NoSQL Activity

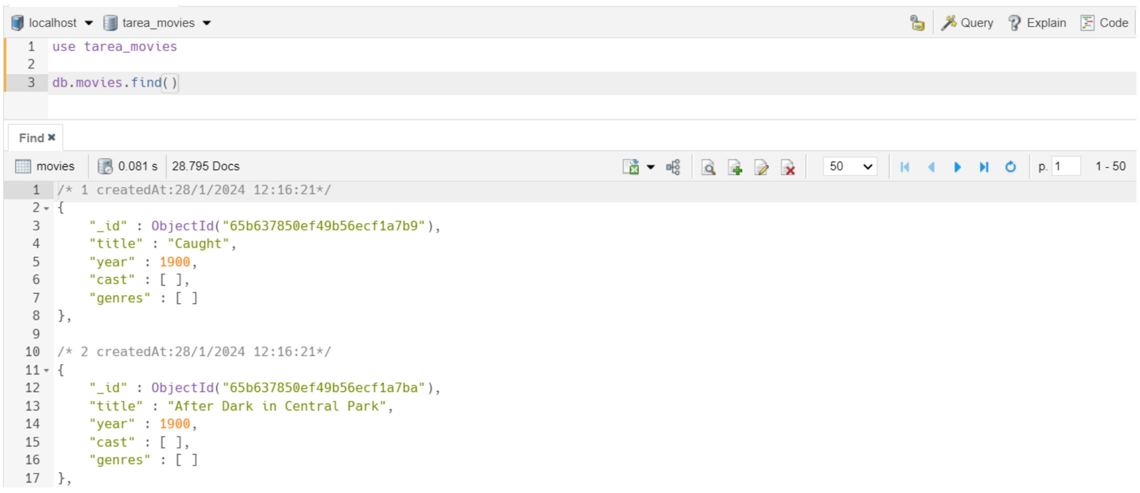
0. Import the 'movies' collection.

EXERCISE 1. Analysis using ‘find’.

Query:

```
use tarea_movies
db.movies.find()
```

Results:



localhost ▾ tarea_movies ▾ Query Explain Code

```
use tarea_movies

db.movies.find()
```

Find ✖

movies 0.081 s 28.795 Docs 50 p. 1 1 - 50

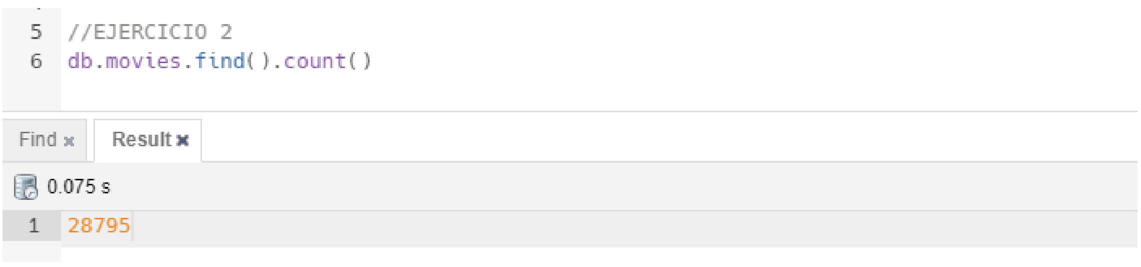
```
/* 1 createdAt:28/1/2024 12:16:21*/
{
    "_id" : ObjectId("65b637850ef49b56ecf1a7b9"),
    "title" : "Caught",
    "year" : 1900,
    "cast" : [ ],
    "genres" : [ ]
},

/* 2 createdAt:28/1/2024 12:16:21*/
{
    "_id" : ObjectId("65b637850ef49b56ecf1a7ba"),
    "title" : "After Dark in Central Park",
    "year" : 1900,
    "cast" : [ ],
    "genres" : [ ]
},
```

EXERCISE 2. Count documents in the collection.

Query:

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

Results:



```
//EJERCICIO 2
db.movies.find( ).count( )
```

Find ✖ Result ✖

0.075 s

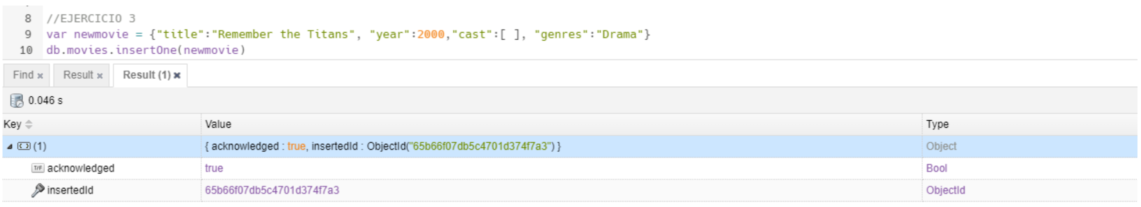
```
28795
```

EXERCISE 3. Insert a new document.

Query:

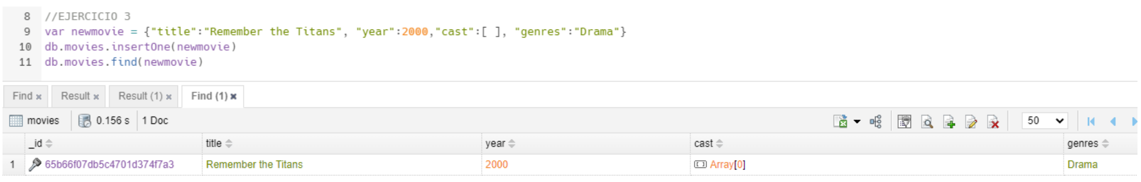
```
var newmovie = {"title":"Remember the Titans", "year":2000,"cast":[ ], "genres":"Drama"}
db.movies.insertOne(newmovie)
db.movies.find(newmovie)
```

Results:



```
//EJERCICIO 3
var newmovie = {"title":"Remember the Titans", "year":2000,"cast":[ ], "genres":"Drama"}
db.movies.insertOne(newmovie)
```

```
Find ×   Result ×   Result (1) ×
0.046 s
Key              Value                                                                     Type
(1)              { acknowledged : true, insertedId : ObjectId("65b66f07db5c4701d374f7a3") } Object
  acknowledged   true                                                                      Bool
  insertedId     65b66f07db5c4701d374f7a3                                                  ObjectId
```



```
//EJERCICIO 3
var newmovie = {"title":"Remember the Titans", "year":2000,"cast":[ ], "genres":"Drama"}
db.movies.insertOne(newmovie)
db.movies.find(newmovie)
```

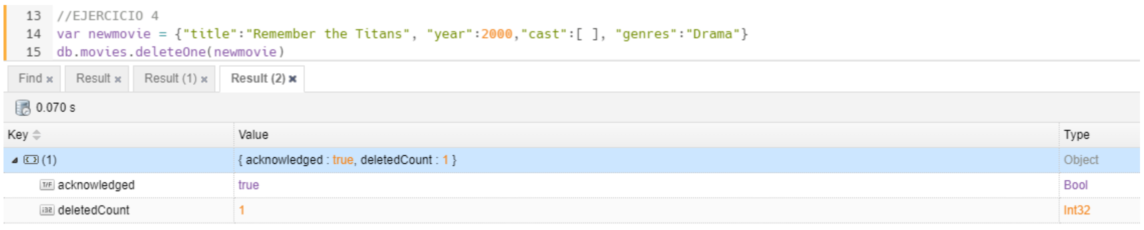
```
Find ×   Result ×   Result (1) ×   Find (1) ×
movies   0.156 s   1 Doc                                   50
   _id                        title                  year    cast        genres
1  65b66f07db5c4701d374f7a3   Remember the Titans    2000    Array[0]    Drama
```

EXERCISE 4. Delete the new document.

Query:

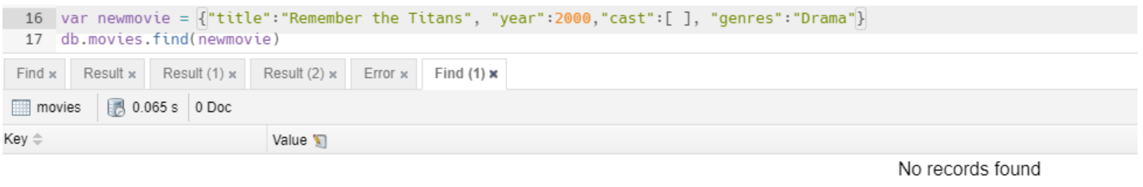
```
var newmovie = {"title":"Remember the Titans", "year":2000,"cast":[ ], "genres":"Drama"}
db.movies.deleteOne(newmovie)

var newmovie = {"title":"Remember the Titans", "year":2000,"cast":[ ], "genres":"Drama"}
db.movies.find(newmovie)
```

Results:



```
//EJERCICIO 4
var newmovie = {"title":"Remember the Titans", "year":2000,"cast":[ ], "genres":"Drama"}
db.movies.deleteOne(newmovie)
```

```
Find ×   Result ×   Result (1) ×   Result (2) ×
0.070 s
Key              Value                                        Type
(1)              { acknowledged : true, deletedCount : 1 }    Object
  acknowledged   true                                         Bool
  deletedCount   1                                            Int32
```



```
var newmovie = {"title":"Remember the Titans", "year":2000,"cast":[ ], "genres":"Drama"}
db.movies.find(newmovie)
```

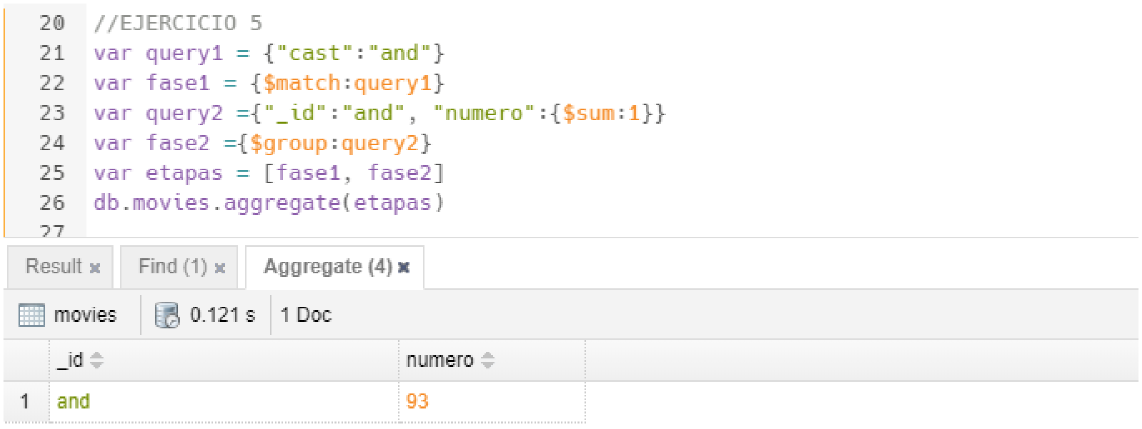
```
Find ×   Result ×   Result (1) ×   Result (2) ×   Error ×   Find (1) ×
movies   0.065 s   0 Doc
Key              Value
                                                No records found
```

EXERCISE 5. Count how many movies have actors named "and".

Query:

```
var query1 = {"cast":"and"}
var fase1 = {$match:query1}
var query2 ={"_id":"and", "numero":{$sum:1}}
var fase2 ={$group:query2}
var etapas = [fase1, fase2]
db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 5
var query1 = {"cast":"and"}
var fase1 = {$match:query1}
var query2 ={"_id":"and", "numero":{$sum:1}}
var fase2 ={$group:query2}
var etapas = [fase1, fase2]
db.movies.aggregate(etapas)
```

Result × Find (1) × Aggregate (4) ×

movies 0.121 s 1 Doc

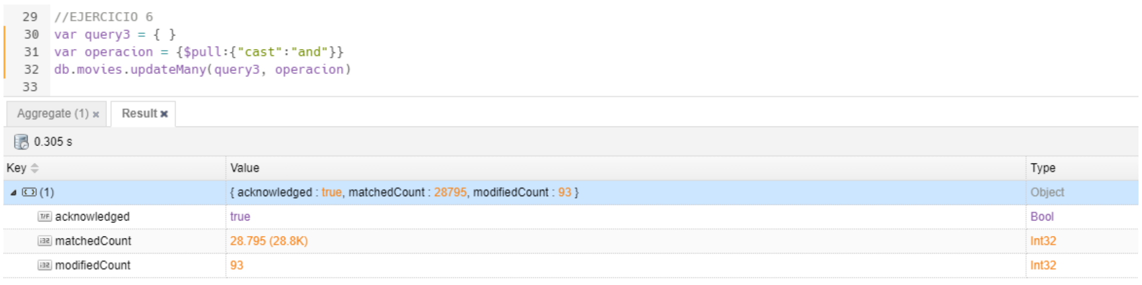
	_id	numero
1	and	93

EXERCISE 6. Extract the value "and" from the array without deleting the document.

Query:

```
var query3 = { }
var operacion = {$pull:{"cast":"and"}}
db.movies.updateMany(query3, operacion)
```

Results:



```
//EJERCICIO 6
var query3 = { }
var operacion = {$pull:{"cast":"and"}}
db.movies.updateMany(query3, operacion)
```

Aggregate (1) × Result ×

0.305 s

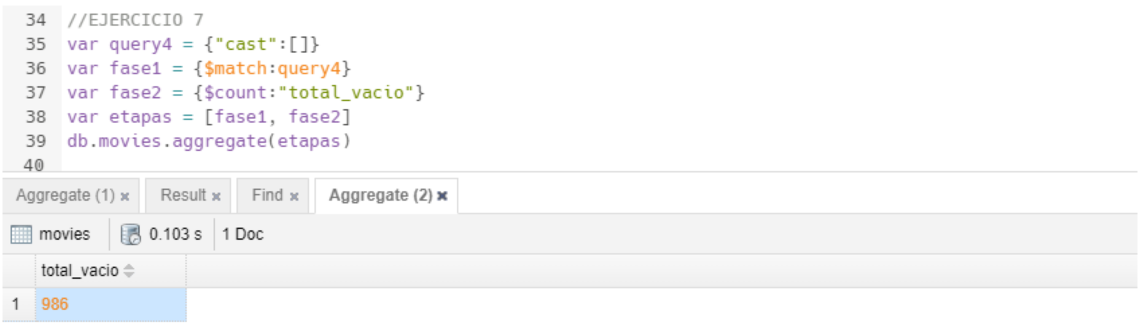
Key	Value	Type
(1)	{ acknowledged : true, matchedCount : 28795, modifiedCount : 93 }	Object
acknowledged	true	Bool
matchedCount	28.795 (28.8K)	Int32
modifiedCount	93	Int32

EXERCISE 7. Count documents where the "cast" array is empty.

Query:

```
var query4 = {"cast":[]}
var fase1 = {$match:query4}
var fase2 = {$count:"total_vacio"}
var etapas = [fase1, fase2]
db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 7
var query4 = {"cast":[]}
var fase1 = {$match:query4}
var fase2 = {$count:"total_vacio"}
var etapas = [fase1, fase2]
db.movies.aggregate(etapas)
```

Aggregate (1) × | Result × | Find × | Aggregate (2) ×

movies | 0.103 s | 1 Doc

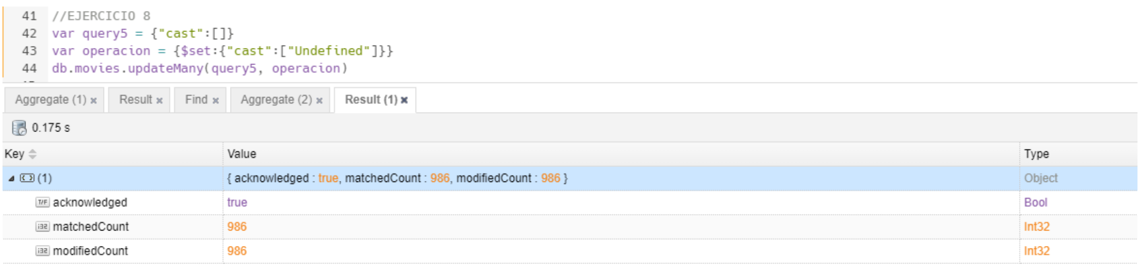
	total_vacio
1	986

EXERCISE 8. Add "Undefined" to empty "cast" arrays.

Query:

```
var query5 = {"cast":[]}

var operacion = {$set:{"cast":["Undefined"]}}

db.movies.updateMany(query5, operacion)
```

Results:

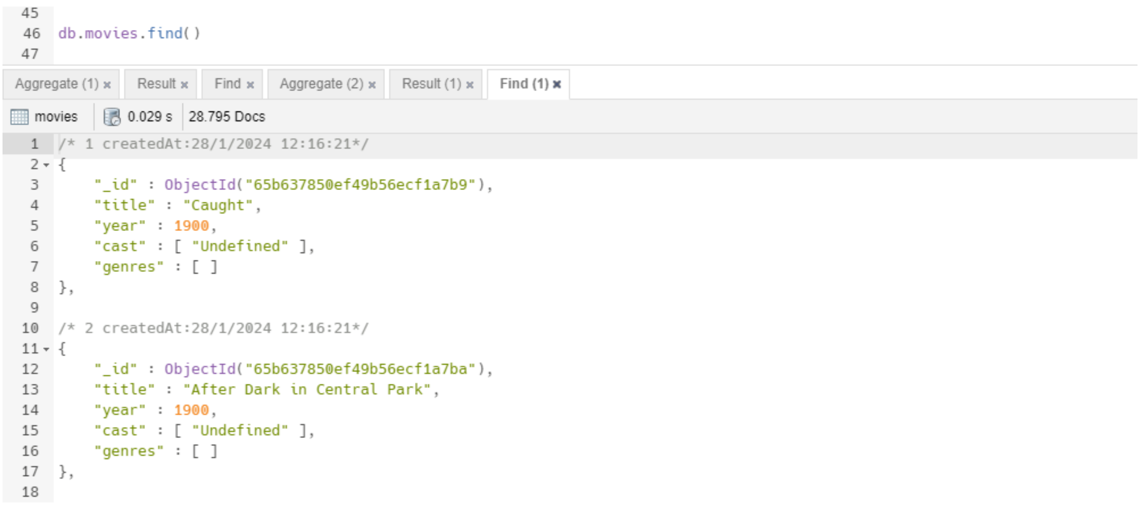


```
//EJERCICIO 8
var query5 = {"cast":[]}
var operacion = {$set:{"cast":["Undefined"]}}
db.movies.updateMany(query5, operacion)
```

Aggregate (1) × | Result × | Find × | Aggregate (2) × | Result (1) ×

0.175 s

Key	Value	Type
(1)	{ acknowledged : true, matchedCount : 986, modifiedCount : 986 }	Object
acknowledged	true	Bool
matchedCount	986	Int32
modifiedCount	986	Int32



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

Aggregate (1) × | Result × | Find × | Aggregate (2) × | Result (1) × | Find (1) ×

movies | 0.029 s | 28.795 Docs

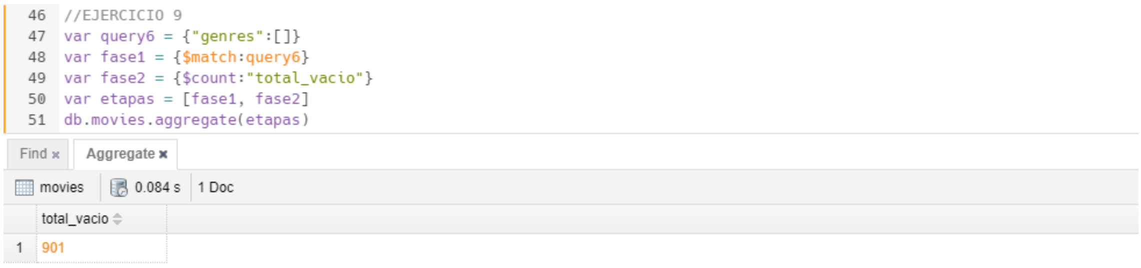
```
/* 1 createdAt:28/1/2024 12:16:21*/
{
    "_id" : ObjectId("65b637850ef49b56ecf1a7b9"),
    "title" : "Caught",
    "year" : 1900,
    "cast" : [ "Undefined" ],
    "genres" : [ ]
},

/* 2 createdAt:28/1/2024 12:16:21*/
{
    "_id" : ObjectId("65b637850ef49b56ecf1a7ba"),
    "title" : "After Dark in Central Park",
    "year" : 1900,
    "cast" : [ "Undefined" ],
    "genres" : [ ]
},
```

EXERCISE 9. Count how many documents have an empty "genres" array.

Query:

```
var query6 = {"genres":[]}

var fase1 = {$match:query6}

var fase2 = {$count:"total_vacio"}

var etapas = [fase1, fase2]

db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 9
var query6 = {"genres":[]}
var fase1 = {$match:query6}
var fase2 = {$count:"total_vacio"}
var etapas = [fase1, fase2]
db.movies.aggregate(etapas)
```

Find × | Aggregate ×

movies | 0.084 s | 1 Doc

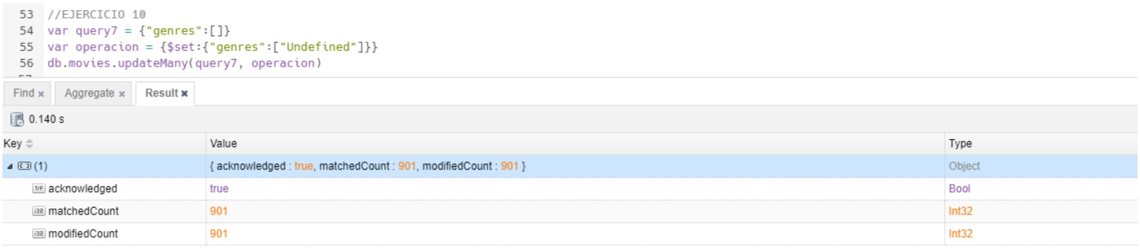
	total_vacio
1	901

EXERCISE 10. Add "Undefined" to empty "genres" arrays.

Query:

```
var query7 = {"genres":[]}

var operacion = {$set:{"genres":["Undefined"]}}

db.movies.updateMany(query7, operacion)
```

Results:

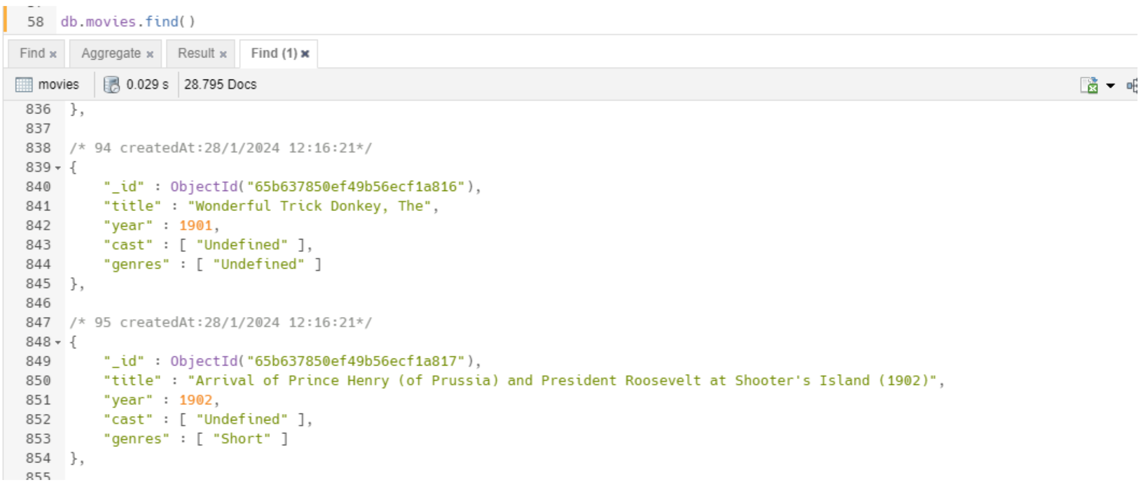


```
//EJERCICIO 10
var query7 = {"genres":[]}
var operacion = {$set:{"genres":["Undefined"]}}
db.movies.updateMany(query7, operacion)
```

Find × | Aggregate × | Result ×

0.140 s

Key	Value	Type
(1)	{ acknowledged : true, matchedCount : 901, modifiedCount : 901 }	Object
acknowledged	true	Bool
matchedCount	901	Int32
modifiedCount	901	Int32



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

Find × | Aggregate × | Result × | Find (1) ×

movies | 0.029 s | 28.795 Docs

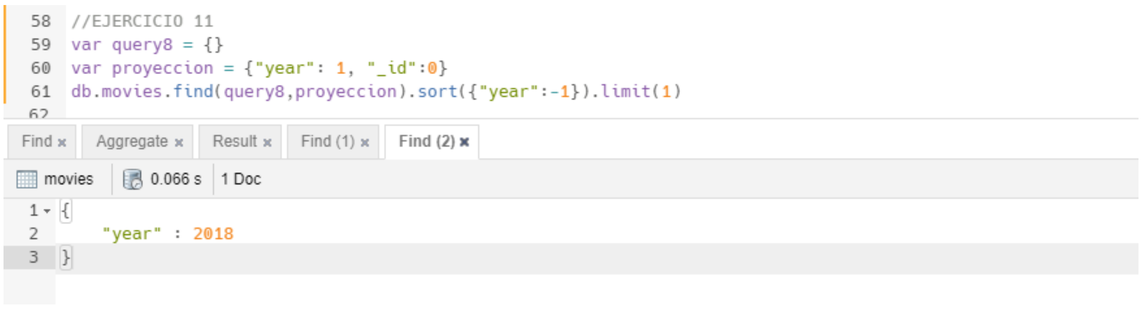
```
},

/* 94 createdAt:28/1/2024 12:16:21*/
{
    "_id" : ObjectId("65b637850ef49b56ecf1a816"),
    "title" : "Wonderful Trick Donkey, The",
    "year" : 1901,
    "cast" : [ "Undefined" ],
    "genres" : [ "Undefined" ]
},

/* 95 createdAt:28/1/2024 12:16:21*/
{
    "_id" : ObjectId("65b637850ef49b56ecf1a817"),
    "title" : "Arrival of Prince Henry (of Prussia) and President Roosevelt at Shooter's Island (1902)",
    "year" : 1902,
    "cast" : [ "Undefined" ],
    "genres" : [ "Short" ]
},
```

EXERCISE 11. Most recent year in the collection.

Query:

```
var query8 = {}

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

db.movies.find(query8,proyeccion).sort({"year":-1}).limit(1)
```

Results:



```
//EJERCICIO 11
var query8 = {}
var proyeccion = {"year": 1, "_id":0}
db.movies.find(query8,proyeccion).sort({"year":-1}).limit(1)
```

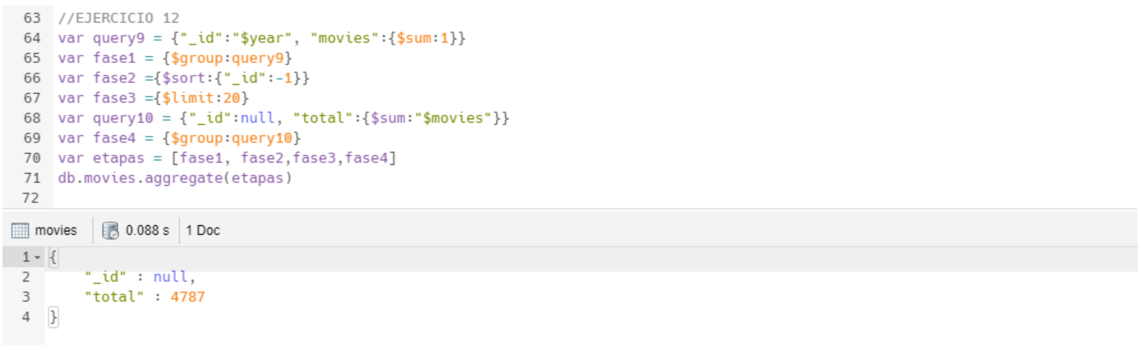
```
{
    "year" : 2018
}
```

EXERCISE 12. Number of movies from the last 20 years.

Query:

```
var query9 = {"_id":"$year", "movies":{$sum:1}}

var fase1 = {$group:query9}

var fase2 ={$sort:{"_id":-1}}

var fase3 ={$limit:20}

var query10 = {"_id":null, "total":{$sum:"$movies"}}

var fase4 = {$group:query10}

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

db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 12
var query9 = {"_id":"$year", "movies":{$sum:1}}
var fase1 = {$group:query9}
var fase2 ={$sort:{"_id":-1}}
var fase3 ={$limit:20}
var query10 = {"_id":null, "total":{$sum:"$movies"}}
var fase4 = {$group:query10}
var etapas = [fase1, fase2,fase3,fase4]
db.movies.aggregate(etapas)
```

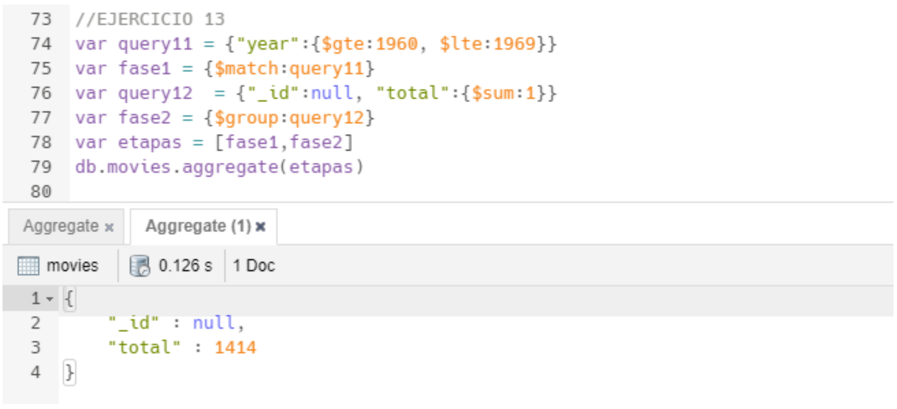
```
{
    "_id" : null,
    "total" : 4787
}
```

EXERCISE 13. Number of movies from the 1960s.

Query:

```
var query11 = {"year":{$gte:1960, $lte:1969}}
var fase1 = {$match:query11}
var query12  = {"_id":null, "total":{$sum:1}}
var fase2 = {$group:query12}
var etapas = [fase1,fase2]
db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 13
var query11 = {"year":{$gte:1960, $lte:1969}}
var fase1 = {$match:query11}
var query12  = {"_id":null, "total":{$sum:1}}
var fase2 = {$group:query12}
var etapas = [fase1,fase2]
db.movies.aggregate(etapas)
```

```
{
    "_id" : null,
    "total" : 1414
}
```

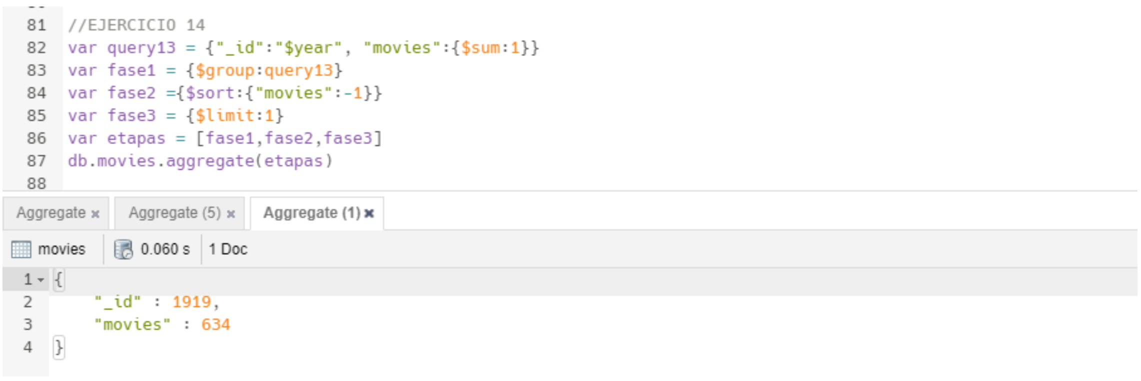
EXERCISE 14. Year or years with the most movies.

First, we execute phase 1 and phase 2. After reviewing the results, we set the "limit". In this case, there are no repetitions.

Query:

```
var query13 = {"_id":"$year", "movies":{$sum:1}}
var fase1 = {$group:query13}
var fase2 ={$sort:{"movies":-1}}
var fase3 = {$limit:1}
var etapas = [fase1,fase2,fase3]
db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 14
var query13 = {"_id":"$year", "movies":{$sum:1}}
var fase1 = {$group:query13}
var fase2 ={$sort:{"movies":-1}}
var fase3 = {$limit:1}
var etapas = [fase1,fase2,fase3]
db.movies.aggregate(etapas)
```

```
{
    "_id" : 1919,
    "movies" : 634
}
```

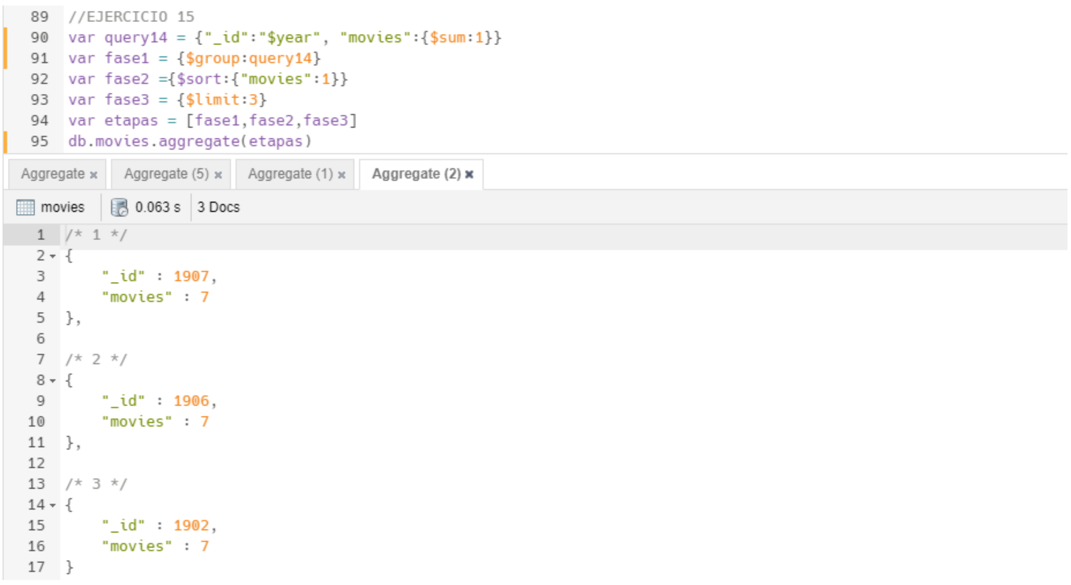
EXERCISE 15. Year or years with the fewest movies.

First, we execute phase 1 and phase 2. After reviewing the results, we set the "limit". In this case, three years are repeated, so we set limit 3.

Query:

```
var query14 = {"_id":"$year", "movies":{$sum:1}}
var fase1 = {$group:query14}
var fase2 ={$sort:{"movies":1}}
var fase3 = {$limit:3}
var etapas = [fase1,fase2,fase3]
db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 15
var query14 = {"_id":"$year", "movies":{$sum:1}}
var fase1 = {$group:query14}
var fase2 ={$sort:{"movies":1}}
var fase3 = {$limit:3}
var etapas = [fase1,fase2,fase3]
db.movies.aggregate(etapas)
```

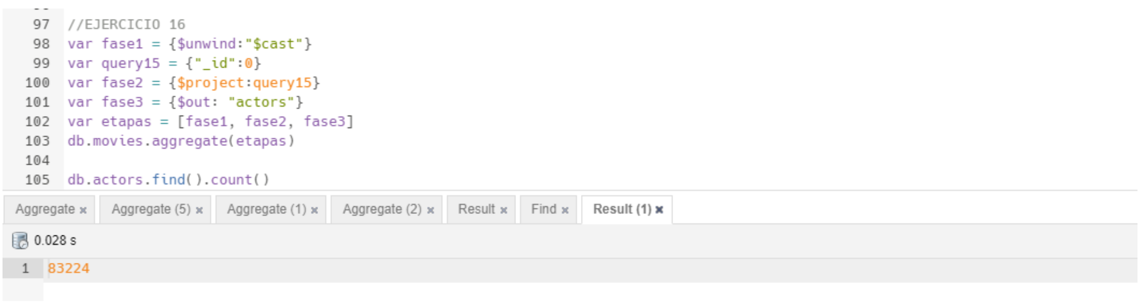
```
/* 1 */
{
    "_id" : 1907,
    "movies" : 7
},

/* 2 */
{
    "_id" : 1906,
    "movies" : 7
},

/* 3 */
{
    "_id" : 1902,
    "movies" : 7
}
```

EXERCISE 16. New collection "actors".

Query:

```
var fase1 = {$unwind:"$cast"}
var query15 = {"_id":0}
var fase2 = {$project:query15}
var fase3 = {$out: "actors"}
var etapas = [fase1, fase2, fase3]
db.movies.aggregate(etapas)

db.actors.find().count()
```

Results:



```
//EJERCICIO 16
var fase1 = {$unwind:"$cast"}
var query15 = {"_id":0}
var fase2 = {$project:query15}
var fase3 = {$out: "actors"}
var etapas = [fase1, fase2, fase3]
db.movies.aggregate(etapas)

db.actors.find().count()
```

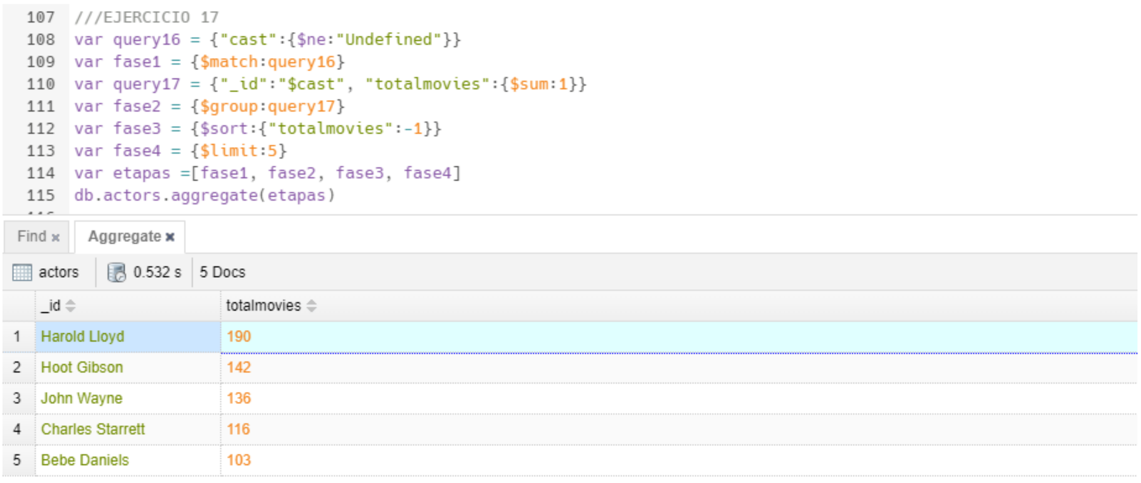
```
1 83224
```

EXERCISE 17: Based on the actors collection (new collection), display a list of the top 5 actors who have participated in the most movies, showing the number of movies they have appeared in.

Query:

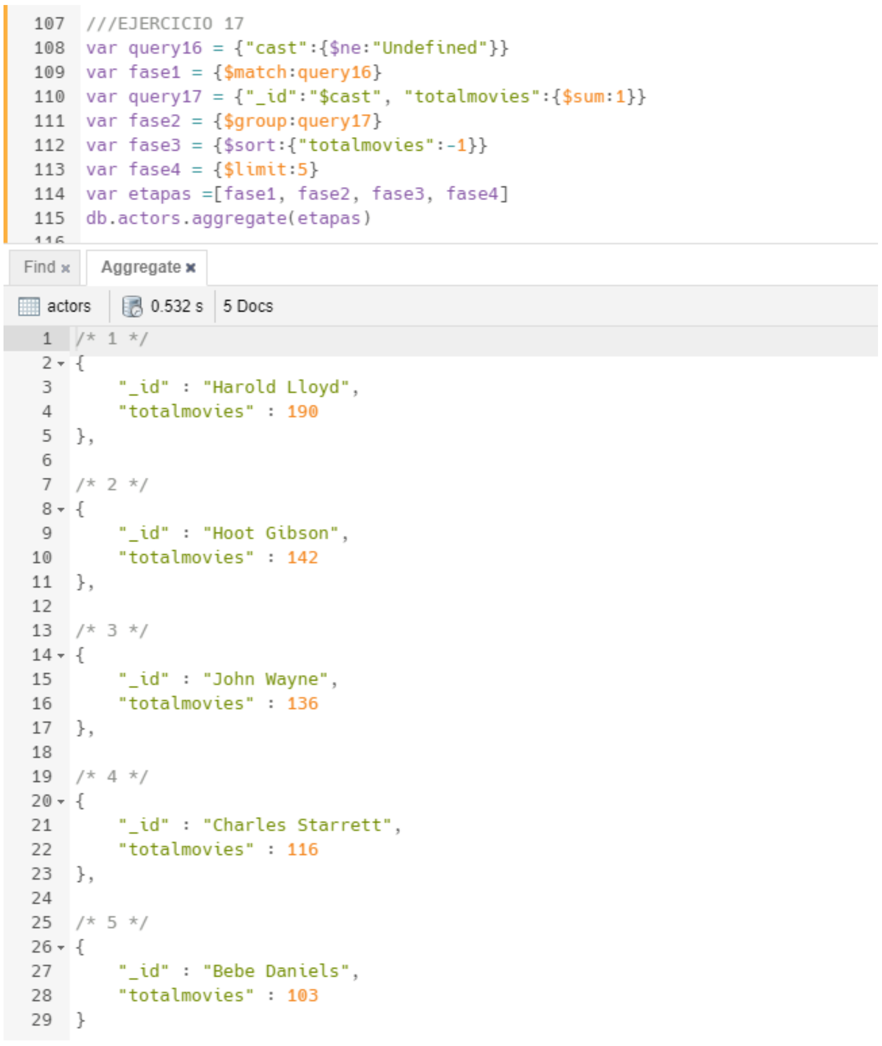
```
var query16 = {"cast":{$ne:"Undefined"}}
var fase1 = {$match:query16}
var query17 = {"_id":"$cast", "totalmovies":{$sum:1}}
var fase2 = {$group:query17}
var fase3 = {$sort:{"totalmovies":-1}}
var fase4 = {$limit:5}
var etapas =[fase1, fase2, fase3, fase4]
db.actors.aggregate(etapas)
```

Results:



```
///EJERCICIO 17
var query16 = {"cast":{$ne:"Undefined"}}
var fase1 = {$match:query16}
var query17 = {"_id":"$cast", "totalmovies":{$sum:1}}
var fase2 = {$group:query17}
var fase3 = {$sort:{"totalmovies":-1}}
var fase4 = {$limit:5}
var etapas =[fase1, fase2, fase3, fase4]
db.actors.aggregate(etapas)
```

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



```
///EJERCICIO 17
var query16 = {"cast":{$ne:"Undefined"}}
var fase1 = {$match:query16}
var query17 = {"_id":"$cast", "totalmovies":{$sum:1}}
var fase2 = {$group:query17}
var fase3 = {$sort:{"totalmovies":-1}}
var fase4 = {$limit:5}
var etapas =[fase1, fase2, fase3, fase4]
db.actors.aggregate(etapas)
```

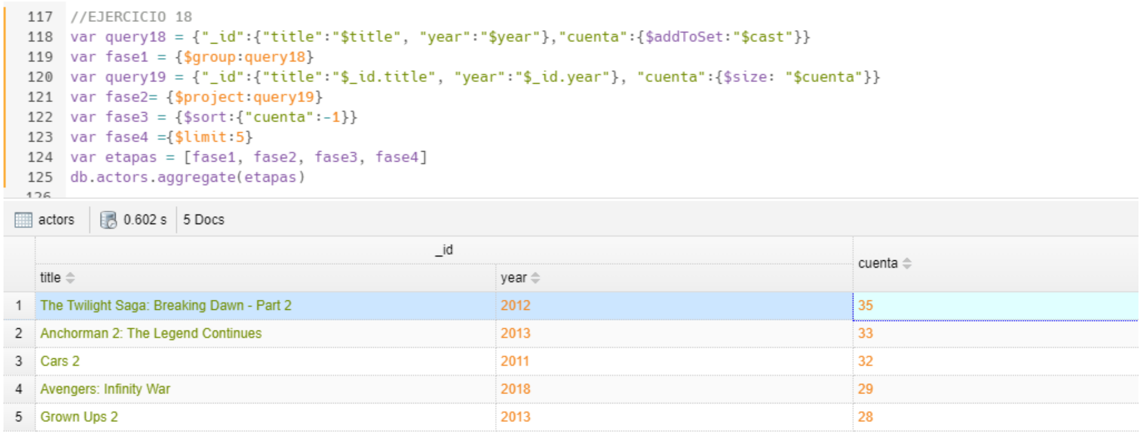
```
/* 1 */
{
    "_id" : "Harold Lloyd",
    "totalmovies" : 190
},

/* 2 */
{
    "_id" : "Hoot Gibson",
    "totalmovies" : 142
},

/* 3 */
{
    "_id" : "John Wayne",
    "totalmovies" : 136
},

/* 4 */
{
    "_id" : "Charles Starrett",
    "totalmovies" : 116
},

/* 5 */
{
    "_id" : "Bebe Daniels",
    "totalmovies" : 103
}
```

EXERCISE 18: Based on the actors collection (new collection), group by movie and year, displaying the top 5 movies with the highest number of actors, showing the total number of actors in each.

Query:

```
var query18 = {"_id":{"title":"$title", "year":"$year"},"cuenta":{$addToSet:"$cast"}}

var fase1 = {$group:query18}

var query19 = {"_id":{"title":"$_id.title", "year":"$_id.year"}, "cuenta":{$size:
"$cuenta"}}

var fase2= {$project:query19}

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

var fase4 ={$limit:5}

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

db.actors.aggregate(etapas)
```

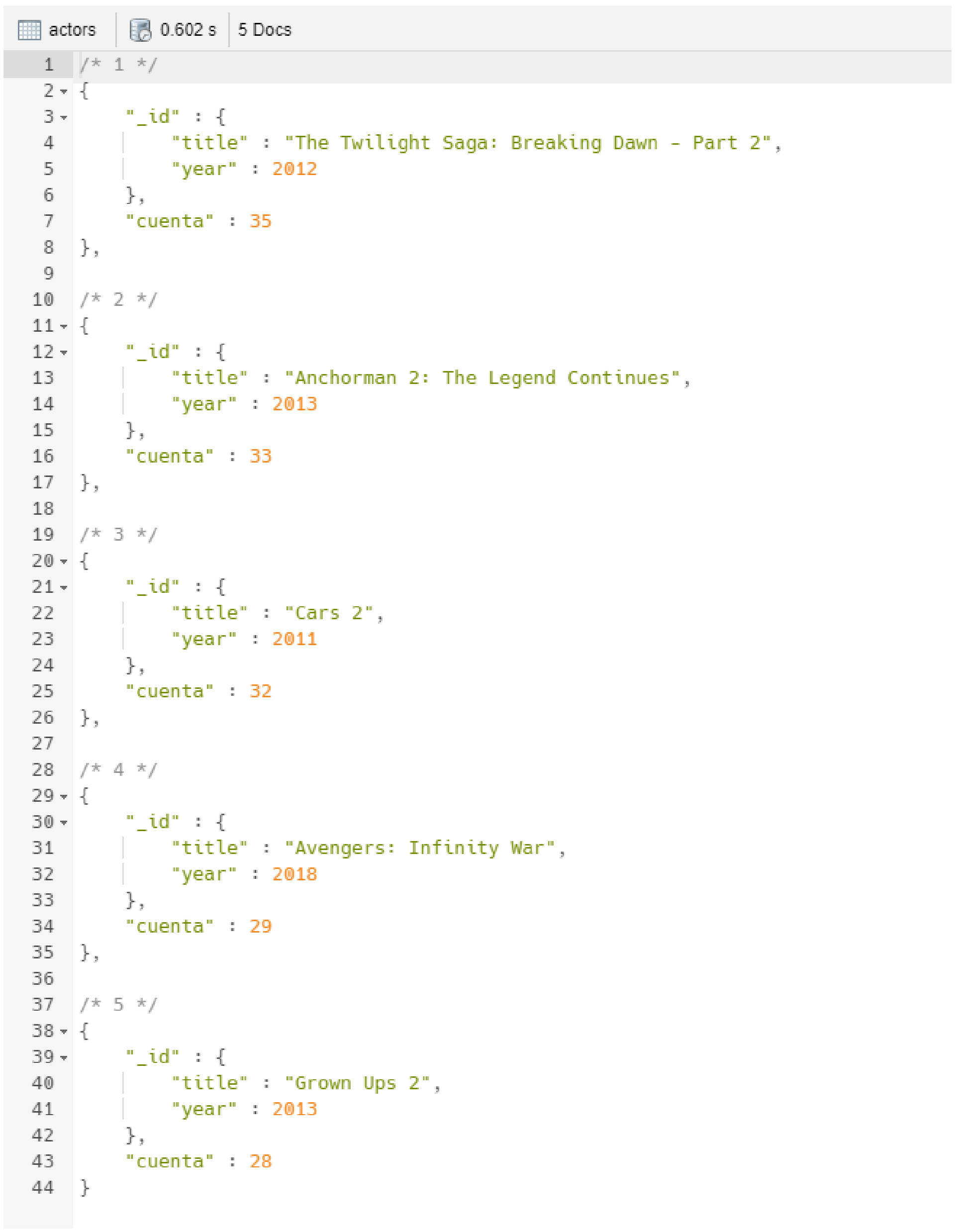
Results:



```
//EJERCICIO 18
var query18 = {"_id":{"title":"$title", "year":"$year"},"cuenta":{$addToSet:"$cast"}}
var fase1 = {$group:query18}
var query19 = {"_id":{"title":"$_id.title", "year":"$_id.year"}, "cuenta":{$size: "$cuenta"}}
var fase2= {$project:query19}
var fase3 = {$sort:{"cuenta":-1}}
var fase4 ={$limit:5}
var etapas = [fase1, fase2, fase3, fase4]
db.actors.aggregate(etapas)
```

actors 0.602 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



actors 0.602 s 5 Docs

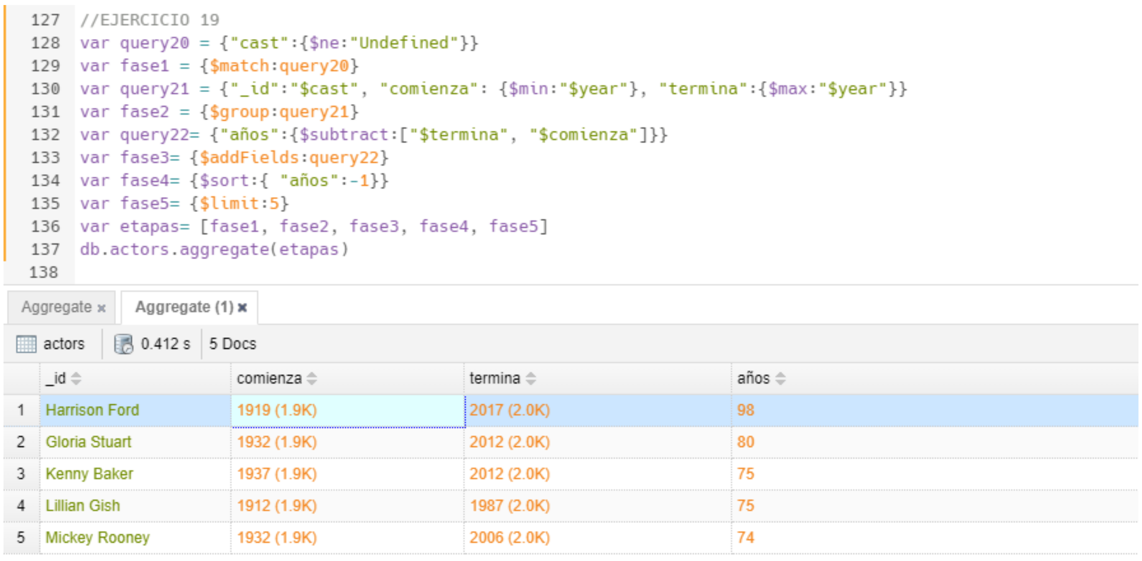
```
/* 1 */
{
    "_id" : {
        "title" : "The Twilight Saga: Breaking Dawn - Part 2",
        "year" : 2012
    },
    "cuenta" : 35
},

/* 2 */
{
    "_id" : {
        "title" : "Anchorman 2: The Legend Continues",
        "year" : 2013
    },
    "cuenta" : 33
},

/* 3 */
{
    "_id" : {
        "title" : "Cars 2",
        "year" : 2011
    },
    "cuenta" : 32
},

/* 4 */
{
    "_id" : {
        "title" : "Avengers: Infinity War",
        "year" : 2018
    },
    "cuenta" : 29
},

/* 5 */
{
    "_id" : {
        "title" : "Grown Ups 2",
        "year" : 2013
    },
    "cuenta" : 28
}
```

EXERCISE 19: Based on the actors collection (new collection), display the top 5 actors with the longest careers. The output should include the year their career started, the year it ended, and the total number of years they worked.

Query:

```
var query20 = {"cast":{$ne:"Undefined"}}
var fase1 = {$match:query20}
var query21 = {"_id":"$cast", "comienza": {$min:"$year"}, "termina":{$max:"$year"}}
var fase2 = {$group:query21}
var query22= {"años":{$subtract:["$termina", "$comienza"]}}
var fase3= {$addFields:query22}
var fase4= {$sort:{ "años":-1}}
var fase5= {$limit:5}
var etapas= [fase1, fase2, fase3, fase4, fase5]
db.actors.aggregate(etapas)
```

Results:

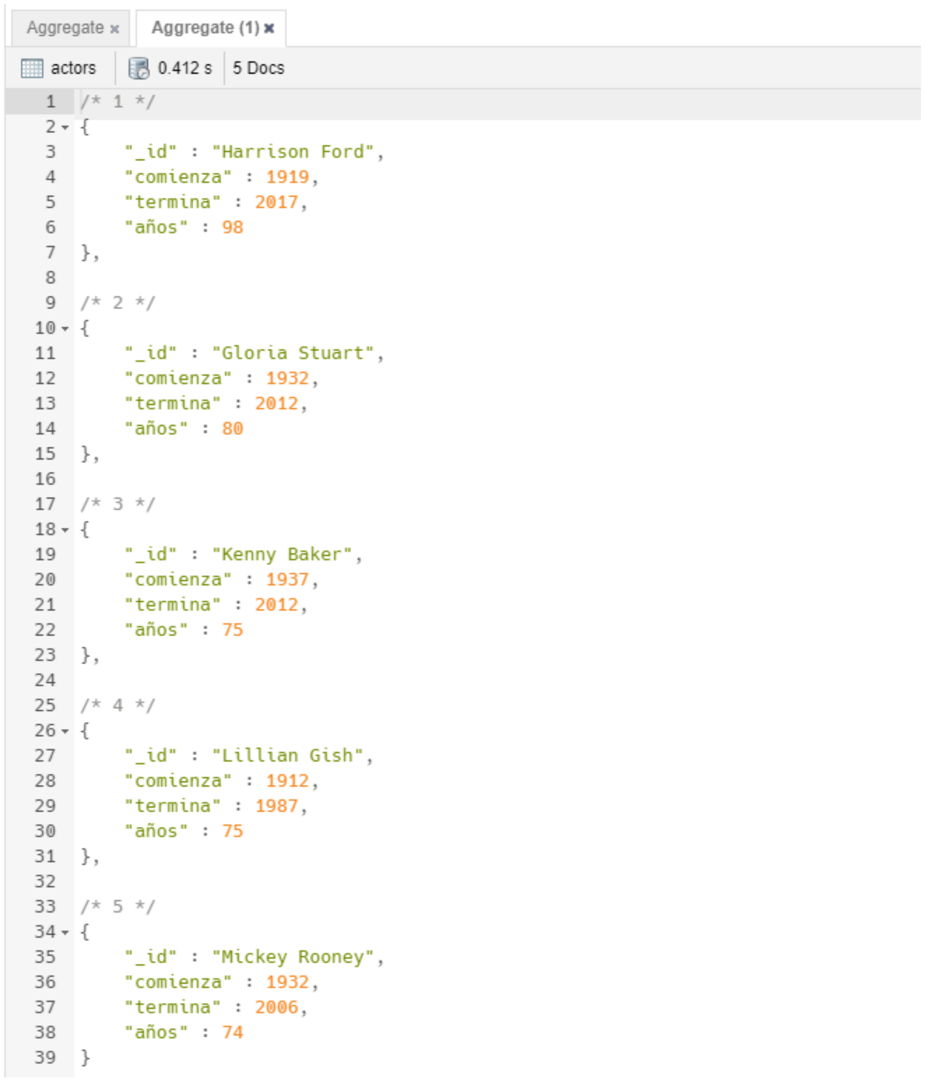


```
//EJERCICIO 19
var query20 = {"cast":{$ne:"Undefined"}}
var fase1 = {$match:query20}
var query21 = {"_id":"$cast", "comienza": {$min:"$year"}, "termina":{$max:"$year"}}
var fase2 = {$group:query21}
var query22= {"años":{$subtract:["$termina", "$comienza"]}}
var fase3= {$addFields:query22}
var fase4= {$sort:{ "años":-1}}
var fase5= {$limit:5}
var etapas= [fase1, fase2, fase3, fase4, fase5]
db.actors.aggregate(etapas)
```

Aggregate | Aggregate (1)

actors | 0.412 s | 5 Docs

	_id	comienza	termina	años
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	Mickey Rooney	1932 (1.9K)	2006 (2.0K)	74



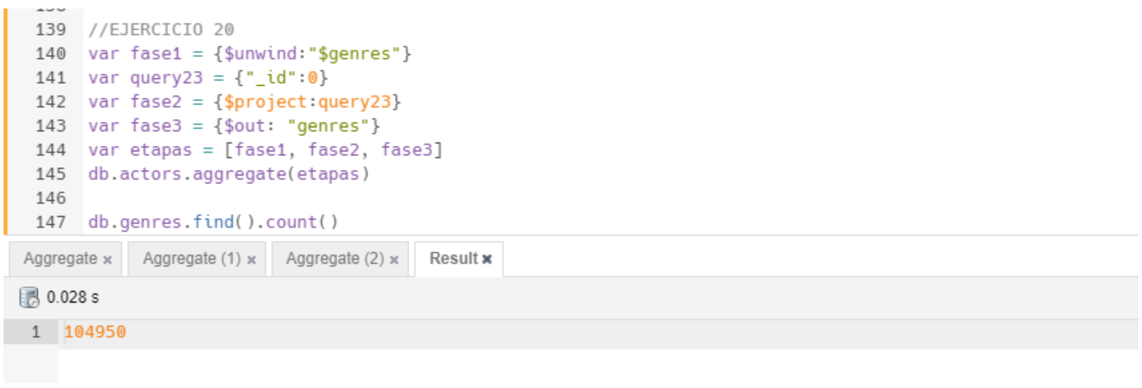
```
/* 1 */
{
    "_id" : "Harrison Ford",
    "comienza" : 1919,
    "termina" : 2017,
    "años" : 98
},

/* 2 */
{
    "_id" : "Gloria Stuart",
    "comienza" : 1932,
    "termina" : 2012,
    "años" : 80
},

/* 3 */
{
    "_id" : "Kenny Baker",
    "comienza" : 1937,
    "termina" : 2012,
    "años" : 75
},

/* 4 */
{
    "_id" : "Lillian Gish",
    "comienza" : 1912,
    "termina" : 1987,
    "años" : 75
},

/* 5 */
{
    "_id" : "Mickey Rooney",
    "comienza" : 1932,
    "termina" : 2006,
    "años" : 74
}
```

EXERCISE 20: Based on the actors collection (new collection), save the data into a new collection called "genres" by performing the ‘$unwind’ stage on the "genres" field. Then, count the number of documents in the new collection.

Query:

```
var fase1 = {$unwind:"$genres"}

var query23 = {"_id":0}

var fase2 = {$project:query23}

var fase3 = {$out: "genres"}

var etapas = [fase1, fase2, fase3]

db.actors.aggregate(etapas)


db.genres.find().count()
```

Results:



```
//EJERCICIO 20
var fase1 = {$unwind:"$genres"}
var query23 = {"_id":0}
var fase2 = {$project:query23}
var fase3 = {$out: "genres"}
var etapas = [fase1, fase2, fase3]
db.actors.aggregate(etapas)

db.genres.find().count()
```

Aggregate × | Aggregate (1) × | Aggregate (2) × | Result ×

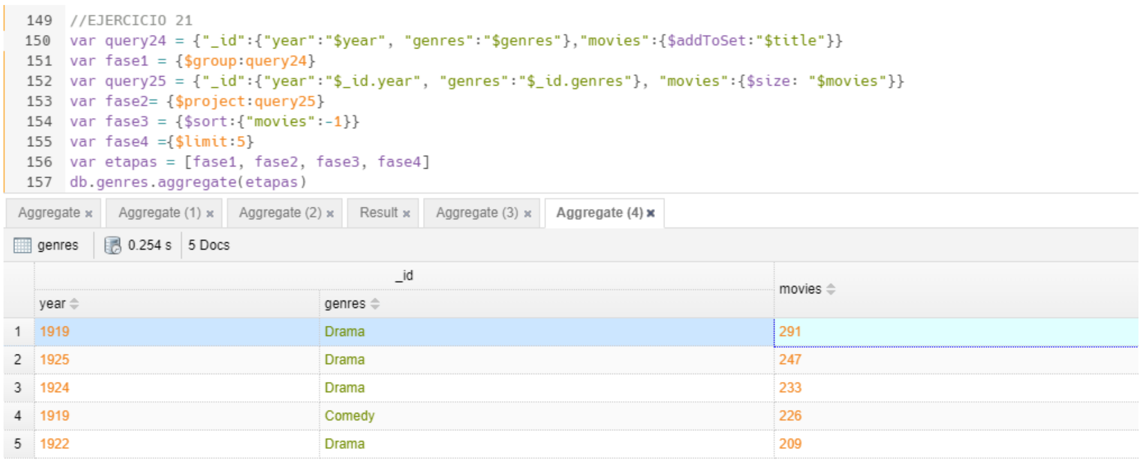
```
0.028 s
1  104950
```

EXERCISE 21: Based on the genres collection (new collection), display the top 5 records grouped by "Year and Genre" that have the highest number of different movies, showing the total number of movies in each group.

Query:

```
var query24 = {"_id":{"year":"$year", "genres":"$genres"},"movies":{$addToSet:"$title"}}

var fase1 = {$group:query24}

var query25 = {"_id":{"year":"$_id.year", "genres":"$_id.genres"}, "movies":{$size:
"$movies"}}

var fase2= {$project:query25}

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

var fase4 ={$limit:5}

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

db.genres.aggregate(etapas)
```

Results:

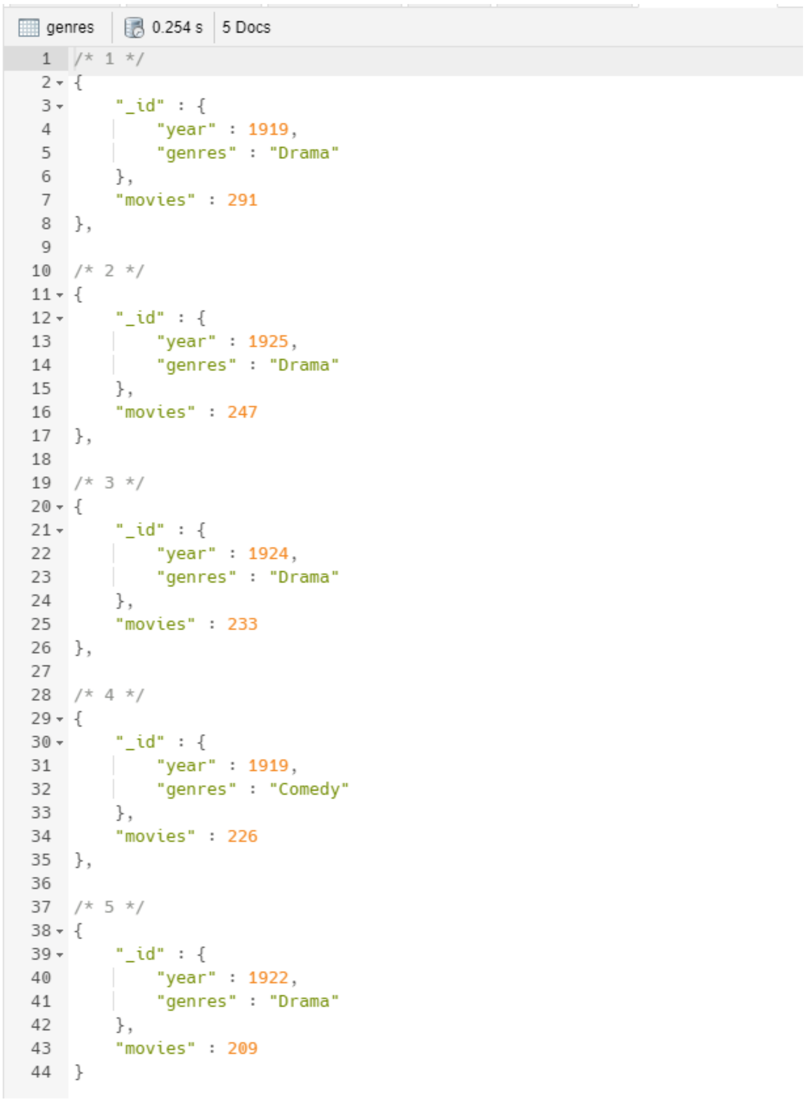


```
//EJERCICIO 21
var query24 = {"_id":{"year":"$year", "genres":"$genres"},"movies":{$addToSet:"$title"}}
var fase1 = {$group:query24}
var query25 = {"_id":{"year":"$_id.year", "genres":"$_id.genres"}, "movies":{$size: "$movies"}}
var fase2= {$project:query25}
var fase3 = {$sort:{"movies":-1}}
var fase4 ={$limit:5}
var etapas = [fase1, fase2, fase3, fase4]
db.genres.aggregate(etapas)
```

Aggregate × | Aggregate (1) × | Aggregate (2) × | Result × | Aggregate (3) × | Aggregate (4) ×

genres | 0.254 s | 5 Docs

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


genres | 0.254 s | 5 Docs

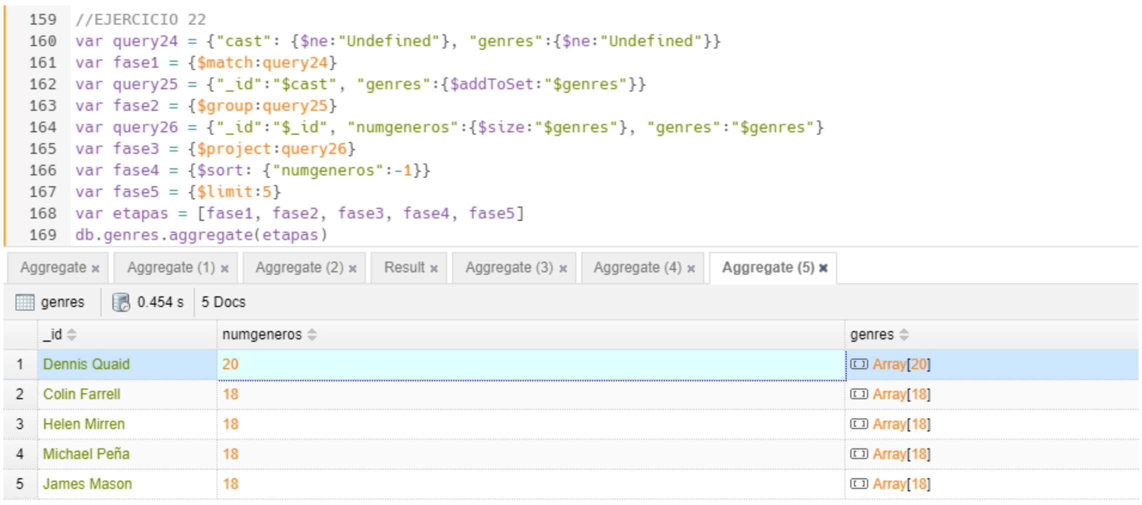
```
/* 1 */
{
    "_id" : {
        "year" : 1919,
        "genres" : "Drama"
    },
    "movies" : 291
},

/* 2 */
{
    "_id" : {
        "year" : 1925,
        "genres" : "Drama"
    },
    "movies" : 247
},

/* 3 */
{
    "_id" : {
        "year" : 1924,
        "genres" : "Drama"
    },
    "movies" : 233
},

/* 4 */
{
    "_id" : {
        "year" : 1919,
        "genres" : "Comedy"
    },
    "movies" : 226
},

/* 5 */
{
    "_id" : {
        "year" : 1922,
        "genres" : "Drama"
    },
    "movies" : 209
}
```

EXERCISE 22: Using the genres collection (new collection), display the top 5 actors who have participated in the most different genres, along with the genres they have acted in. The total number of different genres each actor has participated in should be shown. No data is deleted from the collection, only filtered to exclude certain records.

Query:

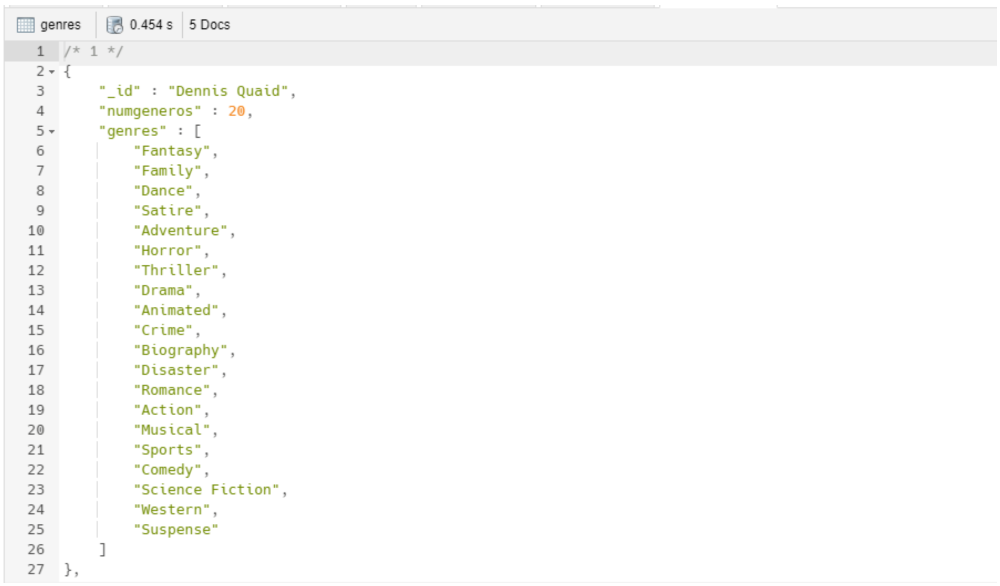
```
var query24 = {"cast": {$ne:"Undefined"}, "genres":{$ne:"Undefined"}}

var fase1 = {$match:query24}

var query25 = {"_id":"$cast", "genres":{$addToSet:"$genres"}}

var fase2 = {$group:query25}

var query26 = {"_id":"$_id", "numgeneros":{$size:"$genres"}, "genres":"$genres"}

var fase3 = {$project:query26}

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

var fase5 = {$limit:5}

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

db.genres.aggregate(etapas)
```

Results:

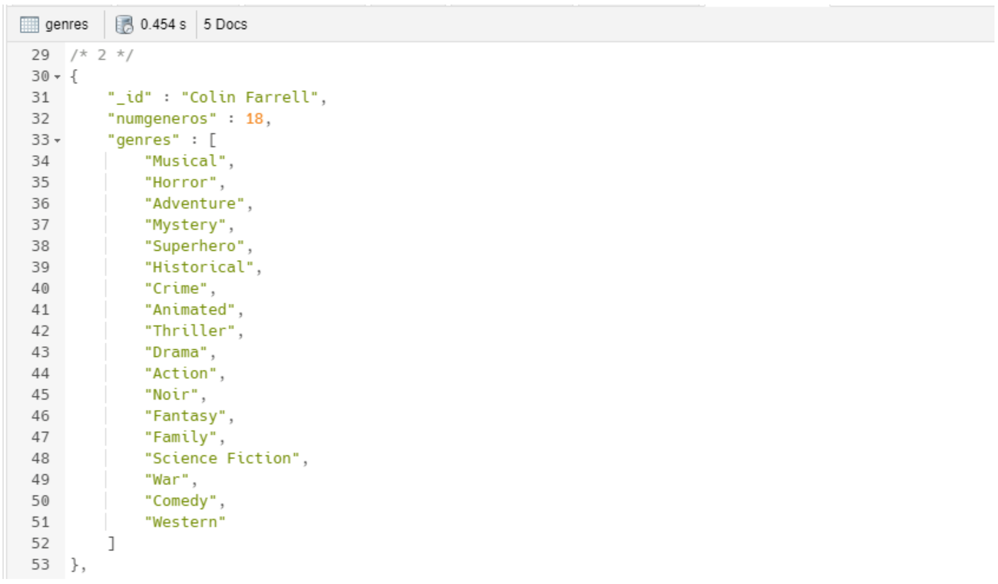


```
//EJERCICIO 22
var query24 = {"cast": {$ne:"Undefined"}, "genres":{$ne:"Undefined"}}
var fase1 = {$match:query24}
var query25 = {"_id":"$cast", "genres":{$addToSet:"$genres"}}
var fase2 = {$group:query25}
var query26 = {"_id":"$_id", "numgeneros":{$size:"$genres"}, "genres":"$genres"}
var fase3 = {$project:query26}
var fase4 = {$sort: {"numgeneros":-1}}
var fase5 = {$limit:5}
var etapas = [fase1, fase2, fase3, fase4, fase5]
db.genres.aggregate(etapas)
```

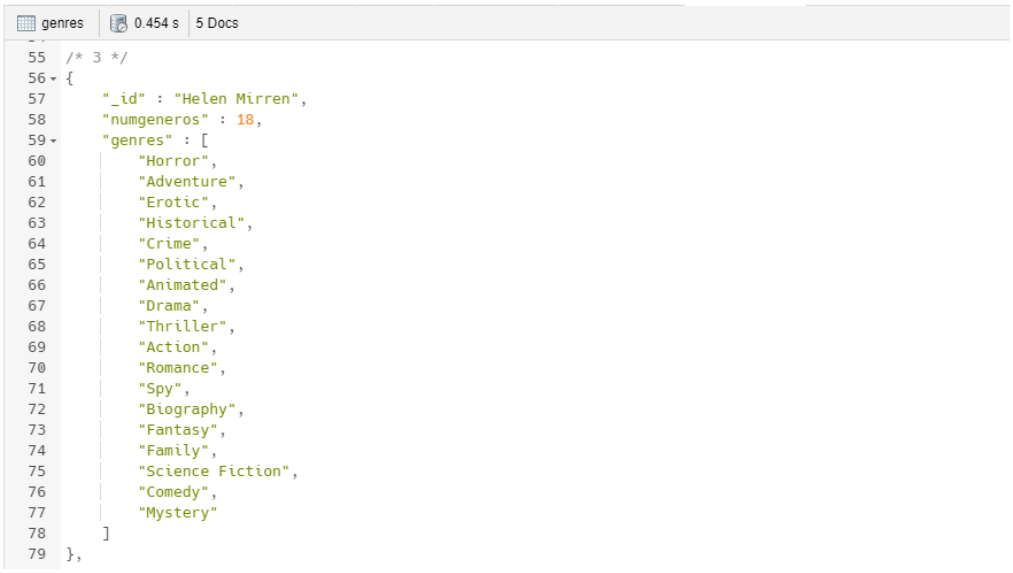
	_id	numgeneros	genres
1	Dennis Quaid	20	Array[20]
2	Colin Farrell	18	Array[18]
3	Helen Mirren	18	Array[18]
4	Michael Peña	18	Array[18]
5	James Mason	18	Array[18]



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

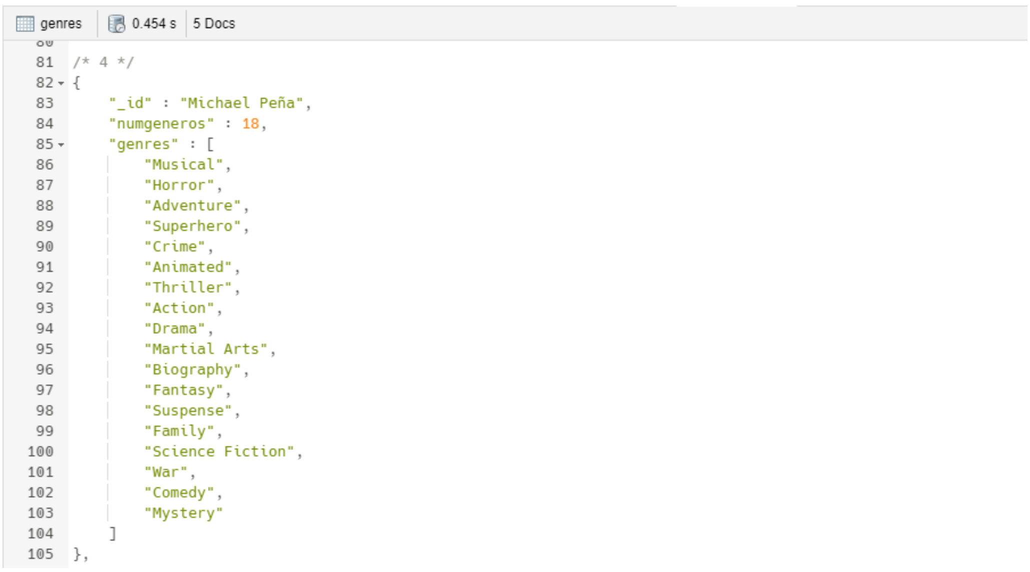


```
/* 2 */
{
    "_id" : "Colin Farrell",
    "numgeneros" : 18,
    "genres" : [
        "Musical",
        "Horror",
        "Adventure",
        "Mystery",
        "Superhero",
        "Historical",
        "Crime",
        "Animated",
        "Thriller",
        "Drama",
        "Action",
        "Noir",
        "Fantasy",
        "Family",
        "Science Fiction",
        "War",
        "Comedy",
        "Western"
    ]
},
```



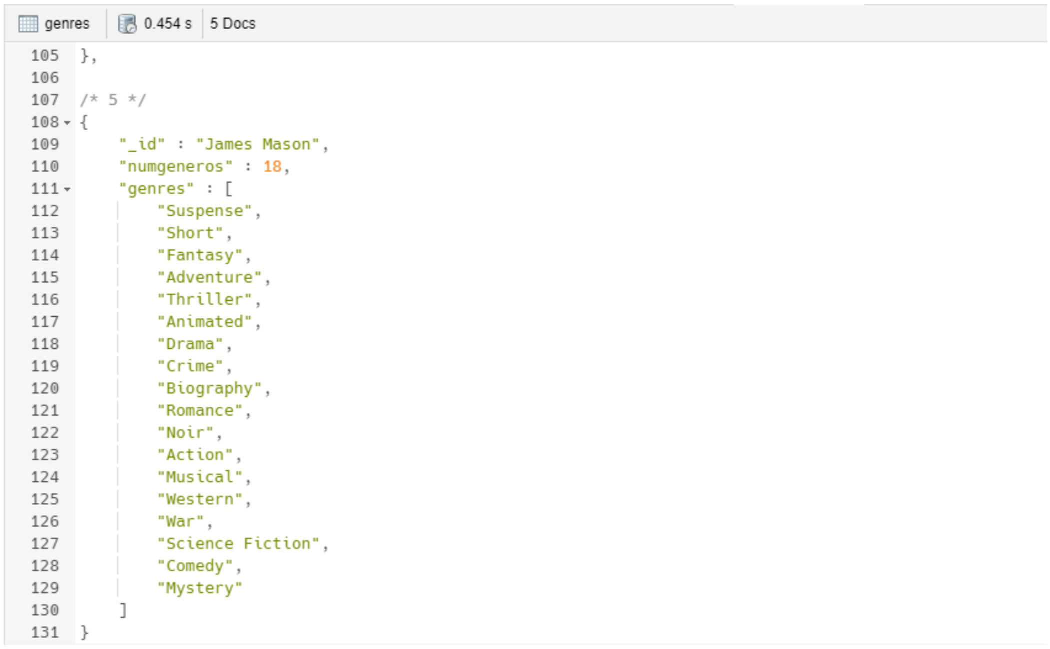
genres | 0.454 s | 5 Docs

```
/* 3 */
{
    "_id" : "Helen Mirren",
    "numgeneros" : 18,
    "genres" : [
        "Horror",
        "Adventure",
        "Erotic",
        "Historical",
        "Crime",
        "Political",
        "Animated",
        "Drama",
        "Thriller",
        "Action",
        "Romance",
        "Spy",
        "Biography",
        "Fantasy",
        "Family",
        "Science Fiction",
        "Comedy",
        "Mystery"
    ]
},
```



genres | 0.454 s | 5 Docs

```
/* 4 */
{
    "_id" : "Michael Peña",
    "numgeneros" : 18,
    "genres" : [
        "Musical",
        "Horror",
        "Adventure",
        "Superhero",
        "Crime",
        "Animated",
        "Thriller",
        "Action",
        "Drama",
        "Martial Arts",
        "Biography",
        "Fantasy",
        "Suspense",
        "Family",
        "Science Fiction",
        "War",
        "Comedy",
        "Mystery"
    ]
},
```



genres | 0.454 s | 5 Docs

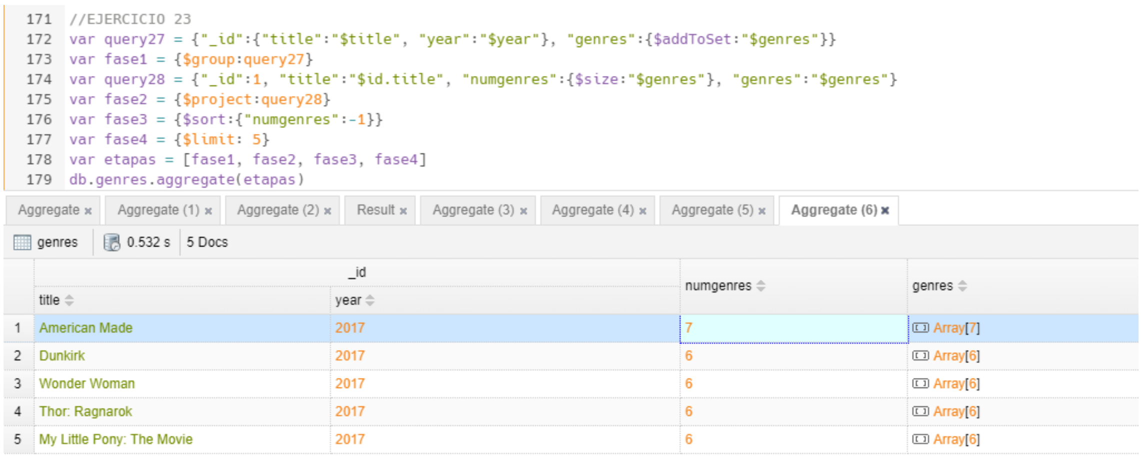
```
},

/* 5 */
{
    "_id" : "James Mason",
    "numgeneros" : 18,
    "genres" : [
        "Suspense",
        "Short",
        "Fantasy",
        "Adventure",
        "Thriller",
        "Animated",
        "Drama",
        "Crime",
        "Biography",
        "Romance",
        "Noir",
        "Action",
        "Musical",
        "Western",
        "War",
        "Science Fiction",
        "Comedy",
        "Mystery"
    ]
}
```

EXERCISE 23: Using the genres collection (new collection), display the top 5 movies and their corresponding year that have been categorized under the most different genres. The genres and the total number of genres for each movie should also be shown.

Query:

```
var query27 = {"_id":{"title":"$title", "year":"$year"}, "genres":{$addToSet:"$genres"}}

var fase1 = {$group:query27}

var query28 = {"_id":1, "title":"$id.title", "numgenres":{$size:"$genres"},
"genres":"$genres"}

var fase2 = {$project:query28}

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

var fase4 = {$limit: 5}

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

db.genres.aggregate(etapas)
```

Results:

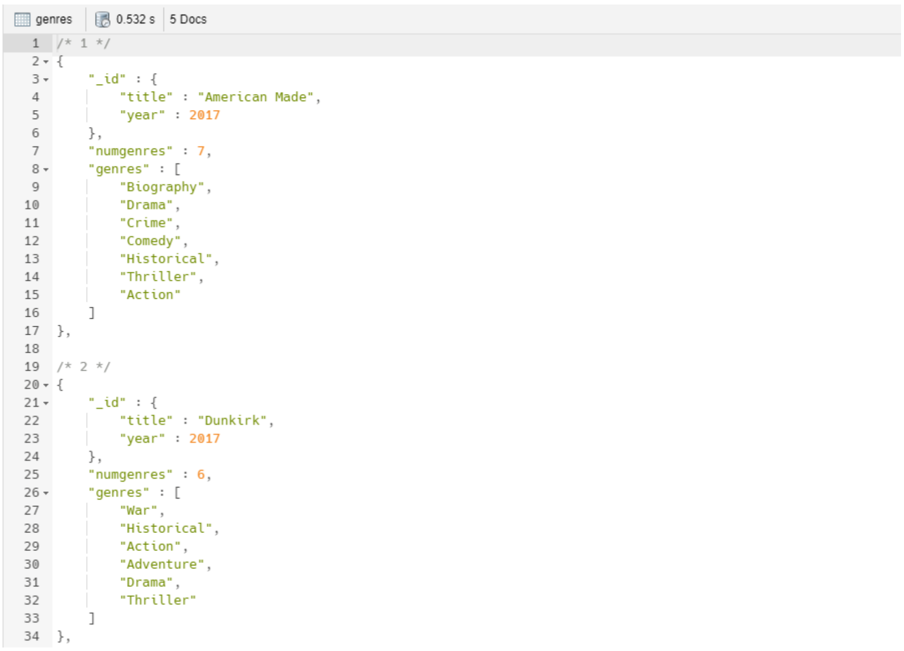


```
//EJERCICIO 23
var query27 = {"_id":{"title":"$title", "year":"$year"}, "genres":{$addToSet:"$genres"}}
var fase1 = {$group:query27}
var query28 = {"_id":1, "title":"$id.title", "numgenres":{$size:"$genres"}, "genres":"$genres"}
var fase2 = {$project:query28}
var fase3 = {$sort:{"numgenres":-1}}
var fase4 = {$limit: 5}
var etapas = [fase1, fase2, fase3, fase4]
db.genres.aggregate(etapas)
```

Aggregate × | Aggregate (1) × | Aggregate (2) × | Result × | Aggregate (3) × | Aggregate (4) × | Aggregate (5) × | Aggregate (6) ×

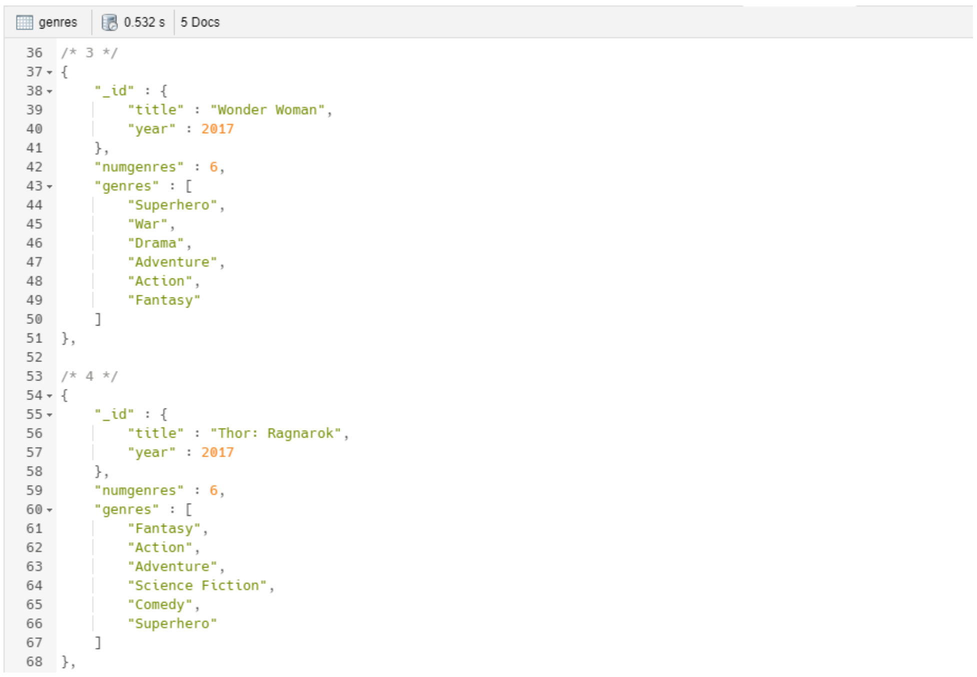
genres | 0.532 s | 5 Docs

	_id		numgenres	genres
	title	year		
1	American Made	2017	7	Array[7]
2	Dunkirk	2017	6	Array[6]
3	Wonder Woman	2017	6	Array[6]
4	Thor: Ragnarok	2017	6	Array[6]
5	My Little Pony: The Movie	2017	6	Array[6]



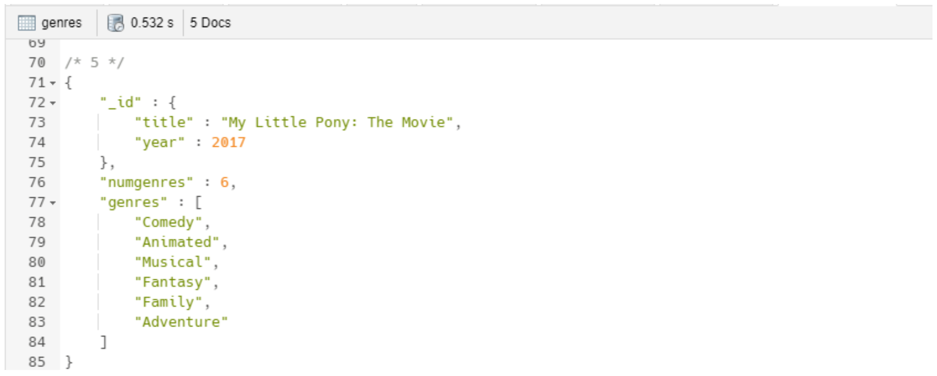
genres | 0.532 s | 5 Docs

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

/* 2 */
{
    "_id" : {
        "title" : "Dunkirk",
        "year" : 2017
    },
    "numgenres" : 6,
    "genres" : [
        "War",
        "Historical",
        "Action",
        "Adventure",
        "Drama",
        "Thriller"
    ]
},
```



genres | 0.532 s | 5 Docs

```
/* 3 */
{
    "_id" : {
        "title" : "Wonder Woman",
        "year" : 2017
    },
    "numgenres" : 6,
    "genres" : [
        "Superhero",
        "War",
        "Drama",
        "Adventure",
        "Action",
        "Fantasy"
    ]
},

/* 4 */
{
    "_id" : {
        "title" : "Thor: Ragnarok",
        "year" : 2017
    },
    "numgenres" : 6,
    "genres" : [
        "Fantasy",
        "Action",
        "Adventure",
        "Science Fiction",
        "Comedy",
        "Superhero"
    ]
},
```



genres | 0.532 s | 5 Docs

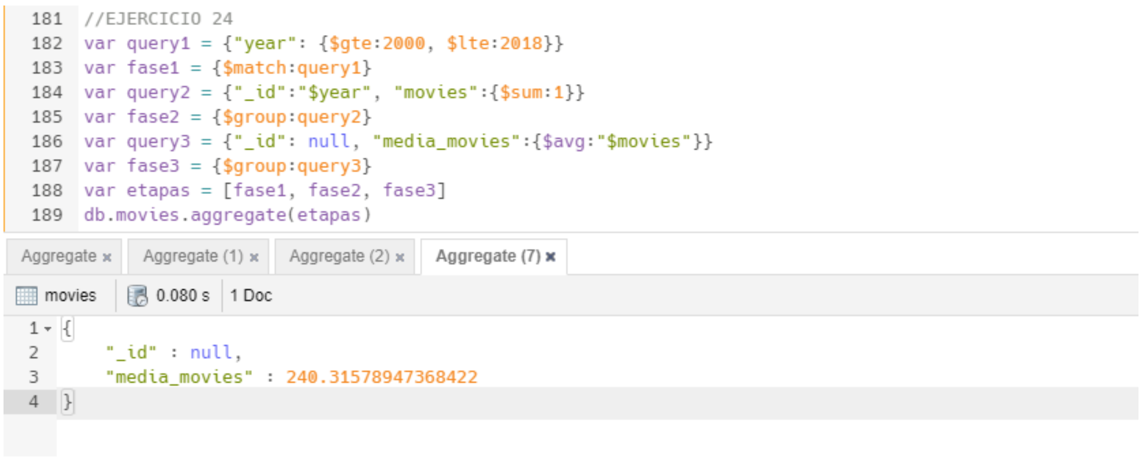
```
/* 5 */
{
    "_id" : {
        "title" : "My Little Pony: The Movie",
        "year" : 2017
    },
    "numgenres" : 6,
    "genres" : [
        "Comedy",
        "Animated",
        "Musical",
        "Fantasy",
        "Family",
        "Adventure"
    ]
}
```

EXERCISE 24. Calculate the average number of movies made per year in the 21st century (up to the most recent date in the collection).

Query:

```
var query1 = {"year": {$gte:2000, $lte:2018}}

var fase1 = {$match:query1}

var query2 = {"_id":"$year", "movies":{$sum:1}}

var fase2 = {$group:query2}

var query3 = {"_id": null, "media_movies":{$avg:"$movies"}}

var fase3 = {$group:query3}

var etapas = [fase1, fase2, fase3]

db.movies.aggregate(etapas)
```

Results:



```
//EJERCICIO 24
var query1 = {"year": {$gte:2000, $lte:2018}}
var fase1 = {$match:query1}
var query2 = {"_id":"$year", "movies":{$sum:1}}
var fase2 = {$group:query2}
var query3 = {"_id": null, "media_movies":{$avg:"$movies"}}
var fase3 = {$group:query3}
var etapas = [fase1, fase2, fase3]
db.movies.aggregate(etapas)
```

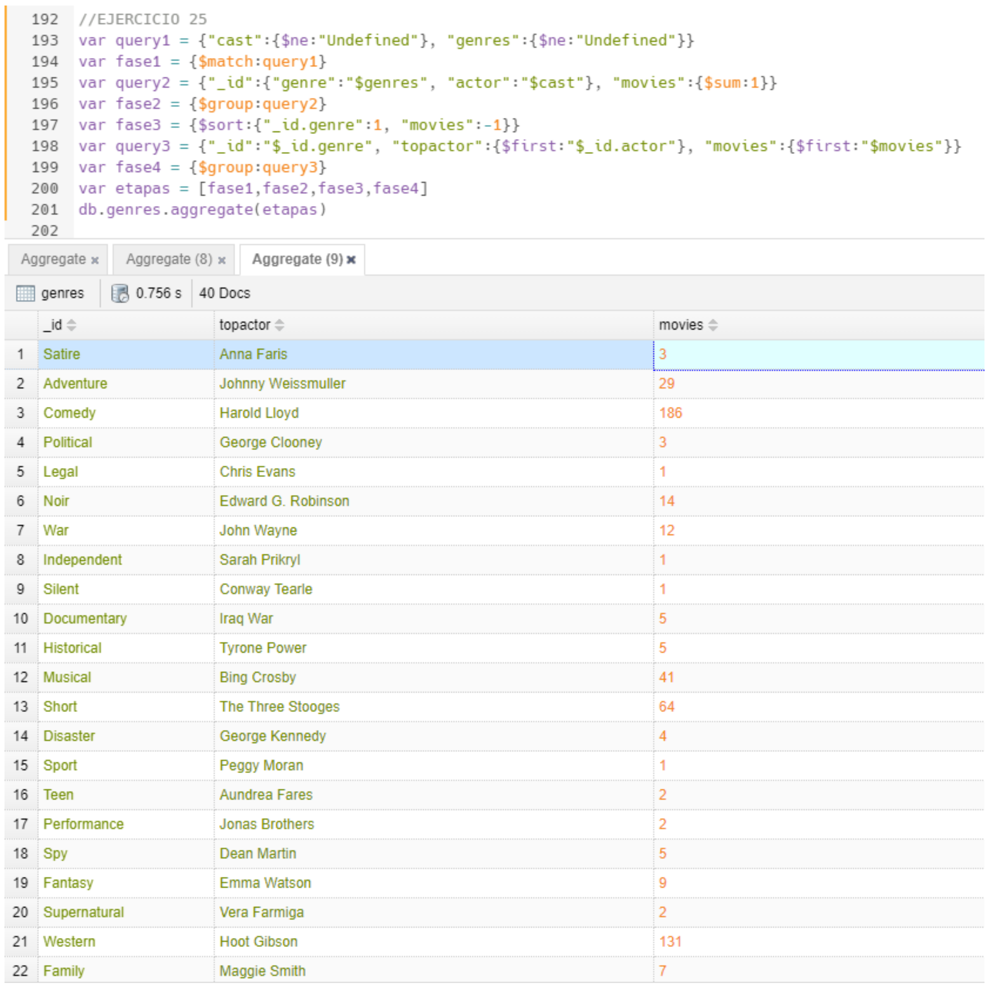
```
{
    "_id" : null,
    "media_movies" : 240.31578947368422
}
```

EXERCISE 25: Query which actor has made the most movies in each movie genre, showing the number of movies they have made.

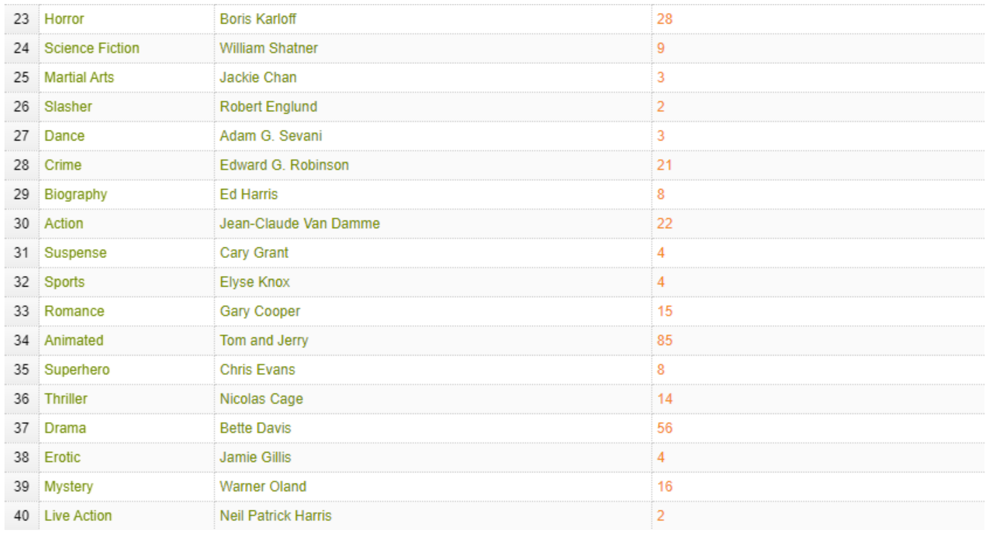
Query:

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

var fase1 = {$match:query1}

var query2 = {"_id":{"genre":"$genres", "actor":"$cast"}, "movies":{$sum:1}}

var fase2 = {$group:query2}

var fase3 = {$sort:{"_id.genre":1, "movies":-1}}

var query3 = {"_id":"$_id.genre", "topactor":{$first:"$_id.actor"},
"movies":{$first:"$movies"}}

var fase4 = {$group:query3}

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

db.genres.aggregate(etapas)
```

Results:



```
//EJERCICIO 25
var query1 = {"cast":{$ne:"Undefined"}, "genres":{$ne:"Undefined"}}
var fase1 = {$match:query1}
var query2 = {"_id":{"genre":"$genres", "actor":"$cast"}, "movies":{$sum:1}}
var fase2 = {$group:query2}
var fase3 = {$sort:{"_id.genre":1, "movies":-1}}
var query3 = {"_id":"$_id.genre", "topactor":{$first:"$_id.actor"}, "movies":{$first:"$movies"}}
var fase4 = {$group:query3}
var etapas = [fase1,fase2,fase3,fase4]
db.genres.aggregate(etapas)
```



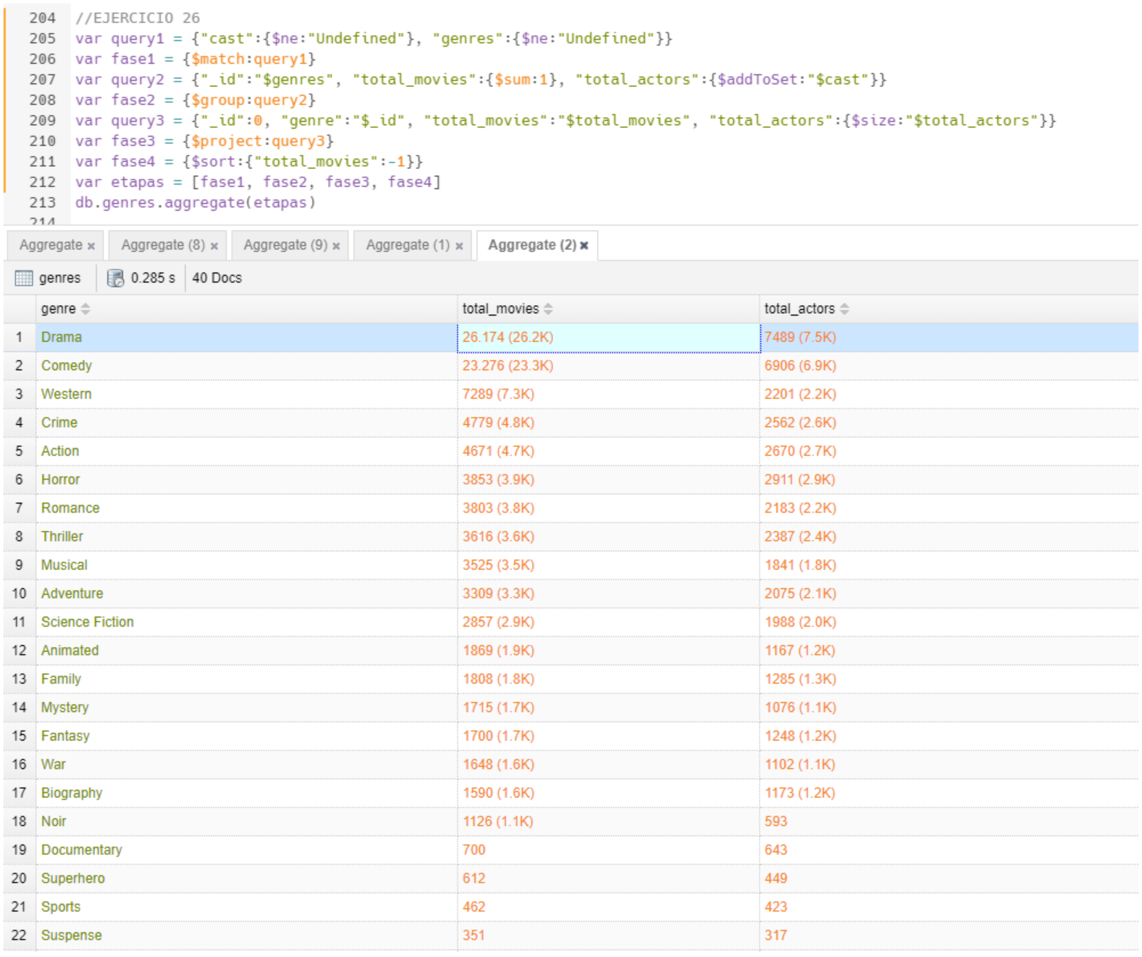
	_id	topactor	movies
1	Satire	Anna Faris	3
2	Adventure	Johnny Weissmuller	29
3	Comedy	Harold Lloyd	186
4	Political	George Clooney	3
5	Legal	Chris Evans	1
6	Noir	Edward G. Robinson	14
7	War	John Wayne	12
8	Independent	Sarah Prikryl	1
9	Silent	Conway Tearle	1
10	Documentary	Iraq War	5
11	Historical	Tyrone Power	5
12	Musical	Bing Crosby	41
13	Short	The Three Stooges	64
14	Disaster	George Kennedy	4
15	Sport	Peggy Moran	1
16	Teen	Aundrea Fares	2
17	Performance	Jonas Brothers	2
18	Spy	Dean Martin	5
19	Fantasy	Emma Watson	9
20	Supernatural	Vera Farmiga	2
21	Western	Hoot Gibson	131
22	Family	Maggie Smith	7
23	Horror	Boris Karloff	28
24	Science Fiction	William Shatner	9
25	Martial Arts	Jackie Chan	3
26	Slasher	Robert Englund	2
27	Dance	Adam G. Sevani	3
28	Crime	Edward G. Robinson	21
29	Biography	Ed Harris	8
30	Action	Jean-Claude Van Damme	22
31	Suspense	Cary Grant	4
32	Sports	Elyse Knox	4
33	Romance	Gary Cooper	15
34	Animated	Tom and Jerry	85
35	Superhero	Chris Evans	8
36	Thriller	Nicolas Cage	14
37	Drama	Bette Davis	56
38	Erotic	Jamie Gillis	4
39	Mystery	Warner Oland	16
40	Live Action	Neil Patrick Harris	2

EXERCISE 26: Calculate the total number of movies and actors who have participated in each movie genre, ordered in descending order by the number of movies.

Query:

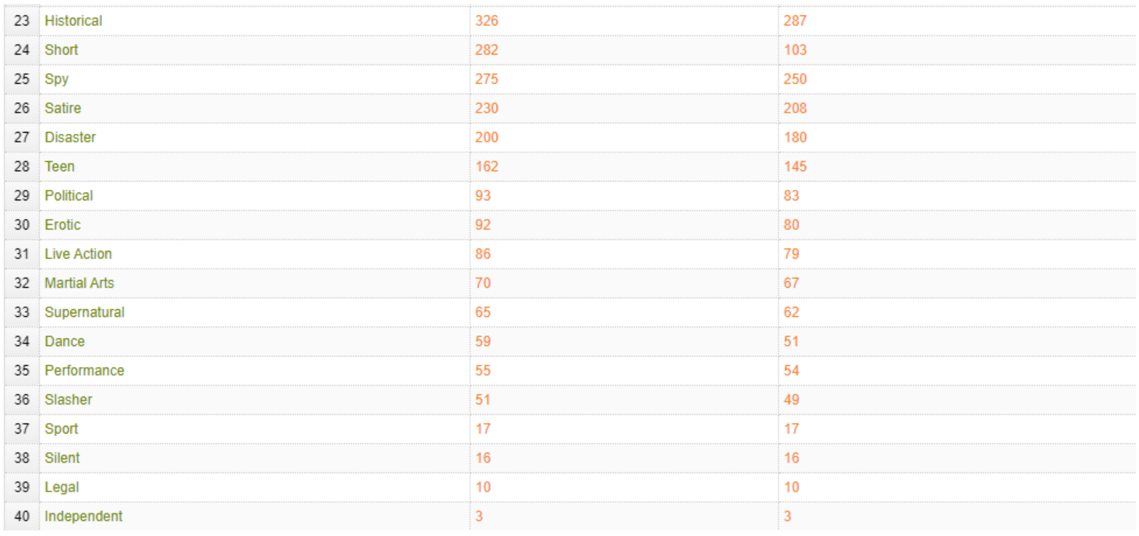
```
var query1 = {"cast":{$ne:"Undefined"}, "genres":{$ne:"Undefined"}}

var fase1 = {$match:query1}

var query2 = {"_id":"$genres", "total_movies":{$sum:1},
"total_actors":{$addToSet:"$cast"}}

var fase2 = {$group:query2}

var query3 = {"_id":0, "genre":"$_id", "total_movies":"$total_movies",
"total_actors":{$size:"$total_actors"}}

var fase3 = {$project:query3}

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

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

db.genres.aggregate(etapas)
```

Results:



```
//EJERCICIO 26
var query1 = {"cast":{$ne:"Undefined"}, "genres":{$ne:"Undefined"}}
var fase1 = {$match:query1}
var query2 = {"_id":"$genres", "total_movies":{$sum:1}, "total_actors":{$addToSet:"$cast"}}
var fase2 = {$group:query2}
var query3 = {"_id":0, "genre":"$_id", "total_movies":"$total_movies", "total_actors":{$size:"$total_actors"}}
var fase3 = {$project:query3}
var fase4 = {$sort:{"total_movies":-1}}
var etapas = [fase1, fase2, fase3, fase4]
db.genres.aggregate(etapas)
```

	genre	total_movies	total_actors
1	Drama	26.174 (26.2K)	7489 (7.5K)
2	Comedy	23.276 (23.3K)	6906 (6.9K)
3	Western	7289 (7.3K)	2201 (2.2K)
4	Crime	4779 (4.8K)	2562 (2.6K)
5	Action	4671 (4.7K)	2670 (2.7K)
6	Horror	3853 (3.9K)	2911 (2.9K)
7	Romance	3803 (3.8K)	2183 (2.2K)
8	Thriller	3616 (3.6K)	2387 (2.4K)
9	Musical	3525 (3.5K)	1841 (1.8K)
10	Adventure	3309 (3.3K)	2075 (2.1K)
11	Science Fiction	2857 (2.9K)	1988 (2.0K)
12	Animated	1869 (1.9K)	1167 (1.2K)
13	Family	1808 (1.8K)	1285 (1.3K)
14	Mystery	1715 (1.7K)	1076 (1.1K)
15	Fantasy	1700 (1.7K)	1248 (1.2K)
16	War	1648 (1.6K)	1102 (1.1K)
17	Biography	1590 (1.6K)	1173 (1.2K)
18	Noir	1126 (1.1K)	593
19	Documentary	700	643
20	Superhero	612	449
21	Sports	462	423
22	Suspense	351	317



23	Historical	326	287
24	Short	282	103
25	Spy	275	250
26	Satire	230	208
27	Disaster	200	180
28	Teen	162	145
29	Political	93	83
30	Erotic	92	80
31	Live Action	86	79
32	Martial Arts	70	67
33	Supernatural	65	62
34	Dance	59	51
35	Performance	55	54
36	Slasher	51	49
37	Sport	17	17
38	Silent	16	16
39	Legal	10	10
40	Independent	3	3