2}
6 var query3 = {cuenta: -1}
7 var fase3 = {$sort: query3}
8 var fase4 = {$limit: 5}
9 etapas = [fase1, fase2, fase3, fase4]
10 db.actors.aggregate(etapas)
```

```

1  /* 1 */
2  {
3    "_id" : "Harold Lloyd",
4    "cuenta" : 190
5  },
6
7  /* 2 */
8  {
9    "_id" : "Hoot Gibson",
10   "cuenta" : 142
11 },
12
13 /* 3 */
14 {
15   "_id" : "John Wayne",
16   "cuenta" : 136
17 },
18
19 /* 4 */
20 {
21   "_id" : "Charles Starrett",
22   "cuenta" : 116
23 },
24
25 /* 5 */
26 {
27   "_id" : "Bebe Daniels",
28   "cuenta" : 103
29 }

```

Figura 17: Resultado 17

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

```

1 // 18.
2 var query1 = { _id: {title:"$title", year: "$year"}, cuenta: {$sum
3   : 1}}
4 var fase1 = {$group: query1}
5 var query2 = {cuenta: -1}
6 var fase2 = {$sort: query2}
7 var fase3 = {$limit: 5}
8 etapas = [fase1, fase2, fase3]
9 db.actors.aggregate(etapas)

```

```

1  /* 1 */
2  {
3    "_id" : {
4      "title" : "The Twilight Saga: Breaking Dawn - Part 2",
5      "year" : 2012
6    },
7    "cuenta" : 35
8  },
9
10 /* 2 */
11 {
12   "_id" : {
13     "title" : "Anchorman 2: The Legend Continues",
14     "year" : 2013
15   },
16   "cuenta" : 33
17 },
18
19 /* 3 */
20 {
21   "_id" : {
22     "title" : "Cars 2",
23     "year" : 2011
24   },
25   "cuenta" : 32
26 },
27
28 /* 4 */
29 {
30   "_id" : {
31     "title" : "Avengers: Infinity War",
32     "year" : 2018
33   },
34   "cuenta" : 29
35 },
36
37 /* 5 */
38 {
39   "_id" : {
40     "title" : "Grown Ups 2",
41     "year" : 2013
42   },
43   "cuenta" : 28
44 }

```

Figura 18: Resultado 18

19. Sobre actores (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.

```

1 // 19.
2 var query1 = {cast: {$ne: "Undefined"}}
3 var fase1 = {$match: query1}
4 var query2 = { _id: "$cast", comienza: {$min: "$year"}, termina: {
5   $max: "$year"}}
6 var fase2 = {$group: query2}
7 var query3 = {anos: {$add:[ { $subtract: ["$termina", "$comienza
8   " ]},1]}}
9 var fase3 = {$addFields: query3}
10 var query4 = {anos: -1}
11 var fase4 = {$sort: query4}
12 var fase5 = {$limit: 5}
13 etapas = [fase1, fase2, fase3, fase4, fase5]
14 db.actors.aggregate(etapas)

```

```

/* 1 */
{
  "_id" : "Harrison Ford",
  "comienza" : 1919,
  "termina" : 2017,
  "anos" : 99
},
/* 2 */
{
  "_id" : "Gloria Stuart",
  "comienza" : 1932,
  "termina" : 2012,
  "anos" : 81
},
/* 3 */
{
  "_id" : "Kenny Baker",
  "comienza" : 1937,
  "termina" : 2012,
  "anos" : 76
},
/* 4 */
{
  "_id" : "Lillian Gish",
  "comienza" : 1912,
  "termina" : 1987,
  "anos" : 76
},
/* 5 */
{
  "_id" : "Angela Lansbury",
  "comienza" : 1944,
  "termina" : 2018,
  "anos" : 75
}

```

Figura 19: Resultado 19

20. Sobre actores (nueva colección), Guardar en nueva colección llamada "genres" realizando la fase \$unwind por genres. Después, contar cuantos documentos existen en la nueva colección.

```

1 // 20.
2 var fase1 = {$unwind: "$genres"}
3 var query2 = {"_id": 0}
4 var fase2 = {$project: query2}
5 var fase3 = { $out: "genres"}
6 var etapas = [fase1,fase2, fase3]
7 db.actors.aggregate(etapas)
8
9 db.genres.countDocuments()

```


	3.798 s
1	104950

Figura 20: Resultado 20

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.

```

1 // 21.
2 var query1 = { _id: {year:"$year", genre: "$genres"}, pelis: {
3   $addToSet: "$title"}}
4 var fase1 = {$group: query1}
5 var query2 = {_id: 1, pelis: {$size: "$pelis"}}
6 var fase2 = {$project: query2}
7 var query3 = {pelis: -1}
8 var fase3 = {$sort: query3}
9 var fase4 = {$limit: 5}
10 var etapas = [fase1, fase2, fase3,fase4]
    db.genres.aggregate(etapas)

```

```

1  /* 1 */
2  {
3    "_id" : {
4      "year" : 1919,
5      "genre" : "Drama"
6    },
7    "pelis" : 291
8  },
9
10 /* 2 */
11 {
12   "_id" : {
13     "year" : 1925,
14     "genre" : "Drama"
15   },
16   "pelis" : 247
17 },
18
19 /* 3 */
20 {
21   "_id" : {
22     "year" : 1924,
23     "genre" : "Drama"
24   },
25   "pelis" : 233
26 },
27
28 /* 4 */
29 {
30   "_id" : {
31     "year" : 1919,
32     "genre" : "Comedy"
33   },
34   "pelis" : 226
35 },
36
37 /* 5 */
38 {
39   "_id" : {
40     "year" : 1922,
41     "genre" : "Drama"
42   },
43   "pelis" : 209
44 }

```

Figura 21: Resultado 21

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, sólo que filtramos para que no aparezcan.

```

1  // 22.
2  var query1 = {cast: {$ne: "Undefined"}}
3  var fase1 = {$match: query1}
4  var query2 = { _id: "$cast",  generos: {$addToSet: "$genres"}}
5  var fase2 = {$group: query2}
6  var query3 = {_id: 1, numgeneros: {$size: "$generos"}, generos: 1}
7  var fase3 = {$project: query3}
8  var query4 = {numgeneros: -1}
9  var fase4 = {$sort: query4}
10 var fase5 = {$limit: 5}
11 var etapas = [fase1, fase2, fase3, fase4, fase5]
12 db.genres.aggregate(etapas)

```

```

/* 1 */
{
  "_id" : "Dennis Quaid",
  "generos" : [
    "Drama",
    "Sports",
    "Fantasy",
    "Science Fiction",
    "Animated",
    "Satire",
    "Dance",
    "Biography",
    "Comedy",
    "Disaster",
    "Crime",
    "Family",
    "Action",
    "Western",
    "Musical",
    "Adventure",
    "Suspense",
    "Thriller",
    "Horror",
    "Romance"
  ],
  "numgeneros" : 20
},

/* 2 */
{
  "_id" : "Michael Caine",
  "generos" : [
    "Horror",
    "Romance",
    "Science Fiction",
    "Drama",
    "Animated",
    "Biography",
    "Comedy",
    "Disaster",
    "Action",
    "Superhero",
    "Mystery",
    "Crime",
    "Family",
    "Undefined",
    "Thriller",
    "Suspense",
    "Adventure",
    "Spy",
    "War"
  ],
  "numgeneros" : 19
},

/* 3 */
{
  "_id" : "James Mason",
  "generos" : [
    "Romance",
    "Science Fiction",
    "Drama",
    "Fantasy",
    "Animated",
    "Biography",
    "Comedy",
    "Action",
    "Mystery",
    "Crime",
    "Musical",
    "Undefined",
    "Western",
    "Noir",
    "Short",
    "Thriller",
    "Adventure",
    "Suspense",
    "War"
  ],
  "numgeneros" : 19
},

/* 4 */
{
  "_id" : "Helen Mirren",
  "generos" : [
    "Spy",
    "Fantasy",
    "Drama",
    "Science Fiction",
    "Animated",
    "Biography",
    "Comedy",
    "Erotic",
    "Mystery",
    "Crime",
    "Action",
    "Political",
    "Family",
    "Adventure",
    "Historical",
    "Romance",
    "Horror",
    "Thriller"
  ],
  "numgeneros" : 18
},

/* 5 */
{
  "_id" : "Gene Hackman",
  "generos" : [
    "Spy",
    "Drama",
    "Sports",
    "Science Fiction",
    "Animated",
    "Biography",
    "Disaster",
    "Comedy",
    "Crime",
    "Superhero",
    "Action",
    "Mystery",
    "Western",
    "Noir",
    "Adventure",
    "Suspense",
    "Thriller",
    "War"
  ],
  "numgeneros" : 18
}

```

Figura 22: Resultado 22

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.

```

1 // 23.
2 var query1 = {cast: {$ne: "Undefined"}}
3 var fase1 = {$match: query1}
4 var query2 = { _id: {title: "$title", year: "$year"},  generos: {
5   $addToSet: "$genres"}}
6 var fase2 = {$group: query2}
7 var query3 = { _id: 1, numgeneros: {$size: "$generos"}, generos:1}
8 var fase3 = {$project: query3}
9 var query4 = {numgeneros: -1}
10 var fase4 = {$sort: query4}
11 var fase5 = {$limit: 5}
12 var etapas = [fase1, fase2, fase3, fase4, fase5]
13 db.genres.aggregate(etapas)

```

```

/* 1 */
{
  "_id" : {
    "title" : "American Made",
    "year" : 2017
  },
  "generos" : [
    "Action",
    "Historical",
    "Thriller",
    "Comedy",
    "Biography",
    "Crime",
    "Drama"
  ],
  "numgeneros" : 7
},

/* 2 */
{
  "_id" : {
    "title" : "The Dark Tower",
    "year" : 2017
  },
  "generos" : [
    "Western",
    "Science Fiction",
    "Action",
    "Adventure",
    "Fantasy",
    "Horror"
  ],
  "numgeneros" : 6
},

/* 3 */
{
  "_id" : {
    "title" : "My Little Pony: The Movie",
    "year" : 2017
  },
  "generos" : [
    "Animated",
    "Comedy",
    "Musical",
    "Family",
    "Fantasy",
    "Adventure"
  ],
  "numgeneros" : 6
},

/* 4 */
{
  "_id" : {
    "title" : "Wonder Woman",
    "year" : 2017
  },
  "generos" : [
    "Superhero",
    "Fantasy",
    "War",
    "Adventure",
    "Drama",
    "Action"
  ],
  "numgeneros" : 6
},

/* 5 */
{
  "_id" : {
    "title" : "Dunkirk",
    "year" : 2017
  },
  "generos" : [
    "Thriller",
    "Drama",
    "Action",
    "War",
    "Historical",
    "Adventure"
  ],
  "numgeneros" : 6
}

```

Figura 23: Resultado 23

24. Sobre genres, mostrar el actor que mas ha participado en las películas de cada uno de los géneros. El actor mas representativo de cada uno de ellos. No se deben contar los genres y actores "Undefined"

```

1 // 24.
2 var query1 = { $and: [ { cast: { $ne: "Undefined" } }, { genres: {
3   $ne: "Undefined" } }]}
4 var fase1 = {$match: query1}
5 var query2 = { _id: {genre: "$genres", actor: "$cast"}, apariciones:
6   {$sum:1}}

```

```

5 var fase2 = {$group: query2};
6 var fase3 = {$sort: {apariciones: -1}}
7 var query3 = {_id: "$_id.genre", actoresunicos: { $addToSet: "$_id.
  actor" }, actor_mas_participativo: { $first: "$_id.actor" },
8   maxapariciones: { $max: "$apariciones" }}
9 var fase4 = {$group: query3};
10 var query4 = {_id: 1, actor_mas_participativo: 1, maxapariciones:
  1}
11 var fase5 = {$project: query4};
12 var query5 = {maxapariciones: -1};
13 var fase6 = {$sort: query5};
14 var etapas = [fase1, fase2, fase3, fase4, fase5,fase6];
15 db.genres.aggregate(etapas);

```

_id *	actor_mas_participativo	maxapariciones
1 Comedy	Harold Lloyd	186
2 Western	Hoot Gibson	131
3 Animated	Tom and Jerry	85
4 Short	The Three Stooges	64
5 Drama	Bette Davis	56
6 Musical	Bing Crosby	41
7 Adventure	Johnny Weissmuller	29
8 Horror	Boris Karloff	28
9 Action	Jean-Claude Van Damme	22
10 Crime	Edward G. Robinson	21
11 Mystery	Warner Oland	16
12 Romance	Gary Cooper	15
13 Thriller	Nicolas Cage	14
14 Noir	Edward G. Robinson	14
15 War	John Wayne	12
16 Fantasy	Robbie Coltrane	9
17 Science Fiction	William Shatner	9
18 Superhero	Chris Evans	8
19 Biography	Ed Harris	8
20 Family	Maggie Smith	7

_id *	actor_mas_participativo	maxapariciones
21 Documentary	Iraq War	5
22 Spy	Dean Martin	5
23 Historical	Tyrone Power	5
24 Suspense	Cary Grant	4
25 Disaster	George Kennedy	4
26 Erotic	Jamie Gillis	4
27 Sports	Elyse Knox	4
28 Satire	Parker Posey	3
29 Political	Jeffrey Wright	3
30 Dance	Adam G. Sevani	3
31 Martial Arts	Jackie Chan	3
32 Performance	Jonas Brothers	2
33 Teen	Sky Hirschhorn	2
34 Slasher	Robert Englund	2
35 Live Action	Fred Armisen	2
36 Supernatural	Katie Featherston	2
37 Legal	Marshall Bell	1
38 Silent	Ralph Ince	1
39 Independent	Sarah Prikryl	1
40 Sport	John Wayne	1

Figura 24: Resultado 24

25. Determina el top 3 de géneros con más películas en los últimos 10 años, registrados en la base de datos, y compáralos con el top 3 de géneros de los 10 primeros años registrados

```
1 //25.
2 var Ultimoano = 2018
3 var Primerano = 1900
4 var Primeradecada = { $gte: Primerano, $lte: Primerano + 9 }
5 var Ultimadecada = { $gte: Ultimoano - 9, $lte: Ultimoano }
6
7 var query1 = { genres: { $ne: "Undefined" } }
8 var fase1 = { $match: query1 }
9
10 // Pipeline para los primeros 10 años
11 var query2a = { year: Primeradecada }
12 var fase2a = { $match: query2a }
13 var query3a = { _id: "$genres", Numpelis: { $sum: 1 } }
14 var fase3a = { $group: query3a }
15 var query4a = { Numpelis: -1 }
16 var fase4a = { $sort: query4a }
17 var fase5a = { $limit: 3 }
18
19 //Pipeline para los ultimos 10 años
20 var query2b = { year: Ultimadecada }
21 var fase2b = { $match: query2b }
22 var query3b = { _id: "$genres", Numpelis: { $sum: 1 } }
23 var fase3b = { $group: query3b }
24 var query4b = { Numpelis: -1 }
25 var fase4b = { $sort: query4b }
26 var fase5b = { $limit: 3 }
27
28 // Ejecutar ambos pipelines usando $facet
29 var etapas = [
30     fase1,{
31         $facet: {
32             Primeradecada: [fase2a, fase3a, fase4a, fase5a],
33             Ultimadecada: [fase2b, fase3b, fase4b, fase5b] } } ]
34
35 db.genres.aggregate(etapas)
```

```

1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- 16- 17- 18- 19- 20-
    {
      "_id" : "Comedy",
      "Numpelis" : 22
    },
    {
      "_id" : "Short",
      "Numpelis" : 19
    },
    {
      "_id" : "Drama",
      "Numpelis" : 12
    }
  ],
  "Ultimadecada" : [
    {
      "_id" : "Comedy",
      "Numpelis" : 4513
    },
    {
      "_id" : "Drama",
      "Numpelis" : 4805
    },
    {
      "_id" : "Action",
      "Numpelis" : 2380
    }
  ]
}

```

Figura 25: Resultado 25

26. Determinar los 5 géneros a los que pertenecen más películas que tengan en su título la palabra "war".

```

1 //26.
2 var query1 = { _id : { title : "$title", genre : "$genres"}}
3 var fase1 = { $group : query1}
4 // Filtramos por los titulos que contengan la palabra war
5 var query2 = {"_id.title" : { $regex : "war", $options : "i"}}
6 var fase2 = { $match : query2}
7 var query3 = { _id: "$_id.genre", total:{ $sum : 1}}
8 var fase3 = { $group : query3}
9 var etapas = [fase1, fase2, fase3]
10 db.genres.aggregate(etapas).sort({total: -1}).limit(5)

```

```

/* 1 */
{
  "_id" : "Drama",
  "total" : 61
},
/* 2 */
{
  "_id" : "Comedy",
  "total" : 50
},
/* 3 */
{
  "_id" : "Western",
  "total" : 30
},
/* 4 */
{
  "_id" : "Action",
  "total" : 24
},
/* 5 */
{
  "_id" : "War",
  "total" : 23
}

```

Figura 26: Resultado 26