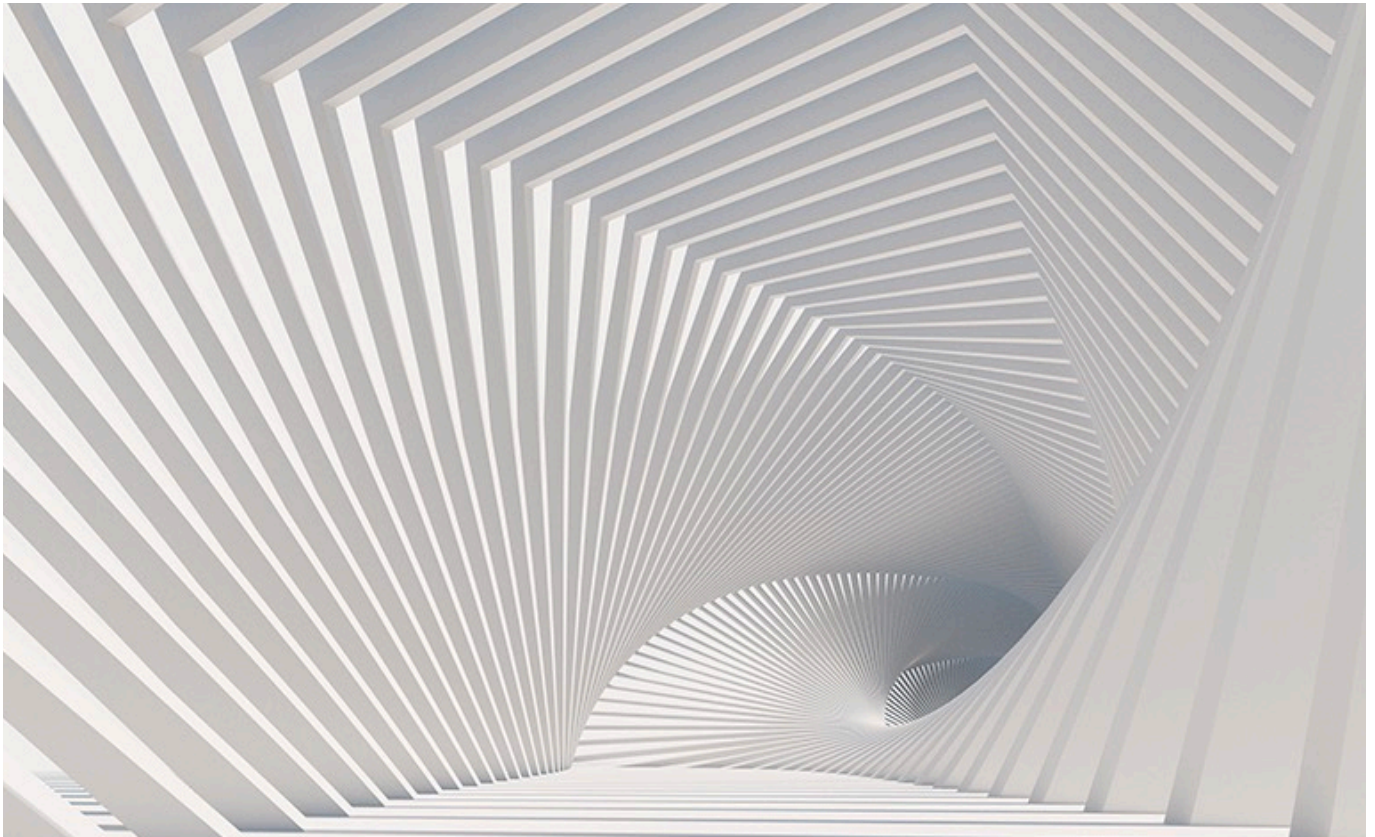


# TAREA BASES DE DATOS NO SQL



# ÍNDICE

<b>Ejercicio 1</b>	<b>4</b>
<b>Ejercicio 2</b>	<b>4</b>
<b>Ejercicio 3</b>	<b>5</b>
<b>Ejercicio 4</b>	<b>5</b>
<b>Ejercicio 5</b>	<b>6</b>
<b>Ejercicio 6</b>	<b>6</b>
<b>Ejercicio 7</b>	<b>7</b>
<b>Ejercicio 8</b>	<b>7</b>
<b>Ejercicio 9</b>	<b>8</b>
<b>Ejercicio 10</b>	<b>8</b>
<b>Ejercicio 11</b>	<b>9</b>
<b>Ejercicio 12</b>	<b>10</b>
<b>Ejercicio 13</b>	<b>11</b>
<b>Ejercicio 14</b>	<b>12</b>
<b>Ejercicio 15</b>	<b>13</b>
<b>Ejercicio 16</b>	<b>14</b>
<b>Ejercicio 17</b>	<b>15</b>
<b>Ejercicio 18</b>	<b>16</b>
<b>Ejercicio 19</b>	<b>17</b>
<b>Ejercicio 20</b>	<b>18</b>
<b>Ejercicio 21</b>	<b>19</b>
<b>Ejercicio 22</b>	<b>20</b>
<b>Ejercicio 23</b>	<b>21</b>
<b>Ejercicio 24</b>	<b>22</b>

<b>Ejercicio 25</b>	<b>23</b>
<b>Ejercicio 26</b>	<b>24</b>

## Ejercicio 1

Analizar con find la colección.

`db.movies.find()`

```
_id: ObjectId('663bfc4af1719db9e08e89c1')
title: "Caught"
year: 1900
▶ cast: Array (empty)
▶ genres: Array (empty)
```

```
_id: ObjectId('663bfc4af1719db9e08e89c2')
title: "After Dark in Central Park"
year: 1900
▶ cast: Array (empty)
▶ genres: Array (empty)
```

```
_id: ObjectId('663bfc4af1719db9e08e89c3')
title: "Buffalo Bill's Wild West Parad"
year: 1900
▶ cast: Array (empty)
▶ genres: Array (empty)
```

```
_id: ObjectId('663bfc4af1719db9e08e89c4')
title: "The Enchanted Drawing"
year: 1900
▶ cast: Array (empty)
▶ genres: Array (empty)
```

```
_id: ObjectId('663bfc4af1719db9e08e89c5')
title: "Clowns Spinning Hats"
year: 1900
▶ cast: Array (empty)
▶ genres: Array (empty)
```

```
> show databases
< TareaNoSQL      2.34 MiB
  admin           40.00 KiB
  config          108.00 KiB
  local           40.00 KiB
> use TareaNoSQL
< switched to db TareaNoSQL
> db.movies.find()
< {
  _id: ObjectId('663bfc4af1719db9e08e89c1'),
  title: 'Caught',
  year: 1900,
  cast: [],
  genres: []
}
```

## Ejercicio 2

Contar cuántos documentos (películas) tiene cargado.

`db.movies.countDocuments()`

```
> db.movies.countDocuments()
< 28795
```

### Ejercicio 3

Insertar una película.

```
db.movies.insertOne({ title: "titulo prueba", year: 1900, cast: [], genres: []})
db.movies.find({ title: "titulo prueba"})
```

```
> db.movies.insertOne({ title: "titulo prueba", year: 1900, cast: [], genres: []})
< {
  acknowledged: true,
  insertedId: ObjectId('663c0507f0ad177b485a8a21')
}
> db.movies.find({ title: "titulo prueba"})
< {
  _id: ObjectId('663c0507f0ad177b485a8a21'),
  title: 'titulo prueba',
  year: 1900,
  cast: [],
  genres: []
}
```

### Ejercicio 4

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

```
db.movies.deleteOne({ title: "titulo prueba"})
db.movies.find({ title: "titulo prueba"})
```

```
> db.movies.deleteOne({ title: "titulo prueba"})
< {
  acknowledged: true,
  deletedCount: 1
}
> db.movies.find({ title: "titulo prueba"})
<
```

## Ejercicio 5

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

```
db.movies.countDocuments({ cast: 'and'})
```

```
> db.movies.countDocuments({cast: 'and'})  
< 93
```

## Ejercicio 6

Actualizar los documentos cuyo actor (cast) tenga por error el valor “and” como si realmente fuera un actor. Para ello, se debe sacar únicamente 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.

```
var query = {cast: 'and'}  
db.movies.countDocuments(query)
```

```
var operación = {$pull: {cast: 'and'}}  
db.movies.updateMany(query, operación)
```

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

```
> db.movies.countDocuments({cast: 'and'})  
< 93  
> var query = {cast: 'and'}  
> db.movies.countDocuments(query)  
< 93  
> var operacion = {$pull: {cast: 'and'}}  
> db.movies.updateMany(query, operacion)  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 93,  
  modifiedCount: 93,  
  upsertedCount: 0  
}  
> db.movies.countDocuments(query)  
< 0
```

## Ejercicio 7

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

```
db.movies.countDocuments({ cast: []})
```

```
> db.movies.countDocuments({cast: []})  
< 986
```

## Ejercicio 8

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

```
var query = {cast: []}  
db.movies.countDocuments(query)
```

```
var operación = {$addToSet: {cast: "Undefined"}}  
db.movies.updateMany(query, operación)
```

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

```
> var query = {cast: []}  
> db.movies.countDocuments(query)  
< 986  
> var operacion = {$addToSet: {cast: "Undefined"}}  
> db.movies.updateMany(query, operacion)  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 986,  
  modifiedCount: 986,  
  upsertedCount: 0  
}  
> db.movies.countDocuments(query)  
< 0
```

## Ejercicio 9

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

```
db.movies.countDocuments({genres: []})
```

```
> db.movies.countDocuments({genres: []})  
< 901
```

## Ejercicio 10

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

```
var query = {genres: []}  
db.movies.countDocuments(query)
```

```
var operación = {$addToSet: {genres: "Undefined"}}  
db.movies.updateMany(query, operación)
```

```
db.movies.countDocuments(query)  
db.movies.countDocuments({genres: "Undefined"})
```

```
> var query = {genres: []}  
> db.movies.countDocuments(query)  
< 901  
> var operacion = {$addToSet: {genres: "Undefined"}}  
> db.movies.updateMany(query, operacion)  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 901,  
  modifiedCount: 901,  
  upsertedCount: 0  
}  
> db.movies.countDocuments(query)  
< 0  
> db.movies.countDocuments({genres: "Undefined"})  
< 901
```



## Ejercicio 11

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

```
db.movies.find().sort({year: -1})
```

```
> db.movies.find().sort({year: -1})
< {
  _id: ObjectId('663bfc4cf1719db9e08ef94f'),
  title: 'Insidious: The Last Key',
  year: 2018,
  cast: [
    'Lin Shaye',
    'Angus Sampson',
    'Leigh Whannell',
    'Spencer Locke',
    'Caitlin Gerard',
    'Kirk Acevedo',
    'Bruce Davison'
```

## Ejercicio 12

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

```
var query1 = { _id: null, latestYear: { $max: "$year" } }
var fase1 = { $group: query1 }
```

```
var query2 = { _id: 0, latestYear: 1, startYear: { $subtract: [ "$latestYear", 19 ] } }
var fase2 = { $project: query2 }
```

```
var query3 = { from: "movies",
  let: { startYear: "$startYear" },
  pipeline: [
    { $match: { $expr: { $gte: [ "$year", "$$startYear" ] } } },
    { $count: "totalMovies" } ],
  as: "moviesInLast20Years" }
```

```
var fase3 = { $lookup: query3 }
var fase4 = { $unwind: "$moviesInLast20Years" }
```

```
var query5 = { totalMovies: "$moviesInLast20Years.totalMovies" }
var fase5 = { $project: query5 }
var etapas = [ fase1, fase2, fase3, fase4, fase5 ]
db.movies.aggregate( etapas )
```

```
var query1 = { _id: null, latestYear: { $max: "$year" } }
var fase1 = { $group: query1 }

var query2 = { _id: 0, latestYear: 1, startYear: { $subtract: [ "$latestYear", 19 ] } }
var fase2 = { $project: query2 }

var query3 = { from: "movies",
  let: { startYear: "$startYear" },
  pipeline: [
    { $match: { $expr: { $gte: [ "$year", "$$startYear" ] } } },
    { $count: "totalMovies" } ],
  as: "moviesInLast20Years" }

var fase3 = { $lookup: query3 }
var fase4 = { $unwind: "$moviesInLast20Years" }

var query5 = { totalMovies: "$moviesInLast20Years.totalMovies" }
var fase5 = { $project: query5 }

var etapas = [ fase1, fase2, fase3, fase4, fase5 ]
db.movies.aggregate( etapas )
```

```
"totalMovies" : 4787
```

### Ejercicio 13

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

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

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

```
> var query1 = {year: { $gte: 1960, $lte: 1969 }}
> var fase1 = { $match: query1 }
> var query2 = { _id: null, count: { $sum: 1 } }
> var fase2 = { $group: query2 }
> var etapas = [fase1, fase2]
> db.movies.aggregate(etapas)
< {
  _id: null,
  count: 1414
}
```

## Ejercicio 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 query1 = { _id: "$year", count: { $sum: 1 } }
var fase1 = { $group : query1 }

var query2 = { count: -1 }
var fase2 = { $sort: query2 }

var query3 = { _id: null,
  maxPelis: { $first: "$count" },
  years: { $push: { year: "$_id", count: "$count" } } }
var fase3 = { $group: query3 }

var query4 = { years: { $filter: { input: "$years", as: "yearInfo",
  cond: { $eq: [ "$$yearInfo.count", "$maxPelis" ] } } }, _id: 0, maxPelis: 1 }

var fase4 = { $project: query4 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.movies.aggregate( etapas )
```

```
var query1 = { _id: "$year", count: { $sum: 1 } }
var fase1 = { $group : query1 }

var query2 = { count: -1 }
var fase2 = { $sort: query2 }

var query3 = { _id: null,
  maxPelis: { $first: "$count" },
  years: {
    $push: {
      year: "$_id",
      count: "$count"
    }
  }
}
var fase3 = { $group: query3 }

var query4 = { years: {
  $filter: {
    input: "$years",
    as: "yearInfo",
    cond: { $eq: [ "$$yearInfo.count", "$maxPelis" ] }
  }
},
  _id: 0,
  maxPelis: 1 }
var fase4 = { $project: query4 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.movies.aggregate( etapas )
```

```
{
  "maxPelis" : 634,
  "years" : [
    {
      "year" : 1919,
      "count" : 634
    }
  ]
}
```

## Ejercicio 15

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

```
var query1 = { _id: "$year", count: { $sum: 1 } }
var fase1 = { $group : query1 }

var query2 = { count: 1 }
var fase2 = { $sort: query2 }

var query3 = { _id: null,
  minPelis: { $first: "$count" },
  years: { $push: { year: "$_id", count: "$count" } } }
var fase3 = { $group: query3 }

var query4 = { years: { $filter: { input: "$years", as: "yearInfo",
  cond: { $eq: [ "$$yearInfo.count", "$minPelis" ] } } }, _id: 0, minPelis: 1 }

var fase4 = { $project: query4 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.movies.aggregate( etapas )
```

```
var query1 = { _id: "$year", count: { $sum: 1 } }
var fase1 = { $group : query1 }

var query2 = { count: 1 }
var fase2 = { $sort: query2 }

var query3 = { _id: null,
  minPelis: { $first: "$count" },
  years: {
    $push: {
      year: "$_id",
      count: "$count"
    }
  }
}
var fase3 = { $group: query3 }

var query4 = { years: {
  $filter: {
    input: "$years",
    as: "yearInfo",
    cond: { $eq: [ "$$yearInfo.count", "$minPelis" ] }
  }
},
  _id: 0,
  minPelis: 1 }

var fase4 = { $project: query4 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.movies.aggregate( etapas )
```

```
{
  "maxPelis" : 7,
  "years" : [
    {
      "year" : 1906,
      "count" : 7
    },
    {
      "year" : 1902,
      "count" : 7
    },
    {
      "year" : 1907,
      "count" : 7
    }
  ]
}
```

## Ejercicio 16

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

```
var fase1 = { $unwind: "$cast" }
var query2 = { "_id": 0 }
var fase2 = { $project: query2 }
var fase3 = { $out: "actors" }
var etapas = [ fase1, fase2, fase3 ]
db.movies.aggregate(etapas)
```

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

```
var fase1 = { $unwind: "$cast" }
var query2 = { "_id": 0 }
var fase2 = { $project: query2 }
var fase3 = { $out: "actors" }
var etapas = [ fase1, fase2, fase3 ]
db.movies.aggregate(etapas)

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

Key	Value	Type
(1) 66429c7b76f535c8ae07676b	{ title: "Caught", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(2) 66429c7b76f535c8ae07676c	{ title: "After Dark In Central Park", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(3) 66429c7b76f535c8ae07676d	{ title: "Buffalo Bill's Wild West Parade", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(4) 66429c7b76f535c8ae07676e	{ title: "The Enchanted Drawing", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(5) 66429c7b76f535c8ae07676f	{ title: "Clowns Spinning Hats", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(6) 66429c7b76f535c8ae076770	{ title: "Capture of Boer Battery by British", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(7) 66429c7b76f535c8ae076771	{ title: "Feeding Sea Lions", year: 1900, cast: "Paul Boyton", genres: [ "Undefined" ] } (5 fields)	Document
(8) 66429c7b76f535c8ae076772	{ title: "How to Make a Fat Wife Out of Two Lean Ones", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(9) 66429c7b76f535c8ae076773	{ title: "Searching Ruins on Broadway, Galveston, for Dead Bodies", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(10) 66429c7b76f535c8ae076774	{ title: "The Tribulations of an Amateur Photographer", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(11) 66429c7b76f535c8ae076775	{ title: "Trouble in Hogan's Alley", year: 1900, cast: "Undefined", genres: [ "Comedy" ] } (5 fields)	Document
(12) 66429c7b76f535c8ae076776	{ title: "Boarding School Girls' Pajama Parade", year: 1900, cast: "Undefined", genres: [ "Undefined" ] } (5 fields)	Document
(13) 66429c7b76f535c8ae076777	{ title: "Two Old Sparks", year: 1900, cast: "Undefined", genres: [ "Short" ] } (5 fields)	Document

0.057 s
1 83224

## Ejercicio 17

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

```
var query1 = {cast: { $ne: "Undefined" }}
var fase1 = { $match : query1 }
```

```
var query2 = { _id: "$cast", count: { $sum: 1 } }
var fase2 = { $group: query2 }
```

```
var query3 = { count: -1 }
var fase3 = { $sort: query3 }
```

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

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

```
var query1 = {cast: { $ne: "Undefined" }}
var fase1 = { $match : query1 }
var query2 = { _id: "$cast", count: { $sum: 1 } }
var fase2 = { $group: query2 }
var query3 = { count: -1 }
var fase3 = { $sort: query3 }
var fase4 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.actors.aggregate( etapas )
```

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

## Ejercicio 18

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

```
var query1 = { _id: { title: "$title", year: "$year" }, totalActores: { $sum: 1 } }
var fase1 = { $group : query1 }
```

```
var query2 = { totalActores: -1 }
var fase2 = { $sort: query2 }
```

```
var fase3 = { $limit: 5 }
```

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

```
var query1 = { _id: { title: "$title", year: "$year" }, totalActores: { $sum: 1 } }
var fase1 = { $group: query1 }
var query2 = { totalActores: -1 }
var fase2 = { $sort: query2 }
var fase3 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3 ]
db.actors.aggregate( etapas )
```

actors   0.192 s   5 Docs			
	_id		totalActores
	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



## Ejercicio 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 query1 = {cast: { $ne: "Undefined" }}
```

```
var fase1 = { $match : query1 }
```

```
var fase2 = { $unwind: "$cast" }
```

```
var query3 = { "_id": 0 }
```

```
var fase3 = { $project: query3 }
```

```
var query4 = { _id: "$cast", primerAnyo: { $min: "$year" }, ultimoAnyo: { $max: "$year" } }
```

```
var fase4 = { $group: query4 }
```

```
var query5 = { anyosDeCarrera: { $subtract: [ "$ultimoAnyo", "$primerAnyo" ] } }
```

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

```
var query6 = { anyosDeCarrera: -1 }
```

```
var fase6 = { $sort: query6 }
```

```
var fase7 = { $limit: 5 }
```

```
var fase8 = { $out: "actors" }
```

```
var etapas = [ fase1, fase2, fase3, fase4, fase5, fase6, fase7, fase8 ]
```

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

```
var query1 = {cast: { $ne: "Undefined" }}
var fase1 = { $match : query1 }

var fase2 = { $unwind: "$cast" }
var query3 = { "_id": 0 }
var fase3 = { $project: query3 }

var query4 = { _id: "$cast",
  primerAnyo: { $min: "$year" },
  ultimoAnyo: { $max: "$year" }
}
var fase4 = { $group: query4 }

var query5 = { anyosDeCarrera: { $subtract: [ "$ultimoAnyo", "$primerAnyo" ] } }
var fase5 = { $addFields: query5 }

var query6 = { anyosDeCarrera: -1 }
var fase6 = { $sort: query6 }

var fase7 = { $limit: 5 }

var fase8 = { $out: "actors" }

var etapas = [ fase1, fase2, fase3, fase4, fase5, fase6, fase7, fase8 ]
db.movies.aggregate(etapas)

db.actors.find()
```

_id	anyosDeCarrera	primerAnyo	ultimoAnyo
1 Harrison Ford	98	1919 (1.9K)	2017 (2.0K)
2 Gloria Stuart	80	1932 (1.9K)	2012 (2.0K)
3 Lillian Gish	75	1912 (1.9K)	1987 (2.0K)
4 Kenny Baker	75	1937 (1.9K)	2012 (2.0K)
5 Mickey Rooney	74	1932 (1.9K)	2006 (2.0K)

## Ejercicio 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.

```
var fase1 = { $unwind: "$genres" }
var query2 = { "_id": 0 }
var fase2 = { $project: query2 }
var fase3 = { $out: "genres" }
var etapas = [ fase1, fase2, fase3 ]
db.actors.aggregate(etapas)
```

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

```
var fase1 = { $unwind: "$genres" }
var query2 = { "_id": 0 }
var fase2 = { $project: query2 }
var fase3 = { $out: "genres" }
var etapas = [ fase1, fase2, fase3 ]
db.actors.aggregate(etapas)

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

genres 0.954 s 104.950 Docs					
	id	cast	genres	title	year
1	6642a01970f535c8ae0	Undefined	Undefined	Caught	1900
2	6642a01970f535c8ae0	Undefined	Undefined	After Dark in Central Park	1900
3	6642a01970f535c8ae0	Undefined	Undefined	Buffalo Bill's Wild West Parade	1900
4	6642a01970f535c8ae0	Undefined	Undefined	The Enchanted Drawing	1900
5	6642a01970f535c8ae0	Undefined	Undefined	Clowns Spinning Hats	1900
6	6642a01970f535c8ae0	Undefined	Short	Capture of Boer Battery by British	1900
7	6642a01970f535c8ae0	Undefined	Documentary	Capture of Boer Battery by British	1900
8	6642a01970f535c8ae0	Paul Boyton	Undefined	Feeding Sea Lions	1900
9	6642a01970f535c8ae0	Undefined	Comedy	How to Make a Fat Wife Out of Two Lea	1900
10	6642a01970f535c8ae0	Undefined	Undefined	Searching Ruins on Broadway, Galvest	1900
11	6642a01970f535c8ae0	Undefined	Undefined	The Tribulations of an Amateur Photogr	1900
12	6642a01970f535c8ae0	Undefined	Comedy	Trouble in Hogan's Alley	1900
13	6642a01970f535c8ae0	Undefined	Undefined	Boarding School Girls' Pajama Parade	1900

0.113 s
1 104950

## Ejercicio 21

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

```
var query1 = { _id: { year: "$year", genre: "$genres" }, count: { $sum: 1 } }
var fase1 = { $group: query1 }
var query2 = { count: -1 }
var fase2 = { $sort: query2 }
var fase3 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3 ]
db.genres.aggregate( etapas )
```

```
var query1 = { _id: { year: "$year", genre: "$genres" }, count: { $sum: 1 } }
var fase1 = { $group: query1 }
var query2 = { count: -1 }
var fase2 = { $sort: query2 }
var fase3 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3 ]
db.genres.aggregate( etapas )
```

genres		0.120 s	5 Docs
	count	_id	
		genre	year
1	790	Comedy	2012
2	694	Comedy	2013
3	677	Drama	2017
4	657	Drama	2012
5	618	Drama	1919

## Ejercicio 22

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

```
var query1 = {cast: {$ne: "Undefined" }}
var fase1 = {$match : query1}
var query2 = {_id: "$cast", genres: { $addToSet: "$genres" }}
var fase2 = { $group: query2 }
var query3 = { cast: "$_id", _id: 0, genres: 1, numGenres: { $size: "$genres" } }
var fase3 = {$project: query3}
var query4 = { numGenres: -1}
var fase4 = {$sort: query4}
var fase5 = {$limit: 5}
var etapas = [ fase1, fase2, fase3, fase4, fase5]
db.genres.aggregate( etapas )
```

```
var query1 = {cast: {$ne: "Undefined" }}
var fase1 = {$match : query1}
var query2 = {_id: "$cast", genres: { $addToSet: "$genres" }}
var fase2 = { $group: query2 }
var query3 = { cast: "$_id", _id: 0, genres: 1, numGenres: { $size: "$genres" } }
var fase3 = {$project: query3}
var query4 = { numGenres: -1}
var fase4 = {$sort: query4}
var fase5 = {$limit: 5}
var etapas = [ fase1, fase2, fase3, fase4, fase5]
db.genres.aggregate( etapas )
```

cast	genres	numGenres
1 Dennis Quaid	Array[20]	20
2 Michael Caine	Array[19]	19
3 James Mason	Array[19]	19
4 Danny Glover	Array[18]	18
5 Gene Hackman	Array[18]	18

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

## Ejercicio 23

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

```
var query1 = { _id: { title: "$title", year: "$year" }, genres: { $addToSet: "$genres" } }
var fase1 = { $group: query1 }
var query2 = { numGenres: { $size: "$genres" }, genres: 1 }
var fase2 = { $project: query2 }
var query3 = { numGenres: -1 }
var fase3 = { $sort: query3 }
var fase4 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.genres.aggregate( etapas )
```

```
var query1 = { _id: { title: "$title", year: "$year" }, genres: { $addToSet: "$genres" } }
var fase1 = { $group: query1 }
var query2 = { numGenres: { $size: "$genres" }, genres: 1 }
var fase2 = { $project: query2 }
var query3 = { numGenres: -1 }
var fase3 = { $sort: query3 }
var fase4 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.genres.aggregate( etapas )
```

	_id		genres	numGenres
	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	My Little Pony: The Movie	2017	Array[6]	6
5	Thor: Ragnarok	2017	Array[6]	6

## Ejercicio 24

Query libre sobre el pipeline de agregación. Obtener el promedio de películas estrenadas por año, segmentadas por género y mostrar los 5 géneros más recurrentes.

```
var query1 = { _id: { year: "$year", genre: "$genres" }, numMovies: { $sum: 1 } }
var fase1 = { $group : query1 }
```

```
var query2 = { _id: "$_id.genre", averageMoviesPerYear: { $avg: "$numMovies" } }
var fase2 = { $group: query2 }
```

```
var query3 = { averageMoviesPerYear: -1 }
var fase3 = { $sort: query3 }
```

```
var query4 = { _id: 0, genre: "$_id", averageMoviesPerYear: 1 }
var fase4 = { $project: query4 }
```

```
var fase5 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3, fase4, fase5 ]
db.movies.aggregate( etapas )
```

```
var query1 = { _id: { year: "$year", genre: "$genres" }, numMovies: { $sum: 1 } }
var fase1 = { $group : query1 }

var query2 = { _id: "$_id.genre", averageMoviesPerYear: { $avg: "$numMovies" } }
var fase2 = { $group: query2 }

var query3 = { averageMoviesPerYear: -1 }
var fase3 = { $sort: query3 }

var query4 = { _id: 0, genre: "$_id", averageMoviesPerYear: 1 }
var fase4 = { $project: query4 }

var fase5 = { $limit: 5 }
var etapas = [ fase1, fase2, fase3, fase4, fase5 ]
db.movies.aggregate( etapas )
```

Key	Value
▶ (1)	{ averageMoviesPerYear : 59.189189189189186, genre : [ "Drama" ] }
▶ (2)	{ averageMoviesPerYear : 44.91228070175438, genre : [ "Comedy" ] }
▶ (3)	{ averageMoviesPerYear : 29.484536082474225, genre : [ "Western" ] }
▶ (4)	{ averageMoviesPerYear : 15, genre : [ "Short", "Comedy" ] }
▶ (5)	{ averageMoviesPerYear : 14.142857142857142, genre : [ "Noir" ] }

## Ejercicio 25

Query libre sobre el pipeline de agregación. Contar el número de películas por década.

```
var query1 = {decade: { $subtract: ["$year", { $mod: ["$year", 10] } ]}}
var fase1 = { $project: query1 }
```

```
var query2 = { _id: "$decade", count: { $sum: 1 } }
var fase2 = { $group: query2 }
```

```
var fase3 = { $sort: { _id: 1 } }
```

```
var query4 = { _id: 0, decade: "$_id", count: 1 }
var fase4 = { $project: query4 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.movies.aggregate( etapas )
```

```
var query1 = {decade: { $subtract: ["$year", { $mod: ["$year", 10] } ]}}
var fase1 = { $project: query1 }

var query2 = { _id: "$decade", count: { $sum: 1 } }
var fase2 = { $group: query2 }

var fase3 = { $sort: { _id: 1 } }

var query4 = { _id: 0, decade: "$_id", count: 1 }
var fase4 = { $project: query4 }
var etapas = [ fase1, fase2, fase3, fase4 ]
db.movies.aggregate( etapas )
```

▶ (1)	{ count : 350, decade : 1900 }
▶ (2)	{ count : 1443, decade : 1910 }
▶ (3)	{ count : 3584, decade : 1920 }
▶ (4)	{ count : 4104, decade : 1930 }
▶ (5)	{ count : 4023, decade : 1940 }
▶ (6)	{ count : 3252, decade : 1950 }
▶ (7)	{ count : 1414, decade : 1960 }
▶ (8)	{ count : 1387, decade : 1970 }

## Ejercicio 26

Query libre sobre el pipeline de agregación. Encontrar el actor con más apariciones por cada década.

```

var query0 = {cast: { $ne: "Undefined" }}
var fase0 = { $match : query0 }
var query1 = {decade: { $subtract: ["$year", { $mod: ["$year", 10] } ]}}
var fase1 = { $addFields : query1 }

var query2 = { _id: { actor: "$cast", decade: "$decade" }, count: { $sum: 1 } }
var fase2 = { $group: query2 }
var fase3 = { $sort: { count: -1 } }

var query4 = { _id: "$_id.decade", actor: { $first: "$_id.actor" }, count: { $first: "$count" } }
var fase4 = { $group: query4 }
var fase5 = { $sort: { _id: 1 } }

var query6 = { _id: 0, decade: "$_id", actor: 1, count: 1 }
var fase6 = { $project: query6 }
var etapas = [fase0, fase1, fase2, fase3, fase4, fase5, fase6]
db.actors.aggregate( etapas )

```

```

var query0 = {cast: { $ne: "Undefined" }}
var fase0 = { $match : query0 }

var query1 = {decade: { $subtract: ["$year", { $mod: ["$year", 10] } ]}}
var fase1 = { $addFields : query1 }

var query2 = { _id: { actor: "$cast", decade: "$decade" }, count: { $sum: 1 } }
var fase2 = { $group: query2 }

var fase3 = { $sort: { count: -1 } }

var query4 = { _id: "$_id.decade", actor: { $first: "$_id.actor" }, count: { $first: "$count" } }
var fase4 = { $group: query4 }

var fase5 = { $sort: { _id: 1 } }

var query6 = { _id: 0, decade: "$_id", actor: 1, count: 1 }
var fase6 = { $project: query6 }

var etapas = [fase0, fase1, fase2, fase3, fase4, fase5, fase6]
db.actors.aggregate( etapas )

```

	actor ↕	count ↕	decade ↕
1	Florence Lawrence	7	1900 (1.9K)
2	Harold Lloyd	160	1910 (1.9K)
3	Hoot Gibson	99	1920 (1.9K)
4	John Wayne	54	1930 (1.9K)
5	Charles Starrett	60	1940 (1.9K)
6	Tom and Jerry	55	1950 (1.9K)
7	Elvis Presley	26	1960 (2.0K)
8	Robert Duvall	20	1970 (2.0K)
9	Burt Reynolds	20	1980 (2.0K)
10	Bruce Willis	25	1990 (2.0K)
11	Samuel L. Jackson	22	2000 (2.0K)
12	Anna Kendrick	21	2010 (2.0K)