Máster Big Data, Data Science & Inteligencia Artificial

Universidad Complutense de Madrid

Bases de Datos NoSQL – MongoDB

Práctica de MongoDB: Operaciones CRUD y Aggregation en colección ‘movies’

María Carolina González Bernal

Febrero 2026

Importación de dataset

movies

	_id ObjectId	title String	year Int32	cast Array	genres Array
1	ObjectId('6983c531291849a…	"Caught"	1900	[] 0 elements	[] 0 elements
2	ObjectId('6983c531291849a…	"After Dark in Central Pa…	1900	[] 0 elements	[] 0 elements
3	ObjectId('6983c531291849a…	"Buffalo Bill's Wild West…	1900	[] 0 elements	[] 0 elements
4	ObjectId('6983c531291849a…	"The Enchanted Drawing"	1900	[] 0 elements	[] 0 elements
5	ObjectId('6983c531291849a…	"Clowns Spinning Hats"	1900	[] 0 elements	[] 0 elements

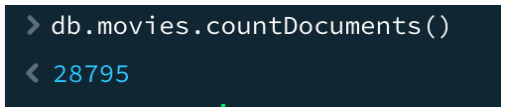
1.Analizar la colección con find.

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

```
> use datascience
< switched to db datascience
> db["movies"].find()
< {
    _id: ObjectId('6983c531291849aae75a0eb7'),
    title: 'Caught',
    year: 1900,
    cast: [],
    genres: []
  }
  {
    _id: ObjectId('6983c531291849aae75a0eb8'),
    title: 'After Dark in Central Park',
    year: 1900,
    cast: [],
    genres: []
  }
  {
    _id: ObjectId('6983c531291849aae75a0eb9'),
    title: "Buffalo Bill's Wild West Parad",
    year: 1900,
    cast: [],
    genres: []
  }
  {
    _id: ObjectId('6983c531291849aae75a0eba'),
    title: 'The Enchanted Drawing',
    year: 1900,
    cast: [],
    genres: []
  }
```

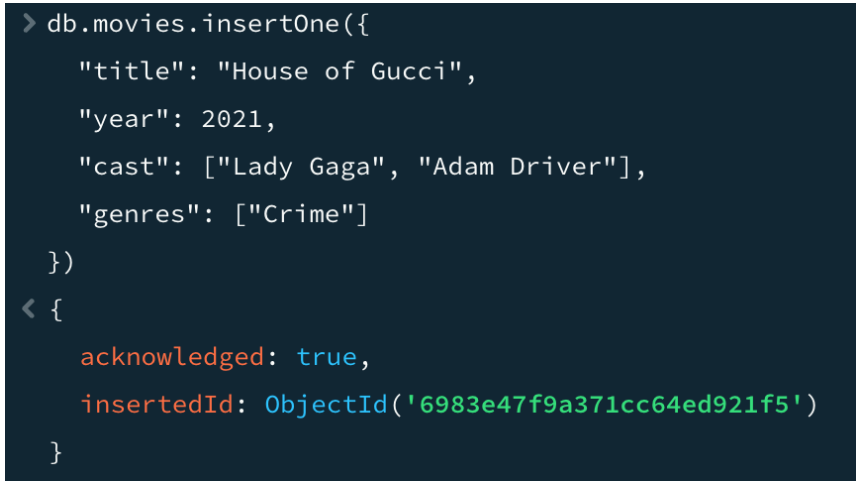
2. Conteo de documentos (películas) cargadas.

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



3. Insertar película.

```
db.movies.insertOne({
  "title": "House of Gucci",
  "year": 2021,
  "cast": ["Lady Gaga", "Adam Driver"],
  "genres": ["Crime"]
})
```



4. Borrar la película insertada en el ejercicio anterior.

Realicé primero un find, luego la eliminé.

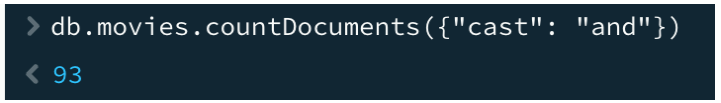
```
db.movies.find({ "title": "House of Gucci" })
```

```
db.movies.deleteOne({ "title": "House of Gucci" })
```

```
> db.movies.find({ "title": "House of Gucci" })
< {
    _id: ObjectId('6982f8860e2269bf070f61f3'),
    title: 'House of Gucci',
    year: 2021,
    cast: [
      'Lady Gaga',
      'Adam Driver'
    ],
    genres: [
      'Crime'
    ]
  }
> db.movies.deleteOne({"title": "House of Gucci"
  })
< {
    acknowledged: true,
    deletedCount: 1
  }
```

5. Contar cuántas películas tienen actores (cast) que se llaman “and” (valor erróneo).

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



6. Actualizar los documentos cuyo actor (cast) tenga el valor erróneo “and” eliminando solo ese valor del array cast.

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

7. Contar cuántos documentos tienen el array cast vacío.

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

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

8. Actualizar todos los documentos con cast vacío, añadiendo ["Undefined"] al array.

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

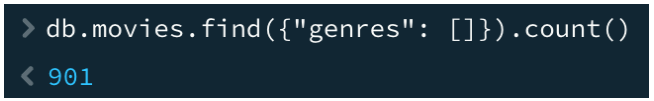
```
> db.movies.updateMany(
    { "cast": [] },
    { $set: { "cast": ["Undefined"] } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 986,
    modifiedCount: 986,
    upsertedCount: 0
  }
```

```
db.movies.find({"cast": "Undefined"})
```

```
> db.movies.find({"cast": "Undefined"})
< {
    _id: ObjectId('6983fa20291849aae75a7f3c'),
    title: 'Caught',
    year: 1900,
    cast: [
      'Undefined'
    ],
    genres: []
  }
```

9. Contar cuántos documentos tienen el array genres vacío.

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



10. Actualizar todos los documentos con genres vacío, añadiendo ["Undefined"] al array.

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

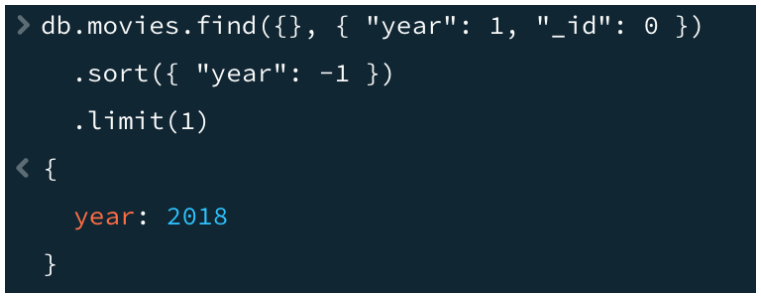
```
> db.movies.updateMany(
    { "genres": [] },
    { $set: { "genres": ["Undefined"] } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 901,
    modifiedCount: 901,
    upsertedCount: 0
  }
```

```
db.movies.find({"genre": "Undefined"})
```

```
> db.movies.find({ "genres": "Undefined" })
< {
    _id: ObjectId('6982f21ec1f1f41d3e534edd'),
    title: 'Caught',
    year: 1900,
    cast: [
      'Unknown Actor'
    ],
    genres: [
      'Undefined'
    ]
  }
```

11. Mostrar el año más reciente de todas las películas.

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

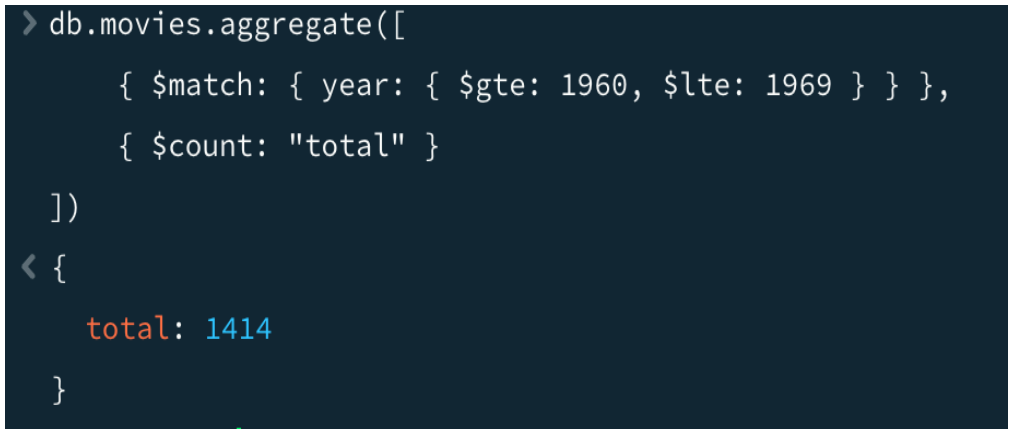


12. Contar cuántas películas han salido en los últimos 20 años desde el año más reciente de la colección (usando Aggregation).

```
db.movies.aggregate([
  {
    $match: {
      year: { $gte: 1999, $lte: 2018 }
    }
  },
  {
    $count: "total"
  }
])
```

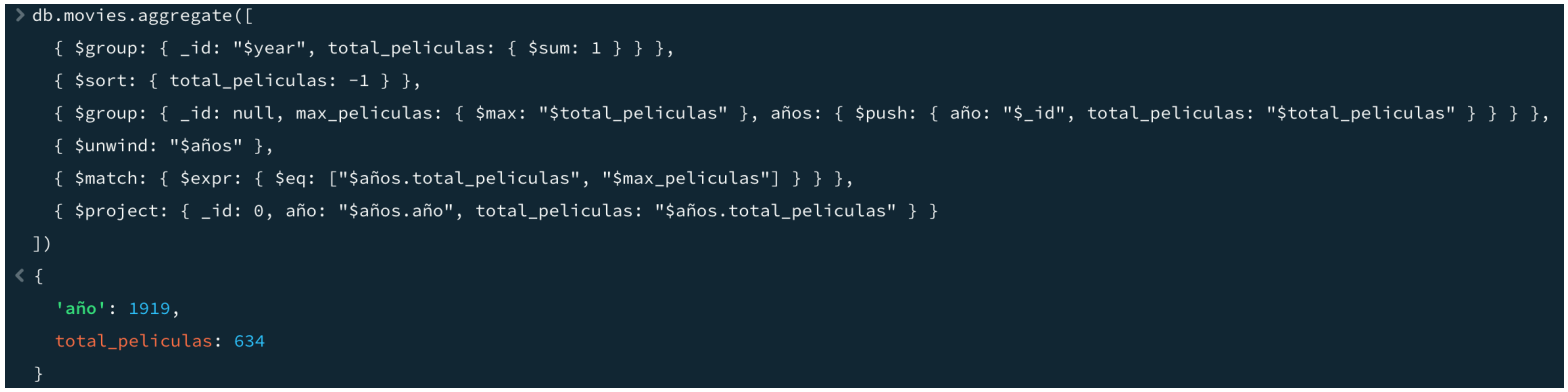
13. Contar cuántas películas salieron en la década de los 60 (1960–1969 incluidos, usando Aggregation).

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



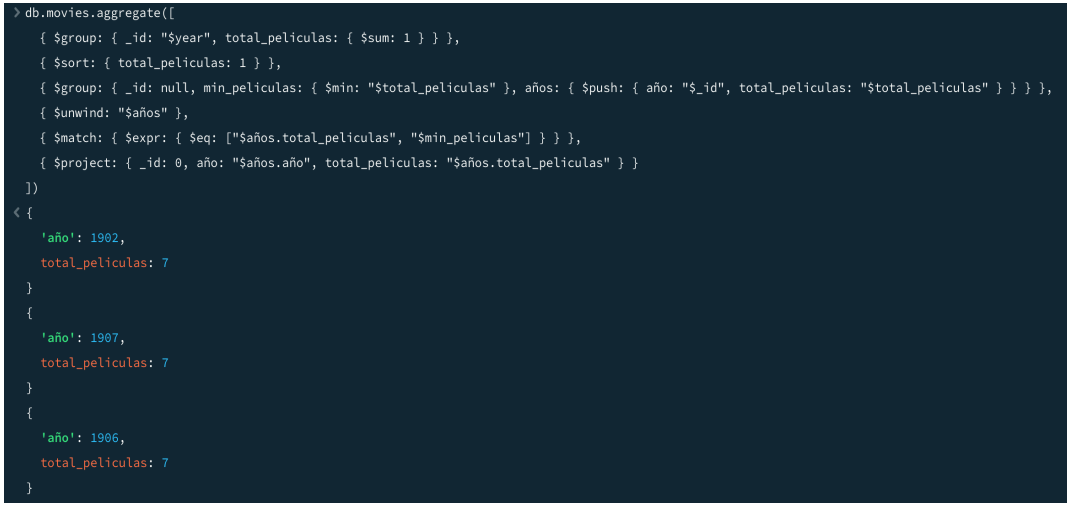
14. Mostrar el año o años con más películas y el número de películas.

```
db.movies.aggregate([
  { $group: { _id: "$year", total_peliculas: { $sum: 1 } } },
  { $sort: { total_peliculas: -1 } },
  { $group: { _id: null, max_peliculas: { $max: "$total_peliculas" }, años: { $push: { año: "$_id", total_peliculas: "$total_peliculas" } } } },
  { $unwind: "$años" },
  { $match: { $expr: { $eq: ["$años.total_peliculas", "$max_peliculas"] } } },
  { $project: { _id: 0, año: "$años.año", total_peliculas: "$años.total_peliculas" } }
])
```



15. Mostrar el año o años con menos películas y el número de películas.

```
db.movies.aggregate([
  { $group: { _id: "$year", total_peliculas: { $sum: 1 } } },
  { $sort: { total_peliculas: 1 } },
  { $group: { _id: null, min_peliculas: { $min: "$total_peliculas" }, años: { $push: { año: "$_id", total_peliculas: "$total_peliculas" } } } },
  { $unwind: "$años" },
  { $match: { $expr: { $eq: ["$años.total_peliculas", "$min_peliculas"] } } },
  { $project: { _id: 0, año: "$años.año", total_peliculas: "$años.total_peliculas" } }
])
```



16 . Guardar en una nueva colección llamada “actors” haciendo $unwind por cast, sin reagrupar, conservando todas las claves. Luego contar documentos en “actors”.

```
db.movies.aggregate([
  { $unwind: "$cast" },
  { $project: { _id: 0, title: 1, year: 1, cast: 1, genres: 1 } },
  { $out: "actors" }
])
db.actors.countDocuments()
```

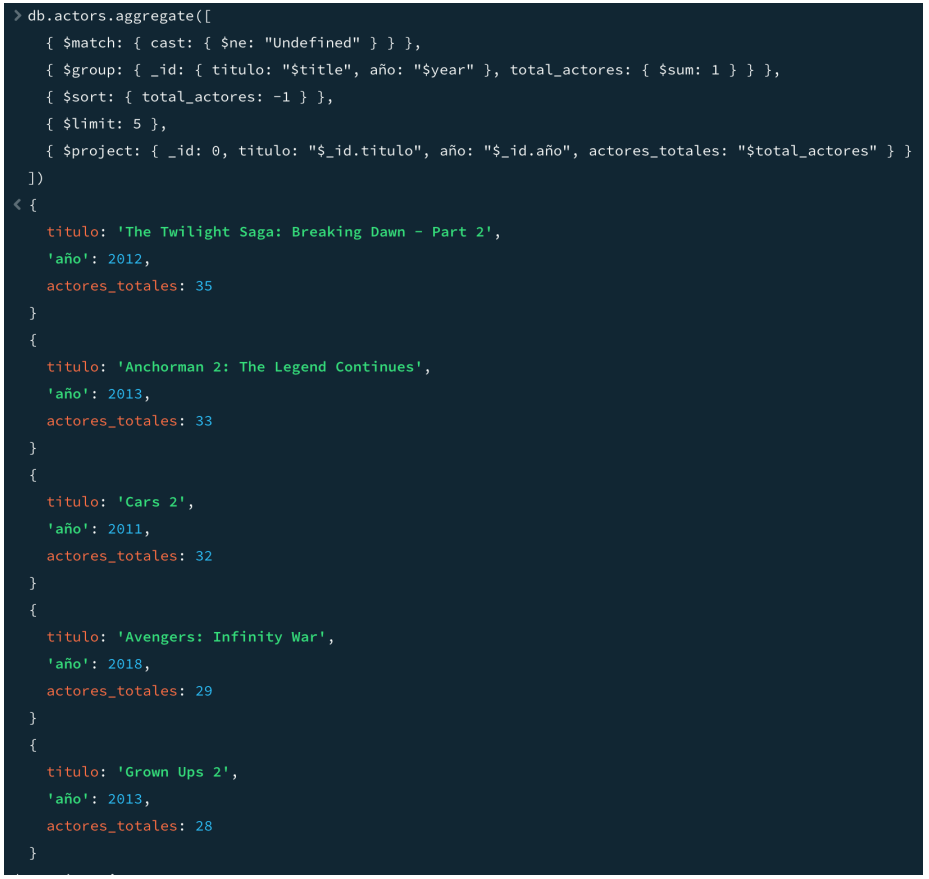
```
> db.movies.aggregate([
    { $unwind: "$cast" },
    { $project: { _id: 0, title: 1, year: 1, cast: 1, genres: 1 } },
    { $out: "actors" }
  ])
<
> db.actors.countDocuments()
< 83317
```

17. Sobre “actors”, mostrar los 5 actores que han participado en más películas (filtrando “Undefined”).

```
db.actors.aggregate([
  { $match: { cast: { $ne: "Undefined" } } },
  { $group: { _id: "$cast", total_peliculas: { $sum: 1 } } },
  { $sort: { total_peliculas: -1 } },
  { $limit: 5 }
])
```

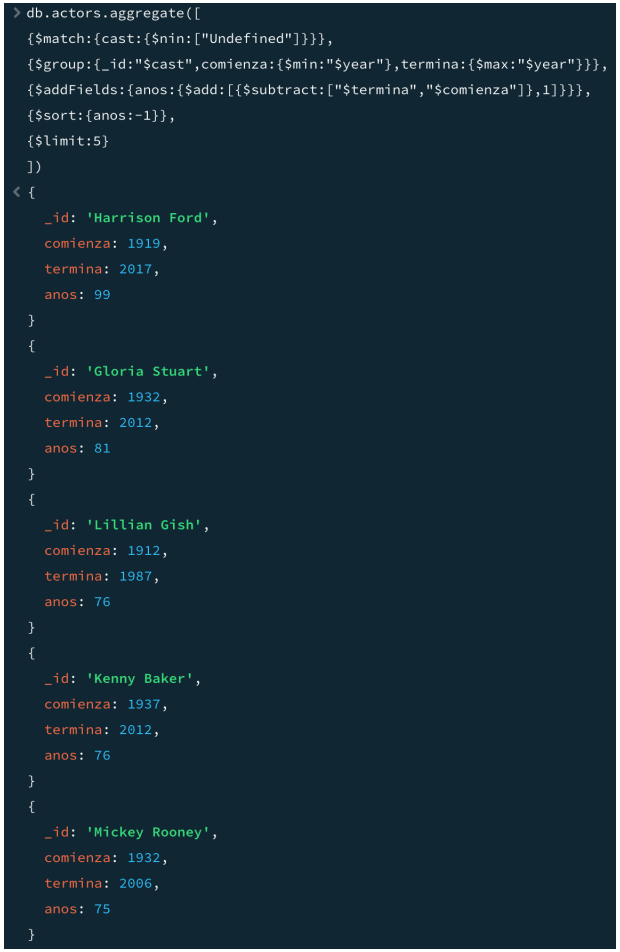
18. Sobre “actors”, agrupar por película y año y mostrar las 5 con más actores.

```
db.actors.aggregate([
  { $match: { cast: { $ne: "Undefined" } } },
  { $group: { _id: { titulo: "$title", año: "$year" }, total_actores: { $sum: 1 } } },
  { $sort: { total_actores: -1 } },
  { $limit: 5 },
  { $project: { _id: 0, titulo: "$_id.titulo", año: "$_id.año", actores_totales: "$total_actores" } }
])
```



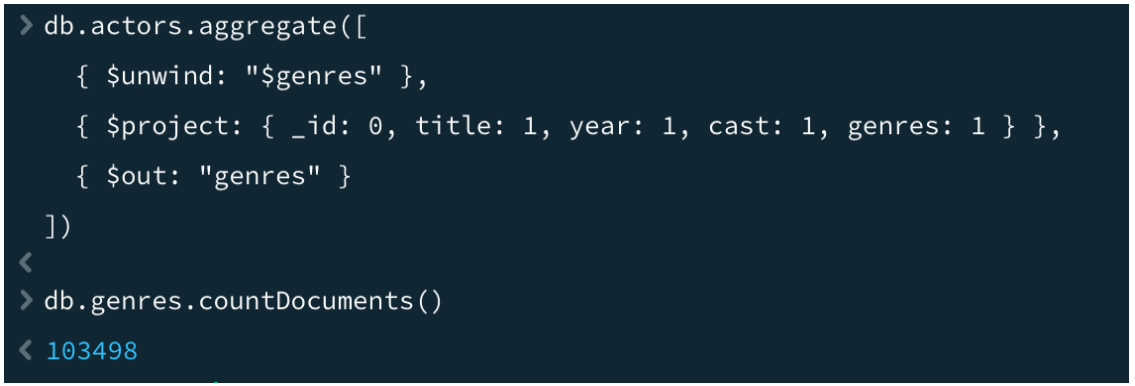
19. Sobre “actors”, mostrar los 5 actores con la carrera más larga (inicio, fin, años trabajados), filtrando “Undefined”.

```
db.actors.aggregate([
{$match:{cast:{$nin:["Undefined"]}}},
{$group:{_id:"$cast",comienza:{$min:"$year"},termina:{$max:"$year"}}},
{$addFields:{anos:{$add:[{$subtract:["$termina","$comienza"]},1]}}},
{$sort:{anos:-1}},
{$limit:5}
])
```



20. Guardar en una nueva colección “genres” haciendo $unwind por genres, sin reagrupar, conservando todas las claves. Contar documentos en “genres”.

```
db.actors.aggregate([
  { $unwind: "$genres" },
  { $project: { _id: 0, title: 1, year: 1, cast: 1, genres: 1 } },
  { $out: "genres" }
])
db.genres.countDocuments()
```



21. Sobre “genres”, mostrar los 5 documentos agrupados por año y género con más películas diferentes (filtrando “Undefined”).

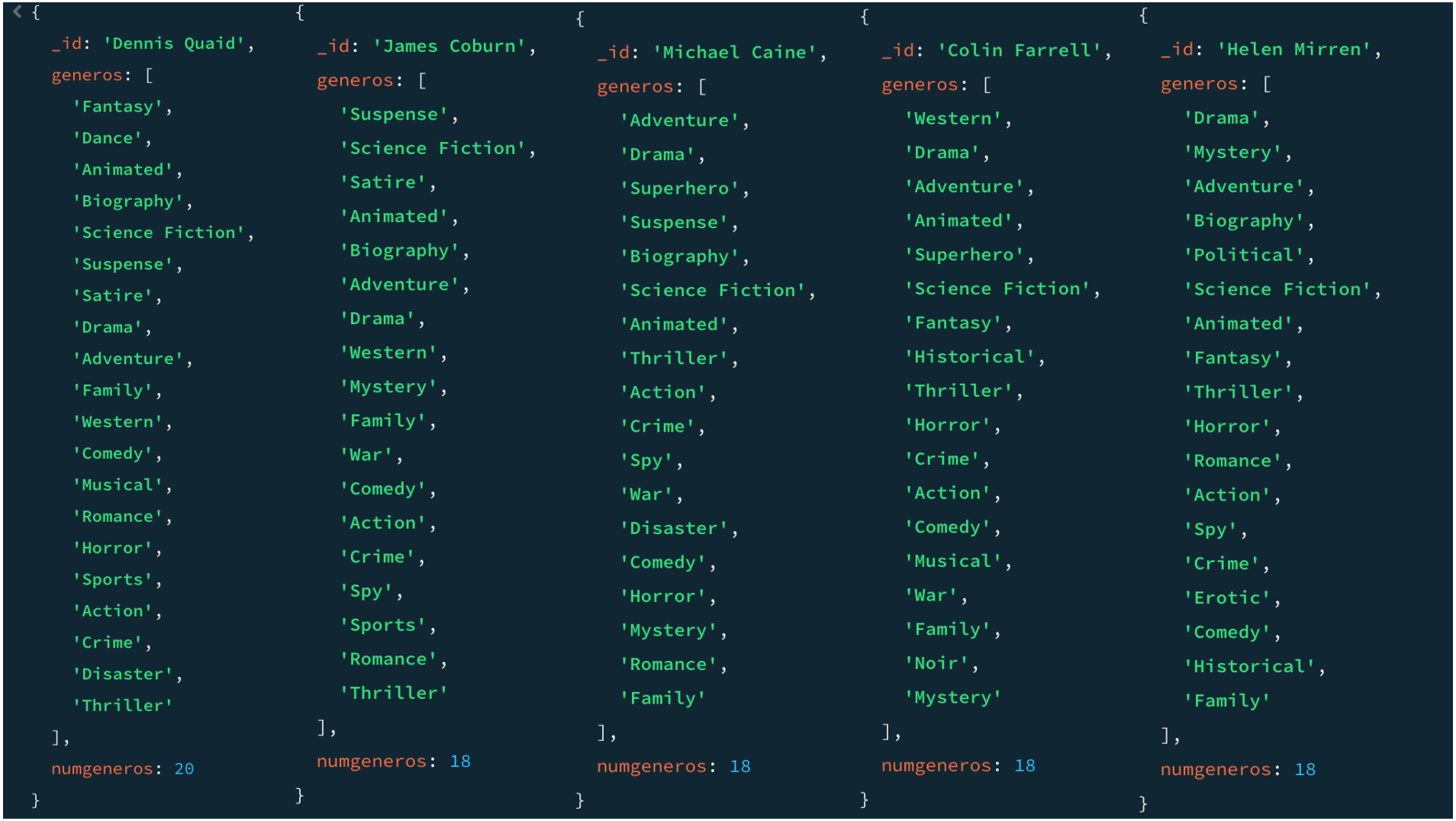
```
db.genres.aggregate([
  { $match: { genres: { $ne: "Undefined" } } },
  {
    $group: {
      _id: { year: "$year", genre: "$genres" },
      pelis: { $addToSet: "$title" }
    }
  },
  {
    $project: {
      _id: 1,
      total_peliculas: { $size: "$pelis" }
    }
  },
  { $sort: { total_peliculas: -1 } },
  { $limit: 5 }
])
```

```
> db.genres.aggregate([
    { $match: { genres: { $ne: "Undefined" } } },
    {
      $group: {
        _id: { year: "$year", genre: "$genres" },
        pelis: { $addToSet: "$title" }
      }
    },
    {
      $project: {
        _id: 1,
        total_peliculas: { $size: "$pelis" }
      }
    },
    { $sort: { total_peliculas: -1 } },
    { $limit: 5 }
  ])
```

```
  _id: {
    year: 1919,
    genre: 'Drama'
  },
  total_peliculas: 291
}
{
  _id: {
    year: 1925,
    genre: 'Drama'
  },
  total_peliculas: 247
}
{
  _id: {
    year: 1924,
    genre: 'Drama'
  },
  total_peliculas: 233
}
{
  _id: {
    year: 1919,
    genre: 'Comedy'
  },
  total_peliculas: 226
}
{
  _id: {
    year: 1922,
    genre: 'Drama'
  },
  total_peliculas: 209
}
```

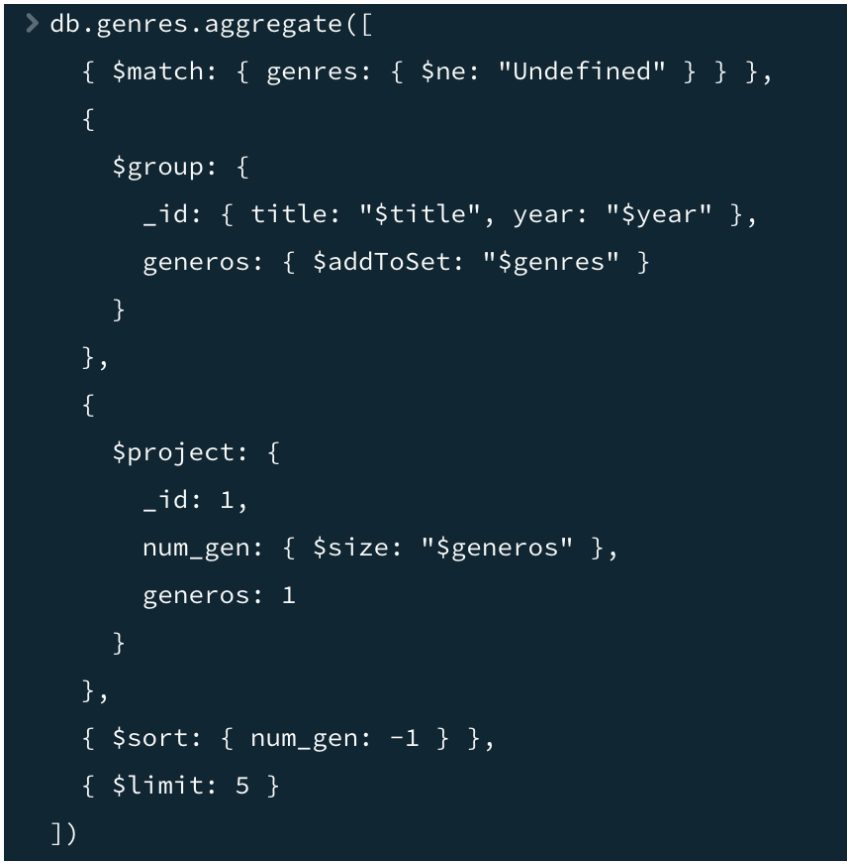
22. Sobre “genres”, mostrar los 5 actores con más géneros diferentes en los que han participado, listando géneros y cantidad (filtrando “Undefined”).

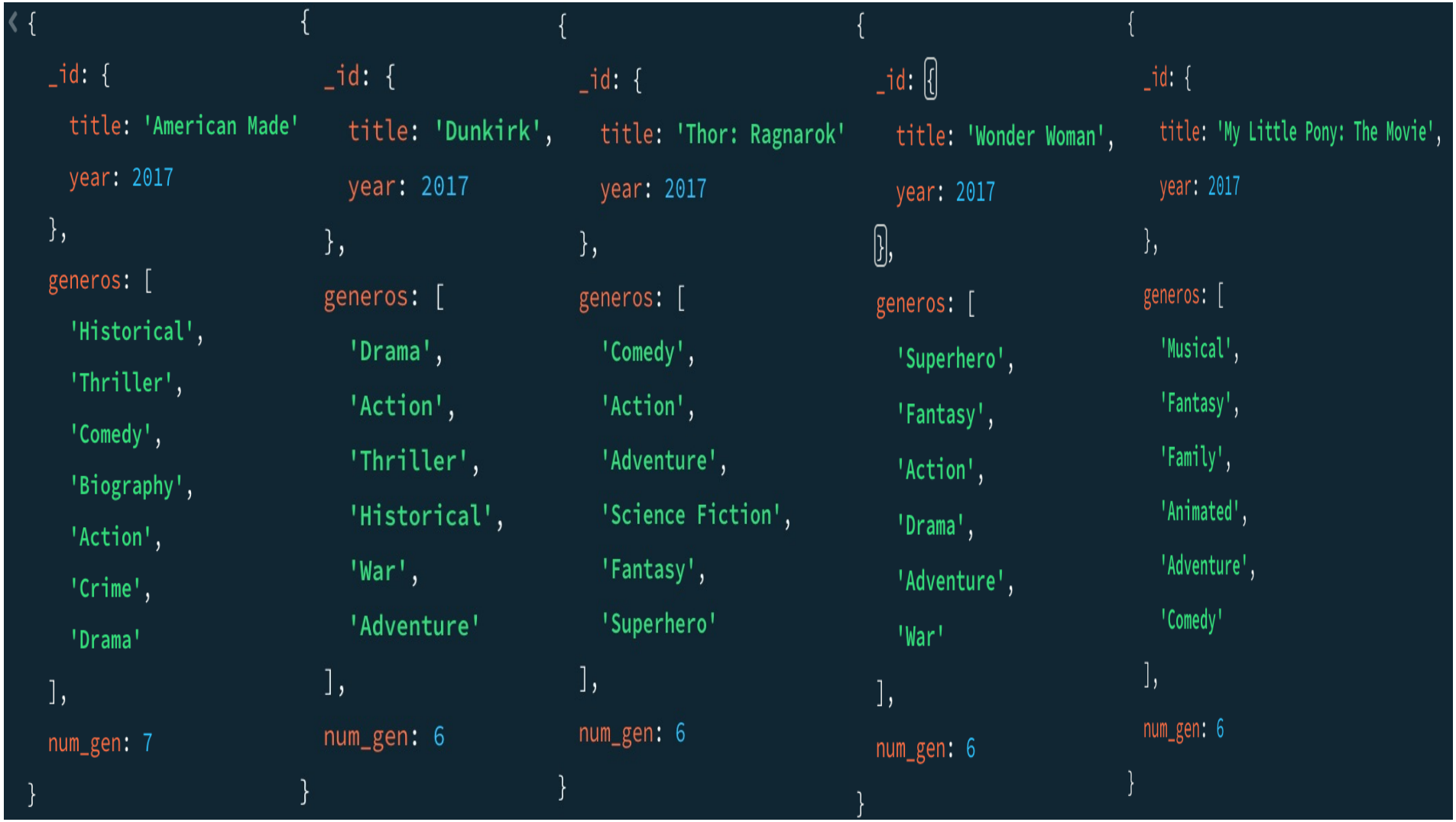
```
db.genres.aggregate([
  { $match: { cast: { $ne: "Undefined" } } },
  {
    $group: {
      _id: "$cast",
      generos: { $addToSet: "$genres" }
    }
  },
  {
    $project: {
      _id: 1,
      numgeneros: { $size: "$generos" },
      generos: 1
    }
  },
  { $sort: { numgeneros: -1 } },
  { $limit: 5 }
])
```


{
_id: 'Dennis Quaid',
generos: [
'Fantasy',
'Dance',
'Animated',
'Biography',
'Science Fiction',
'Suspense',
'Satire',
'Drama',
'Adventure',
'Family',
'Western',
'Comedy',
'Musical',
'Romance',
'Horror',
'Sports',
'Action',
'Crime',
'Disaster',
'Thriller'
],
numgeneros: 20
}
{
_id: 'James Coburn',
generos: [
'Suspense',
'Science Fiction',
'Satire',
'Animated',
'Biography',
'Adventure',
'Drama',
'Western',
'Mystery',
'Family',
'War',
'Comedy',
'Action',
'Crime',
'Spy',
'Sports',
'Romance',
'Thriller'
],
numgeneros: 18
}
{
_id: 'Michael Caine',
generos: [
'Adventure',
'Drama',
'Superhero',
'Suspense',
'Biography',
'Science Fiction',
'Animated',
'Thriller',
'Action',
'Crime',
'Spy',
'War',
'Disaster',
'Comedy',
'Horror',
'Mystery',
'Romance',
'Family'
],
numgeneros: 18
}
{
_id: 'Colin Farrell',
generos: [
'Western',
'Drama',
'Adventure',
'Animated',
'Superhero',
'Science Fiction',
'Fantasy',
'Historical',
'Thriller',
'Horror',
'Crime',
'Action',
'Comedy',
'Musical',
'War',
'Family',
'Noir',
'Mystery'
],
numgeneros: 18
}
{
_id: 'Helen Mirren',
generos: [
'Drama',
'Mystery',
'Adventure',
'Biography',
'Political',
'Science Fiction',
'Animated',
'Fantasy',
'Thriller',
'Horror',
'Romance',
'Action',
'Spy',
'Crime',
'Erotic',
'Comedy',
'Historical',
'Family'
],
numgeneros: 18
}

23. Sobre “genres”, mostrar las 5 películas con más géneros diferentes, listando géneros y cantidad (filtrando “Undefined”).

```
db.genres.aggregate([
  { $match: { genres: { $ne: "Undefined" } } },
  { $group: {
      _id: { title: "$title", year: "$year" },
      generos: { $addToSet: "$genres" }
    }},
  { $project: {
      _id: 1,
      num_gen: { $size: "$generos" },
      generos: 1
    }},
  { $sort: { num_gen: -1 } },
  { $limit: 5 }
])
```




{
_id: {
title: 'American Made'
year: 2017
},
generos: [
'Historical',
'Thriller',
'Comedy',
'Biography',
'Action',
'Crime',
'Drama'
],
num_gen: 7
}
{
_id: {
title: 'Dunkirk',
year: 2017
},
generos: [
'Drama',
'Action',
'Thriller',
'Historical',
'War',
'Adventure'
],
num_gen: 6
}
{
_id: {
title: 'Thor: Ragnarok'
year: 2017
},
generos: [
'Comedy',
'Action',
'Adventure',
'Science Fiction',
'Fantasy',
'Superhero'
],
num_gen: 6
}
{
_id: {
title: 'Wonder Woman',
year: 2017
},
generos: [
'Superhero',
'Fantasy',
'Action',
'Drama',
'Adventure',
'War'
],
num_gen: 6
}
{
_id: {
title: 'My Little Pony: The Movie',
year: 2017
},
generos: [
'Musical',
'Fantasy',
'Family',
'Animated',
'Adventure',
'Comedy'
],
num_gen: 6
}

24. Género más popular por año desde 2010 a 2018 (por Aggregate)

```
// Género más popular por año desde 2010 a 2018
db.movies.aggregate([
// Filtrar películas de 2010-2018
{$match:{year:{$gte:2010,$lte:2018}}},
// Unwind para convertir array de géneros en documentos individuales
{$unwind:"$genres"},
// Contar películas por año y género
{$group:{_id:{year:"$year",genre:"$genres"},count:{$sum:1}}},
// Ordenar por año ascendente, y por conteo descendente
{$sort:{"_id.year":1,"count":-1}},
// Género más popular por año
{$group:{_id:"$_id.year",most_popular_genre:{$first:"$_id.genre"},movie_count:{$first:"$count"}}},
// Mostrar por año ascendente
{$sort:{_id:1}},
// Resultados
{$project:{year:"$_id",most_popular_genre:1,movie_count:1}}
])
```

```
> db.movies.aggregate([
    { $match: { year: { $gte: 2010, $lte: 2018 } } },
    { $unwind: "$genres" },
    { $group: { _id: { year: "$year", genre: "$genres" }, count: { $sum: 1 } } },
    { $sort: { "_id.year": 1, "count": -1 } },
    { $group: {
        _id: "$_id.year",
        most_popular_genre: { $first: "$_id.genre" },
        movie_count: { $first: "$count" }
      }
    },
    { $sort: { _id: 1 } },
    { $project: { year: "$_id", most_popular_genre: 1, movie_count: 1 } }
  ])
```

```
{
  _id: 2010,
  most_popular_genre: 'Comedy',
  movie_count: 78,
  year: 2010
}
{
  _id: 2011,
  most_popular_genre: 'Comedy',
  movie_count: 68,
  year: 2011
}
{
  _id: 2012,
  most_popular_genre: 'Comedy',
  movie_count: 94,
  year: 2012
}
{
  _id: 2013,
  most_popular_genre: 'Drama',
  movie_count: 103,
  year: 2013
}
{
  _id: 2014,
  most_popular_genre: 'Comedy',
  movie_count: 62,
  year: 2014
}
```

```
{
  _id: 2015,
  most_popular_genre: 'Comedy',
  movie_count: 41,
  year: 2015
}
{
  _id: 2016,
  most_popular_genre: 'Comedy',
  movie_count: 36,
  year: 2016
}
{
  _id: 2017,
  most_popular_genre: 'Drama',
  movie_count: 111,
  year: 2017
}
{
  _id: 2018,
  most_popular_genre: 'Drama',
  movie_count: 104,
  year: 2018
}
```

25. Las 5 películas con más cast entre 2015 y 2018

```
db.movies.aggregate([
// Filtrar películas de 2015–2018
{$match:{year:{$gte:2015,$lte:2018}}},
// Añadir campo con tamaño del array cast
{$addFields:{cast_size:{$size:"$cast"}}},
// Ordenar por tamaño del cast descendente
{$sort:{cast_size:-1}},
// Top 5 resultados
{$limit:5},
// Resultados
{$project:{title:1,year:1,cast_size:1,cast:1}}
])
```

```
{
  _id: ObjectId('6983fa20291849aae75aef1a'),
  title: 'Avengers: Infinity War',
  year: 2018,
  cast: [
    'Robert Downey Jr.',
    'Chris Hemsworth',
    'Mark Ruffalo',
    'Chris Evans',
    'Scarlett Johansson',
    'Benedict Cumberbatch',
    'Don Cheadle',
    'Tom Holland',
    'Chadwick Boseman',
    'Paul Bettany',
    'Elizabeth Olsen',
    'Anthony Mackie',
    'Sebastian Stan',
    'Danai Gurira',
    'Letitia Wright',
    'Dave Bautista',
    'Zoe Saldana',
    'Pom Klementieff',
    'Karen Gillan',
    'Benedict Wong',
    'Idris Elba',
    'Peter Dinklage',
    'Tom Hiddleston',
    'Vin Diesel',
    'Bradley Cooper',
    'Gwyneth Paltrow',
    'Benicio del Toro',
    'Josh Brolin',
    'Chris Pratt'
  ],
  cast_size: 29
}
```

```
{
  _id: ObjectId('6983fa20291849aae75aef24'),
  title: 'Deadpool 2',
  year: 2018,
  cast: [
    'Ryan Reynolds',
    'Josh Brolin',
    'Zazie Beetz',
    'Julian Dennison',
    'Morena Baccarin',
    'T. J. Miller',
    'Brianna Hildebrand',
    'Stefan Kapičić',
    'Jack Kesy',
    'Leslie Uggams',
    'Karan Soni',
    'Shioli Kutsuna',
    'Eddie Marsan',
    'Terry Crews',
    'Bill Skarsgård',
    'Lewis Tan',
    'Rob Delaney'
  ],
  cast_size: 17
}
```

```
{
  _id: ObjectId('6983fa20291849aae75aef46'),
  title: 'Hotel Transylvania 3: Summer Vacation',
  year: 2018,
  cast: [
    'Adam Sandler',
    'Andy Samberg',
    'Selena Gomez',
    'David Spade',
    'Steve Buscemi',
    'Molly Shannon',
    'Keegan-Michael Key',
    'Kevin James',
    'Fran Drescher',
    'Asher Blinkoff',
    'Sadie Sandler',
    'Joe Whyte',
    'Kathryn Hahn',
    'Jim Gaffigan',
    'Mel Brooks',
    'Joe Jonas'
  ],
  cast_size: 16
}
```

```
{
  _id: ObjectId('6983fa20291849aae75aef37'),
  title: 'Incredibles 2',
  year: 2018,
  cast: [
    'Holly Hunter',
    'Craig T. Nelson',
    'Sarah Vowell',
    'Huck Milner',
    'Samuel L. Jackson',
    'Bob Odenkirk',
    'Catherine Keener',
    'Brad Bird',
    'Jonathan Banks',
    'Michael Bird',
    'Sophia Bush',
    'Phil LaMarr',
    'Paul Eiding',
    'Bill Wise',
    'Isabella Rossellini',
    'John Ratzenberger'
  ],
  cast_size: 16
}
```

```
{
  _id: ObjectId('6983fa20291849aae75aee3e'),
  title: 'War Machine',
  year: 2017,
  cast: [
    'Michael Hastings',
    'Brad Pitt',
    'Anthony Michael Hall',
    'Topher Grace',
    'Anthony Hayes',
    'John Magaro',
    'Emory Cohen',
    'Daniel Betts',
    'Lakeith Stanfield',
    'RJ Cyler',
    'Aymen Hamdouchi',
    'Alan Ruck',
    'Meg Tilly',
    'Will Poulter',
    'Ben Kingsley',
    'Tilda Swinton'
  ],
  cast_size: 16
}
```

26. Buscar si Lady Gaga forma parte del cast de alguna película

```
db.movies.aggregate([
// Filtrar documentos donde Lady Gaga esté en el array cast
{$match:{cast:"Lady Gaga"}},
// Ordenar por año descendente
{$sort:{year:-1}},
// Resultados
{$project:{
    title:1,
    year:1,
    num_cast:{$size:"$cast"},
    has_lady_gaga:{$cond:[{$in:["Lady Gaga","$cast"]},"Sí","No"]}
}}
])
```

```
> db.movies.aggregate([
    {$match: {cast: "Lady Gaga"}},
    {$sort: {year: -1}},
    {$project: {
        title: 1,
        year: 1,
        num_cast: {$size: "$cast"},
        with_lady_gaga: {$cond: [
          {$in: ["Lady Gaga", "$cast"]},
          "Sí",
          "No"
        ]}
    }}
])
< {
    _id: ObjectId('6983fa20291849aae75aef87'),
    title: 'A Star Is Born',
    year: 2018,
    num_cast: 5,
    with_lady_gaga: 'Sí'
}
```