

Tarea NO SQL

Documento de resolución de la tarea - Unidad 3: NO SQL

Profesor: Alvaro Bravo

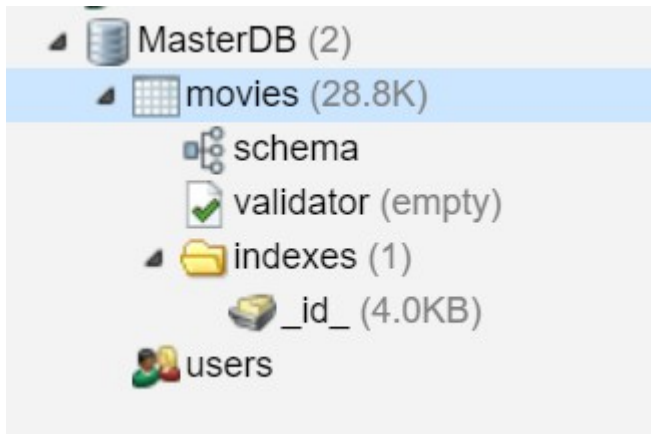
Alumno: Pablo Paez



UNIVERSIDAD
COMPLUTENSE
DE MADRID

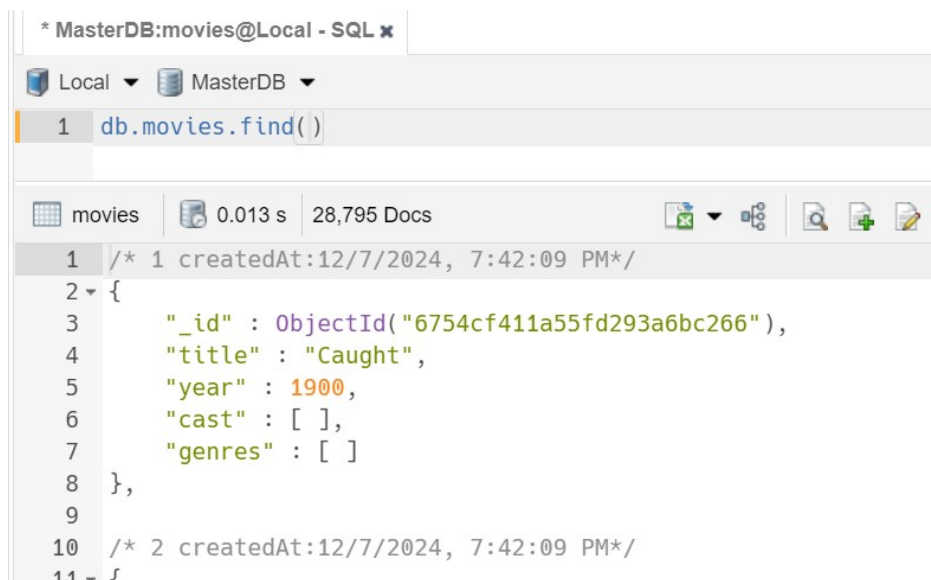


0 - Importar dataset



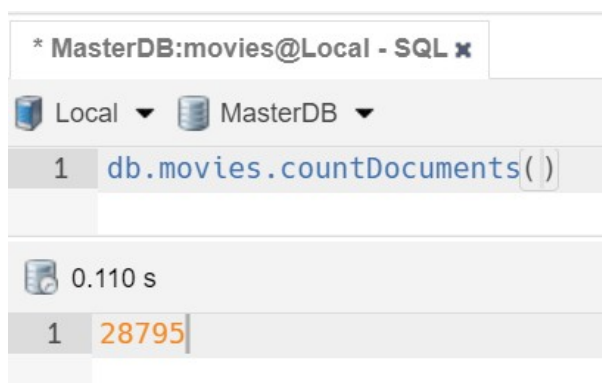
1 - Analizar con find la colección

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



2 - Contar cuántos documentos (películas) tiene cargado.

```
db.movies.countDocuments()
```



3. Insertar una película

```
db.movies.insertOne({"cast": "emma watson", "genres": "fantasy", "title": "Harry Potter"})
```

```
7 //3
8 db.movies.insertOne({"cast": "emma watson", "genres": "fantasy", "title": "Harry Potter 1"})
```

Result x Result (1) x

0.120 s

JSON

```
1 {
2   "acknowledged" : true,
3   "insertedId" : ObjectId("67561ada1a55fd293a6c32e1")
4 }
```

4. Borrar la película insertada en el punto anterior

```
db.movies.deleteOne({"title": "Harry Potter 1"})
```

```
10 //4
11 db.movies.deleteOne({"title": "Harry Potter 1"})
```

Result x Result (1) x Result (2) x

0.088 s

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

5. Contar cuantas películas tienen actores (cast) que se llaman “and”

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

```
13 //5
14 db.movies.find({"cast": {$regex: / and /}}).count()
15
16 //6
```

Result x	Find x	Result (1) x
0.069 s		
1	153	

6. Actualizar los documentos cuyo actor (cast) tenga por error el valor “and”

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

```
16 //6
17 db.movies.updateMany({"cast": {$regex: / and /}},{$pull:{"cast": " and "}})
```

Result x	Find x	Result (1) x	Error x	Result (2) x
0.078 s				
1	{			
2	"acknowledged" : true,			
3	"matchedCount" : 153,			
4	"modifiedCount" : 0			
5	}			

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

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

```
19 //7
20 db.movies.find({"cast": []}).count()
```

Result x	Find x	Result (1) x	Error x	Result (2) x	Result
0.046 s					
1	986				

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.

```
db.movies.updateMany({"cast": []},{$set: {"cast": [undefined]}})
```

```
22 //8
23 db.movies.updateMany({"cast": []},{$set: {"cast": [undefined]}})
```

Error x Result x

0.134 s

```
1 {
2   "acknowledged" : true,
3   "matchedCount" : 986,
4   "modifiedCount" : 986
5 }
```

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

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

```
25 //9
26 db.movies.find({"genres": []}).count()
```

Error x Result x Result (1) x Error (1) x Result (2) x

0.040 s

```
1 901
```

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

```
db.movies.updateMany({"genres": []},{$set: {"genres": [undefined]}})
```

```

28 //10
29 db.movies.updateMany({"genres":[]},{ $set:{"genres": [undefined]}})

```

Error x	Result x	Result (1) x	Error (1) x	Result (2) x	Result (3) x	Result (4) x
0.071 s						
1	{					
2	"acknowledged" : true,					
3	"matchedCount" : 901,					
4	"modifiedCount" : 901					
5	}					

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

```
db.movies.aggregate({$match: {year: {$exists: true}}},{ $sort: {year:-1}},
{$limit: 1})
```

```

31 //11
32 db.movies.aggregate({$match: {year: {$exists: true}}},{ $sort: {year:-1}}, {$limit: 1})

```

Error x	Result x	Result (1) x	Error (1) x	Result (2) x	Result (3) x	Result (4) x	Aggregate x	Aggregate (1) x
movies 0.101 s 1 Doc								
1	{							
2	"_id" : ObjectId("6754cf431a55fd293a6c31f4"),							
3	"title" : "Insidious: The Last Key",							
4	"year" : 2018,							
5	"cast" : [
6	"Lin Shaye",							
7	"Angus Sampson",							
8	"Leigh Whannell",							
9	"Spencer Locke",							
10	"Caitlin Gerard",							
11	"Kirk Acevedo",							
12	"Bruce Davison"							
13],							
14	"genres" : ["Horror", "Thriller"]							
15	}							

12. Contar cuántas películas han salido en los últimos 20 años

```
db.movies.aggregate([
  {
    $match: {
      year:{$gte: 1998}
    }
  },
  {
    $count: "last20yearsNumberOfFilms"
  }
])
```

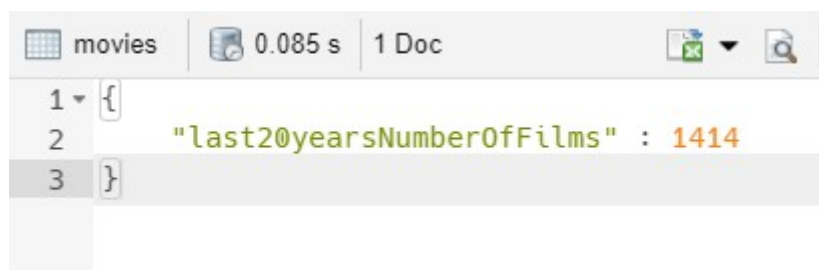


13. Contar cuántas películas han salido en la década de los 60 (del 60 al 69 incluidos)

```

db.movies.aggregate([
  {
    $match: {
      $and: [{year: {$gte: 1960}}, {year: {$lte: 1969}}]
    }
  },
  {
    $count: "last20yearsNumberOfFilms"
  }
])

```



14. Mostrar el año u años con más películas mostrando el número de películas de ese año.

```

db.movies.aggregate([
  {
    $group: {
      _id: "$year",
      moviesNumber: { $sum: 1 }
    }
  },
  {
    $sort: {"moviesNumber": -1}
  },
  {
    $group: {
      _id: null,
      maxNumberMovies: { $first: "$moviesNumber" }
    }
  }
])

```

```

{
  $project: {
    maxNumberMovies: 1
  }
},
{
  $lookup: {
    from: "movies",
    let: {maxNumberMovies: "$maxNumberMovies" },
    pipeline: [
      {
        $group: {
          _id: "$year",
          moviesNumber: { $sum: 1 }
        }
      },
      {
        $match: {
          $expr: { $eq: ['$moviesNumber', "$$
$maxNumberMovies" ] } },
      },
      {
        $project: {
          moviesNumber : false
        }
      }
    ],
    as: "years"
  }
}
]
)

```

	movies	0.087 s	1 Doc
1	{		
2	"_id" : null,		
3	"maxNumberMovies" : 634,		
4	"years" : [
5	{		
6	"_id" : 1919		
7	}		
8]		
9	}		

15. Mostrar el año u años con menos películas mostrando el número de películas de ese año.

```
db.movies.aggregate([
  {
    $match:{
      year : {$ne:null}
    }
  },
  {
    $group: {
      _id: "$year",
      moviesNumber: { $sum: 1 }
    }
  },
  {
    $sort: {"moviesNumber":1}
  },
  {
    $group: {
      _id :null,
      minNumberMovies: { $first: "$moviesNumber" }
    }
  },
  {
    $project: {
      minNumberMovies: 1
    }
  },
  {
    $lookup: {
      from: "movies",
      let: {minNumberMovies: "$minNumberMovies" },
      pipeline: [
        {
          $group: {
            _id: "$year",
            moviesNumber: { $sum: 1 }
          }
        },
        {
          $match: {
            $expr: { $eq: ['$moviesNumber', "$minNumberMovies" ] },
          }
        },
        {
          $project: {
            moviesNumber : false
          }
        }
      ],
      as: "years"
    }
  }
])
```



```
1 {
2   "_id" : null,
3   "minNumberMovies" : 7,
4   "years" : [
5     {
6       "_id" : 1906
7     },
8     {
9       "_id" : 1902
10    },
11    {
12      "_id" : 1907
13    }
14  ]
15 }
```

16. Guardar en nueva colección llamada “actors” realizando la fase \$unwind por actor

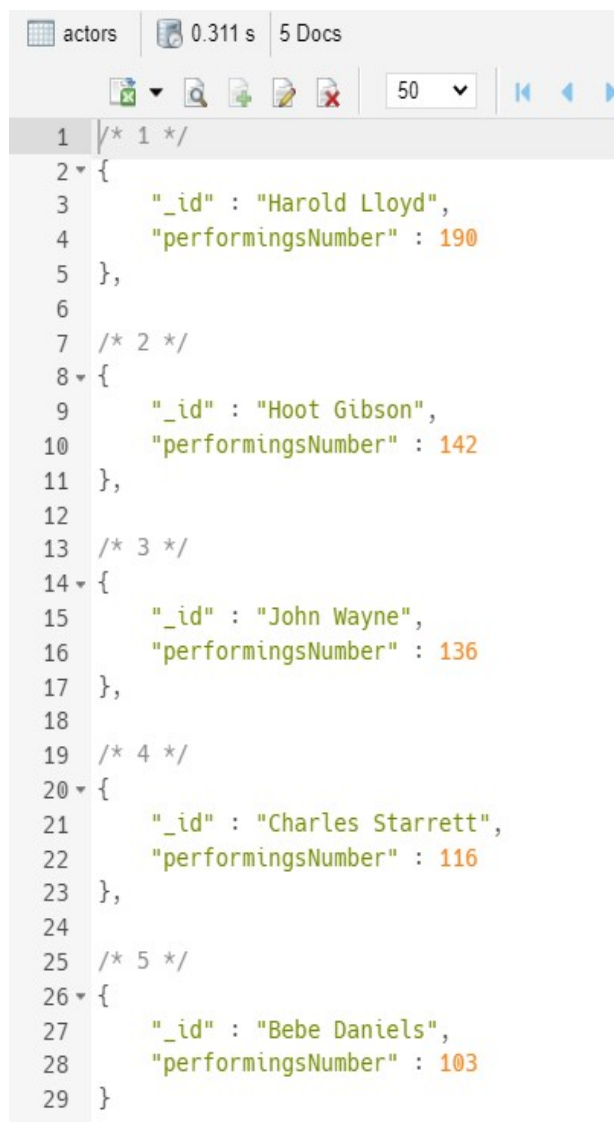
```
db.movies.aggregate(
  { $unwind : "$cast" },
  {
    $project: {_id:0}
  },
  {
    $out: "actors"
  }
)
```

```
db.actors.countDocuments()
```

Result x	Error x	Result (1
0.044 s		
1	83321	

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.

```
db.actors.aggregate(  
  { $match : {"cast": {$ne: null}}},  
  { $unwind: "$cast"},  
  { $group: { "_id": "$cast",  
              "performingsNumber": {$sum: 1}  
            }  
    },  
  {  
    $sort: {"performingsNumber": -1}  
  },  
  {  
    $limit: 5  
  }  
)
```



```
actors 0.311 s 5 Docs  
1 /* 1 */  
2 {  
3   "_id" : "Harold Lloyd",  
4   "performingsNumber" : 190  
5 },  
6  
7 /* 2 */  
8 {  
9   "_id" : "Hoot Gibson",  
10  "performingsNumber" : 142  
11 },  
12  
13 /* 3 */  
14 {  
15   "_id" : "John Wayne",  
16   "performingsNumber" : 136  
17 },  
18  
19 /* 4 */  
20 {  
21   "_id" : "Charles Starrett",  
22   "performingsNumber" : 116  
23 },  
24  
25 /* 5 */  
26 {  
27   "_id" : "Bebe Daniels",  
28   "performingsNumber" : 103  
29 }
```

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

```
db.actors.aggregate(  
  { $match : { "cast": { $ne: null } } },  
  {  
    $group: {  
      _id: { "title": "$title", "year": "$year" },  
      actorsNumber: { $sum : 1 }  
    }  
  },  
  {  
    $sort: { "actorsNumber": -1 }  
  },  
  {  
    $limit: 5  
  }  
)
```



```
1 /* 1 */  
2 {  
3   "_id" : {  
4     "title" : "The Twilight Saga: Breaking Dawn - P  
5     "year" : 2012  
6   },  
7   "actorsNumber" : 35  
8 },  
9  
10 /* 2 */  
11 {  
12   "_id" : {  
13     "title" : "Anchorman 2: The Legend Continues",  
14     "year" : 2013  
15   },  
16   "actorsNumber" : 33  
17 },  
18  
19 /* 3 */  
20 {  
21   "_id" : {  
22     "title" : "Cars 2",  
23     "year" : 2011  
24   },  
25   "actorsNumber" : 32  
26 },  
27  
28 /* 4 */  
29 {  
30   "_id" : {  
31     "title" : "Avengers: Infinity War",  
32     "year" : 2018  
33   },  
34   "actorsNumber" : 29  
35 }
```

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

```
db.actors.aggregate(  
  { $match :  
    { $and: [  
      {"cast": {$ne: null}},  
      {"cast": {$ne: "and"}}  
    ]  
  },  
  {  
    $group: { _id: "$cast",  
      startCareer: {$min: "$year"},  
      endCareer: {$max: "$year"}  
    }  
  },  
  {  
    $project: {  
      _id: 0,  
      actorName: "$_id"  
      startCareer: "$startCareer",  
      endCareer: "$endCareer",  
      yearsOfWork: {$subtract: ["$endCareer", "$startCareer"]} }  
    }  
  },  
  { $sort: {"yearsOfWork" : -1} },  
  { $limit: 5 }  
})
```

actors	0.250 s	5 Docs
1	/* 1 */	
2	{	
3	"actorName" : "Harrison Ford",	
4	"startCareer" : 1919,	
5	"endCareer" : 2017,	
6	"yearsOfWork" : 98	
7	},	
8		
9	/* 2 */	
10	{	
11	"actorName" : "Gloria Stuart",	
12	"startCareer" : 1932,	
13	"endCareer" : 2012,	
14	"yearsOfWork" : 80	
15	},	
16		
17	/* 3 */	
18	{	
19	"actorName" : "Kenny Baker",	
20	"startCareer" : 1937,	
21	"endCareer" : 2012,	
22	"yearsOfWork" : 75	
23	},	
24		
25	/* 4 */	
26	{	
27	"actorName" : "Lillian Gish",	
28	"startCareer" : 1912,	
29	"endCareer" : 1987,	
30	"yearsOfWork" : 75	
31	},	
32		
33	/* 5 */	
34	{	
35	"actorName" : "Angela Lansbury",	
36	"startCareer" : 1944,	
37	"endCareer" : 2018,	
38	"yearsOfWork" : 74	
39	}	

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

```
db.movies.aggregate(  
  { $unwind : "$genres" },  
  {  
    $project: {_id:0}  
  },  
  {  
    $out: "genres"  
  }  
)
```

```
db.genres.countDocuments()
```



The screenshot shows the MongoDB command line interface with three tabs: « (9) x, Error (12) x, and Aggregate (10). The main window displays the result of the countDocuments command, which is 33914. The result is shown in a table with one row and one column, with the value 33914 highlighted in orange.

1 33914

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.

```
db.genres.aggregate(  
  {  
    $group: {  
      _id: ["$genres", "$year"]  
      moviesNumber : {$sum:1}  
    }  
  },  
  {$sort:{"moviesNumber":-1 }},  
  {$limit:5}  
)
```

```

2 ▾ {
3     "_id" : [ "Drama", 1919 ],
4     "moviesNumber" : 292
5 },
6
7 /* 2 */
8 ▾ {
9     "_id" : [ "Drama", 1925 ],
10    "moviesNumber" : 247
11 },
12
13 /* 3 */
14 ▾ {
15     "_id" : [ "Drama", 1924 ],
16     "moviesNumber" : 233
17 },
18
19 /* 4 */
20 ▾ {
21     "_id" : [ "Comedy", 1919 ],
22     "moviesNumber" : 227
23 },
24
25 /* 5 */
26 ▾ {
27     "_id" : [ "Drama", 1922 ],
28     "moviesNumber" : 209
29 }

```

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

```

db.genres.aggregate
(
    {
        $match :
        { $and: [
            {"cast": {$ne: null}},
            {"cast": {$ne: "and"}},
            {"genres": {$ne: null}}
        ]
        }
    }
    {
        $unwind: "$cast"
    }
)

```

```

    $group: {
      _id: "$cast",
      genres: {$addToSet: "$genres"}
    }
  }
  {
    $project: {
      genres: "$genres"
      genresCount: {$size: "$genres" }
    }
  }
  {$sort: {"genresCount": -1}}
  {$limit: 5}
)

```

genres 0.192 s 5 Docs			
	_id *	genres	genresCount
1	Dennis Quaid	Array[20]	20
2	Michael Peña	Array[18]	18
3	James Mason	Array[18]	18
4	Michael Caine	Array[18]	18
5	Colin Farrell	Array[18]	18

genres 0.192 s 5 Docs			
1	/* 1 */		
2	{		
3	"_id" : "Dennis Quaid",		
4	"genres" : [
5	"Thriller",		
6	"Biography",		
7	"Fantasy",		
8	"Family",		
9	"Horror",		
10	"Science Fiction",		
11	"Action",		
12	"Suspense",		
13	"Sports",		
14	"Romance",		
15	"Adventure",		
16	"Crime",		
17	"Drama",		
18	"Comedy",		
19	"Satire",		
20	"Western",		
21	"Animated",		
22	"Disaster",		
23	"Musical",		
24	"Dance"		
25],		
26	"genresCount" : 20		
27	},		
28			
29	/* 2 */		
30	{		
31	"_id" : "Michael Peña",		
32	"genres" : [
33	"War",		
34	"Martial Arts",		
35	"Animated",		
36	"Musical",		
37	"Drama",		
38	"Crime",		
39	"Superhero"		

23. Sobre géneros (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

```
db.genres.aggregate(
  {
    $match : {"genres": {$ne: null}}
  },
  {
    $group: {
      _id: {"title": "$title", "year": "$year"},
      genres: {$addToSet: "$genres"}
    }
  },
  {
    $project: {
      genres: "$genres"
      genresCount: {$size: "$genres" }
    }
  },
  {$sort: {"genresCount": -1}}
  {$limit: 5}
)
```

genres 0.236 s 5 Docs			
	_id		genres
	title	year	
1	American Made	2017	Array[7]
2	The Dark Tower	2017	Array[6]
3	Wonder Woman	2017	Array[6]
4	Thor: Ragnarok	2017	Array[6]
5	Dunkirk	2017	Array[6]

```
genres 0.236 s 5 Docs
1 /* 1 */
2 {
3   "_id" : {
4     "title" : "American Made",
5     "year" : 2017
6   },
7   "genres" : [
8     "Drama",
9     "Historical",
10    "Crime",
11    "Action",
12    "Biography",
13    "Thriller",
14    "Comedy"
15  ],
16   "genresCount" : 7
17 },
18
19 /* 2 */
20 {
21   "_id" : {
22     "title" : "The Dark Tower",
23     "year" : 2017
24   },
25   "genres" : [
```

24. Devolver el promedio del numero de actores que participaron en las películas producidas en cada año, ordenadas de manera decreciente por el nro de actores promedio

```
db.movies.aggregate(  
  {  
    $match :  
    { $and: [  
      {"cast": {$ne: null}},  
      {"cast": {$ne: "and"}},  
      {"genres": {$ne: null}},  
      { "year": {$ne: null}}  
    ]  
  }  
  { $group: { _id: "$year",  
    averageCastNumber: {$avg:{$cond: [{ $isArray: "$cast" }, {$size:  
"$cast"},1 ]}}  
  }  
  {  
    $project: {  
      _id: 0,  
      year: "$_id"  
      averageCastNumber: { $toDouble: "$averageCastNumber" }  
    }  
  }  
  {  
    $sort: {averageCastNumber:-1}  
  }  
})
```

	movies	0.294 s	114 Docs
	year	averageCastNumber	
1	2011	8.476744186046512	
2	2012	7.488636363636363	
3	2017	6.770212765957447	
4	2010	6.636363636363637	
5	2018	6.391489361702128	
6	2013	5.933139534883721	
7	2009	5.393805309734513	
8	1998	4.152892561983471	

25 Devolver el promedio del numero de actores que participaron en las películas agrupados por año y genero, ordenadas de manera decreciente por el nro de actores promedio.

```

db.movies.aggregate(
  {
    $match :
    { $and: [
      {"cast": {$ne: null}},
      {"cast": {$ne: "and"}},
      {"genres": {$ne: null}},
      { "year": {$ne: null}}
    ]
    }
  }
  { $unwind: "$genres" }
  { $group: { _id: "$genres",
    averageCastNumber: {$avg:{$cond: [{ $isArray: "$cast" }, {$size:
"$cast"},1 ]}}
  }
  }
  {
    $project: {
      _id :0,
      genre: "$_id"
      averageCastNumber: { $toInt:"$averageCastNumber" }
    }
  }
  {
    $sort: {averageCastNumber:-1}
  }
)

```

movies 0.444 s 40 Docs		
	genre	averageCastNumber
1	Legal	10
2	Live Action	9
3	Teen	7
4	Superhero	7
5	Political	6
6	Disaster	6
7	Supernatural	5
8	Fantasy	5

26 Devolver el nro de actores promedio para las películas agrupadas por genero y año ordenados por genero y año

```
db.movies.aggregate(  
  {  
    $match :  
    { $and: [  
      {"cast": {$ne: null}},  
      {"cast": {$ne: "and"}},  
      {"genres": {$ne: null}},  
      { "year": {$ne: null}}  
    ]  
  }  
  { $unwind: "$genres"  
  { $group: { _id: [{genres:"$genres"}, {year: "$year"}],  
    averageCastNumber: {$avg:{$cond: [{ $isArray: "$cast" }, {$size:  
"$cast"},1 ]}  
  }  
  }  
  {  
    $project: {  
      _id: 0,  
      genre: { $arrayElemAt: ["$_id", 0] },  
      year: { $arrayElemAt: ["$_id", 1] },  
      averageCastNumber: { $toInt:"$averageCastNumber" }  
    }  
  }  
  {  
    $project: {  
      genre: "$genre.genres",  
      year: "$year.year",  
      averageCastNumber: 1  
    }  
  }  
  {  
    $sort: {"genre": -1,"year":-1}  
  }  
})
```

	movies	0.345 s	Fetch Count
	averageCastNumber	genre	year
1	4	Western	2018
2	7	Western	2017
3	2	Western	2016
4	2	Western	2015
5	6	Western	2013
6	15	Western	2012
7	8	Western	2011
8	6	Western	2010
9	3	Western	2009