

>>>>>>>>>>>>>>> SPRINT-9 <<<<<<<<<<<<<<<<<<

===== NIVEL 1 =====

Crea elabora una base de datos con MongoDB utilizando como colecciones los archivos entregados.

localhost:27017

...

My Queries

Performance

Databases

Search

admin

config

local

sprint9

+

-

comments

movies

sessions

theaters

users

localhost:27017 > sprint9

+ Create collection

Refresh

View

Sort by

Collection Name

comments

Storage size: 6.96 MB	Documents: 60 K	Avg. document size: 284.00 B	Indexes: 1	Total index size: 524.29 kB
--------------------------	--------------------	---------------------------------	---------------	--------------------------------

movies

Storage size: 21.64 MB	Documents: 24 K	Avg. document size: 1.60 kB	Indexes: 1	Total index size: 364.54 kB
---------------------------	--------------------	--------------------------------	---------------	--------------------------------

sessions

Storage size: 20.48 kB	Documents: 1	Avg. document size: 540.00 B	Indexes: 1	Total index size: 20.48 kB
---------------------------	-----------------	---------------------------------	---------------	-------------------------------

theaters

Storage size: 147.46 kB	Documents: 1.6 K	Avg. document size: 223.00 B	Indexes: 1	Total index size: 32.77 kB
----------------------------	---------------------	---------------------------------	---------------	-------------------------------

users

Storage size: 36.86 kB	Documents: 185	Avg. document size: 159.00 B	Indexes: 1	Total index size: 20.48 kB
---------------------------	-------------------	---------------------------------	---------------	-------------------------------

Ejercicio 1

Muestra los 2 primeros comentarios que aparecen en la base de datos.

Se trabaja en la plataforma Compass: Seleccionamos la colección comments, luego en el apartado Aggregations, generamos un Add Stage donde se indaga con un \$limit para condicionarlo en los 2 primeros comentarios.

Type a query: { field: 'value' } or [Generate query](#) 

Project { field: 0 }


Sort { field: -1 } or [['field', -1]]

Collation { locale: 'simple' }

Index Hint { field: -1 }

 ADD DATA

 EXPORT DATA

 UPDATE

 DELETE

```
_id: ObjectId('5a9427648b0beebe69579cc')
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd418c')
text: "Rem officiiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
```

```
_id: ObjectId('5a9427648b0beebe69579cf')
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd41b1')
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00
```

```
> _MONGOSH
> use sprint9
< switched to db sprint9
> db.comments.find().limit(2)
< [
  {
    _id: ObjectId('5a9427648b0beebe69579cc'),
    name: 'Andrea Le',
    email: 'andrea_le@fakegmail.com',
    movie_id: ObjectId('573a1390f29313caabcd418c'),
    text: 'Rem officijs eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibus ullam.',
    date: 2012-03-26T23:20:16.000Z
  },
  {
    _id: ObjectId('5a9427648b0beebe69579cf'),
    name: 'Greg Powell',
    email: 'greg_powell@fakegmail.com',
    movie_id: ObjectId('573a1390f29313caabcd41b1'),
    text: 'Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Commodi nisi sit placeat rerum vero cupiditate neque. Dolorum nihil vero animi.',
    date: 1987-02-10T00:29:36.000Z
  }
]
sprint9>
```

¿Cuántos usuarios tenemos registrados?

En este ejercicio se selecciona la colección users, luego efectuamos Add Stage y con \$count hago la indagación.

RESPUESTA: 185 usuarios registrados

The screenshot shows the MongoDB Compass interface. On the left, the database 'sprint9' is selected, and the 'users' collection is highlighted. The main panel shows a preview of documents, with two examples visible: one for 'Ned Stark' and another for 'Robert Baratheon'. Below this, a terminal window shows the command `db.users.countDocuments({})` being executed, resulting in `185`.

¿Cuántos cines existen en el estado de California?

Para este ejercicio se selecciona la colección theaters, luego en el apartado Aggregations efectúo un Add Stage donde selecciono \$match con la query: "location.address.state":"CA"

Posteriormente efectuo otro Add Stage con un \$count.

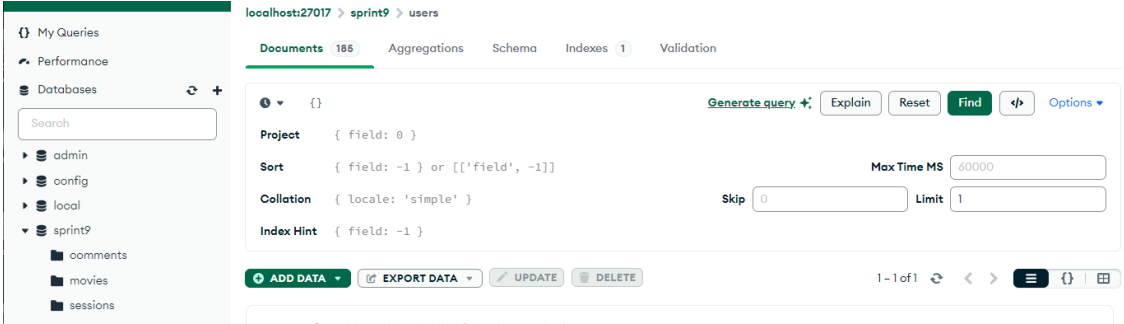
RESPUESTA: 169 cines registrados en California.

The screenshot shows the MongoDB Compass interface. The 'theaters' collection is selected. The 'Aggregations' tab is active, showing a pipeline with two stages: '\$match' and '\$count'. The '\$match' stage has a query of `{ "location.address.state": "CA" }`. The '\$count' stage is selected, and the output shows a count of 169. Below this, a terminal window shows the command `db.theaters.countDocuments({'location.address.state':'CA'})` being executed, resulting in `169`.

¿Cuál fue el primer usuario en registrarse?

En este ejercicio seleccionamos la colección users, luego a Documents, se procede abrir en options y selecciono Limit 1.

RESPUESTA: El primer usuario registrado en la DB es: Ned Stark



```
>_MONGOSH
> db.users.findOne()
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz08HM/jLhgUCNNI39R3AqMUQ74cr131Vu'
}
sprint9>
```

¿Cuántas películas de comedia existen en nuestra base de datos?

En este ejercicio seleccionamos la colección movies, en el apartado de Aggregations, creamos un Add Stage donde efectúo un \$match con el query: "genres":"Comedy". Luego otro Stage con un \$match con : "type":"movie", Como ultimo un Add Stage y selecciono \$count.

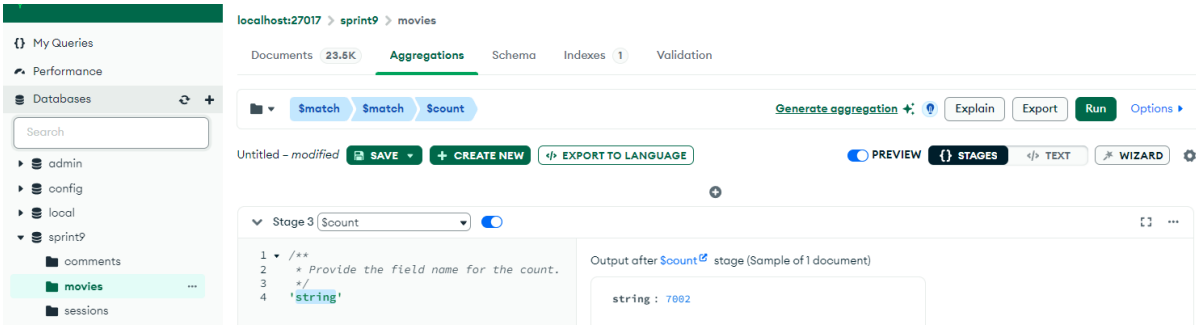
RESPUESTA: La consulta ha alcanzado 7002 películas en el género de comedia.



```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "genres": "Comedy"
6 }
```



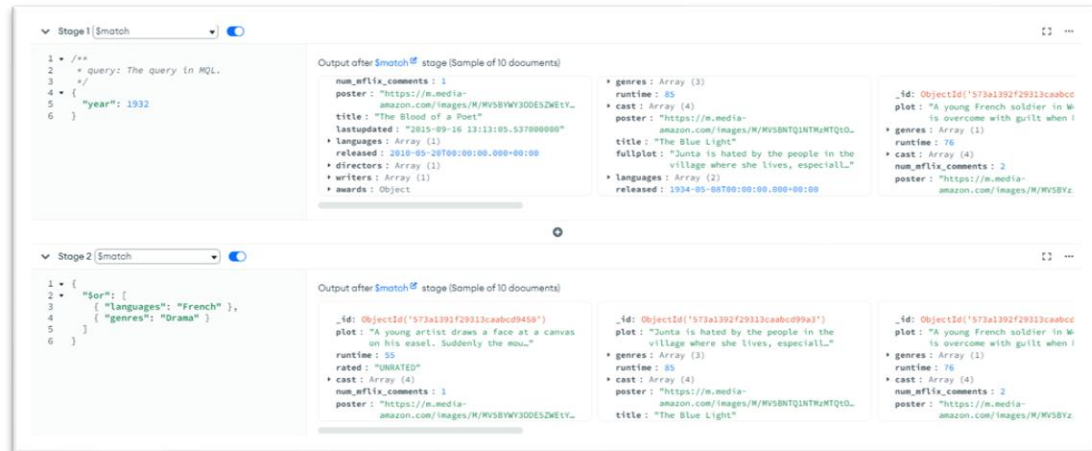
```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   type: "movie"
6 }
```



```
>_MONGOSH
>
> db.movies.countDocuments({
  "$and": [
    { "genres": "Comedy" },
    { "type": "movie" }
  ]
});
< 7002
sprint9>
```

Ejercicio 2

Muéstrame todos los documentos de las películas producidas en 1932, pero que el género sea drama o estén en francés.



En este ejercicio se desarrolla el proceso seleccionando la colección de movies, en el apartado Aggregations, efectuamos un AddStage y seleccionamos \$match declarando year: 1932.

Adicionalmente se efectúa un nuevo AddStage seleccionando un match con la declaración siguiente: { \$or: [{genres: "Drama"}],{languages: "French"}]}

RESPUESTA: 18 películas registradas

```
> _MONGOSH
> db.movies.countDocuments({
  "$and": [
    { "year": 1932 },
    {
      "$or": [
        { "languages": "French" },
        { "genres": "Drama" }
      ]
    }
  ]
});
< 18
sprint9>
```

Ejercicio 3

Muéstrame todos los documentos de películas estadounidenses que tengan entre 5 y 9 premios que fueron producidas entre 2012 y 2014.

En este ejercicio seleccionamos la colección movies, desde apartado Aggregations efectuamos los siguientes 3 Stages:

RESULTADO: 166 documentos que cumplan los requisitos en la DB.



```
Stage 3 $match

1 /**
2  * query: The query in MQL.
3  */
4 {
5   "year": {
6     "$gte": 2012,
7     "$lte": 2014
8   }
9 }
```

```
>_MONGOSH

> db.movies.countDocuments({
  "$and": [
    { "countries": "USA" },

    { "awards.wins" : {$gte:5, $lte:9 }},
    { "year" : {$gte:2012, $lte:2014 }
  ]
});
< 166
sprint9>
```

===== NIVEL 2 =====

Ejercicio 1

Cuenta cuántos comentarios escribe un usuario/a que utiliza "GAMEOFTHRON.ES" como dominio de correo electrónico.

The screenshot shows the MongoDB Aggregations Playground interface. On the left, a sidebar lists databases and collections, with 'sprint9' and 'comments' selected. The main area displays an aggregation pipeline with two stages:

- Stage 1 (\$match):** The query is `{ "email": { "$regex": "@gameofthron.es" } }`. The output shows a sample of 10 documents, including one with `email: "oona_chaplin@gameofthron.es"`.
- Stage 2 (\$count):** The query is `'string'`. The output shows a single document with `string: 22841`, which is highlighted with a red box.

At the top, the document count is 50.3K. Buttons for 'Explain', 'Export', 'Run', and 'Options' are available. The 'PREVIEW' toggle is turned on.

RESPUESTA: 22841 comentarios según a consulta

```
>_MONGOSH
> db.comments.countDocuments({
  "$and": [
    { "email": {$regex:"gameofthron.es"} }
  ]
}):
< 22841
sprint9>
```

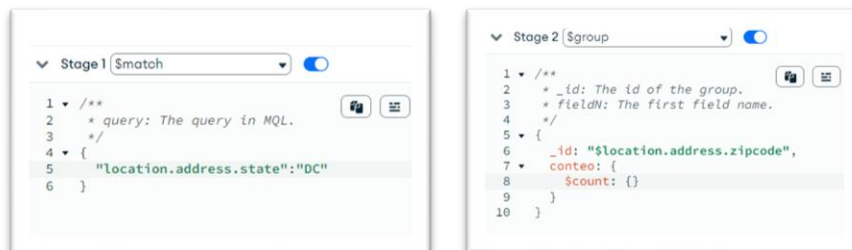
Ejercicio 2

¿Cuántos cines hay en cada código postal situados dentro del estado Washington D. C. (DC)?

Hemos seleccionado la colección theatres en el apartado Aggregations efectuando 2 AddStage.

- Se buscará los cines ubicados en el estado DC
- Luego se los agrupará por Zipcode.

RESPUESTA: 3 registros de cines según el zipcode de DC



Output after \$group stage (Sample of 3 documents)

_id: "20016"
conteo : 1

_id: "20002"
conteo : 1

_id: "20010"
conteo : 1

```
>_MONGOSH
> db.theaters.aggregate([
  {$match: {"location.address.state": "DC"}},
  {$group: {_id: "$location.address.zipcode",
    conteo: {$count: {}}}}
])
< {
  _id: '20016',
  conteo: 1
}
{
  _id: '20002',
  conteo: 1
}
{
  _id: '20010',
  conteo: 1
}
sprint9>
```

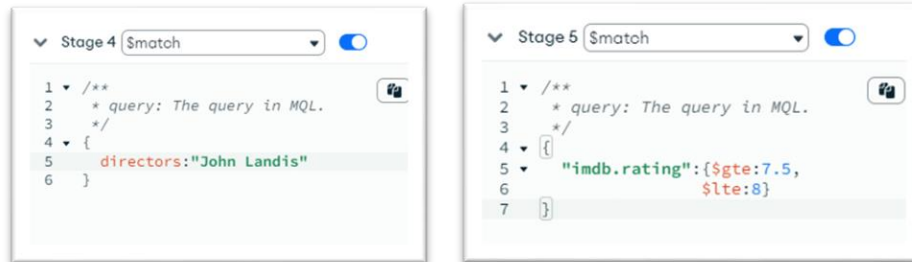
===== NIVEL 3 =====

Ejercicio 1

Encuentra todas las películas dirigidas por John Landis con una puntuación IMDb (Internet Movie Database) de entre 7,5 y 8

La consulta genera un total de 4 películas dirigidas por John Landis que cumplen con el rating imdb.

RESPUESTA: 4 registros de películas según la consulta

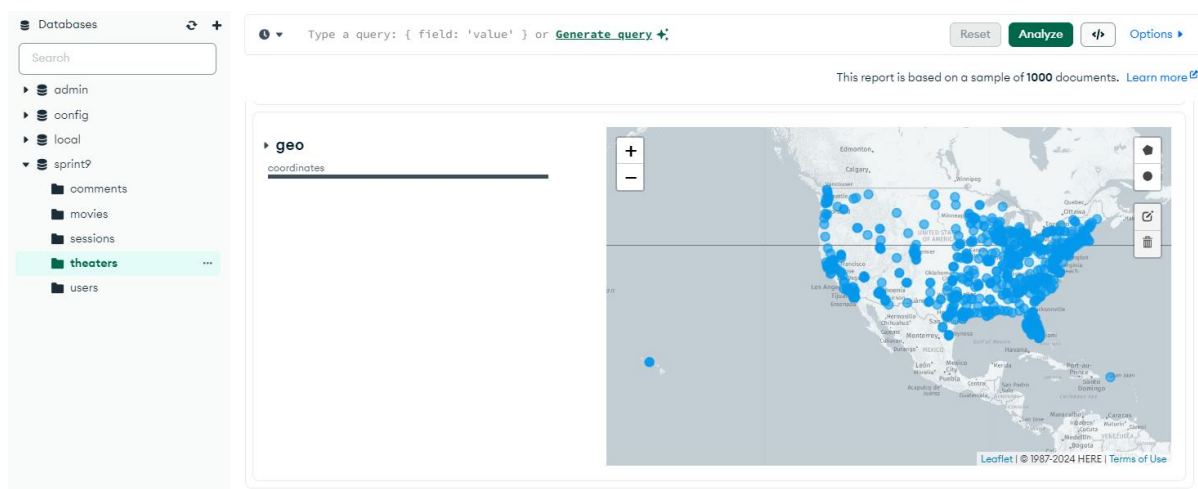


```
> db.movies.find({
  'type': 'movie',
  'directors': 'John Landis',
  'imdb.rating': { '$gte': 7.5, '$lte': 8 }
}) .count()
< 4
sprint9>
```

Ejercicio 2

Muestra en un mapa la ubicación de todos los teatros de la base de datos.

Para mostrar las ubicaciones geográficas de los teatros en el mapa, se accedió en analizado 'schema' de la colección 'theaters'. Aparecen automáticamente las ubicaciones por contener coordenadas, también podemos acceder a la opción geo-coordinates.



coordinates

coordinates

