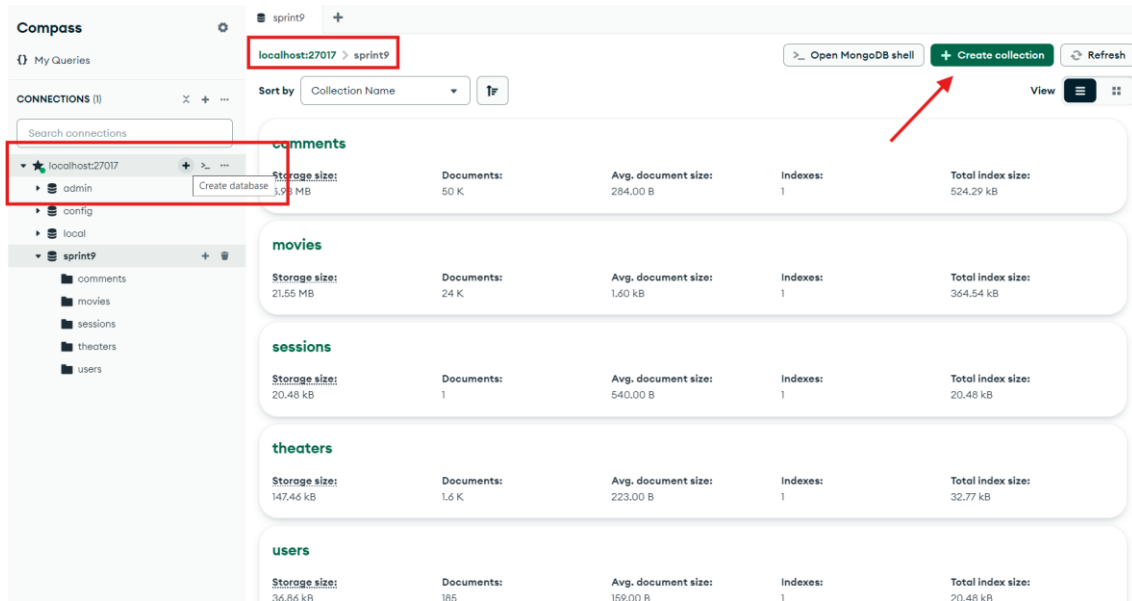


Sprint 9. Sara Gutierrez Amigo

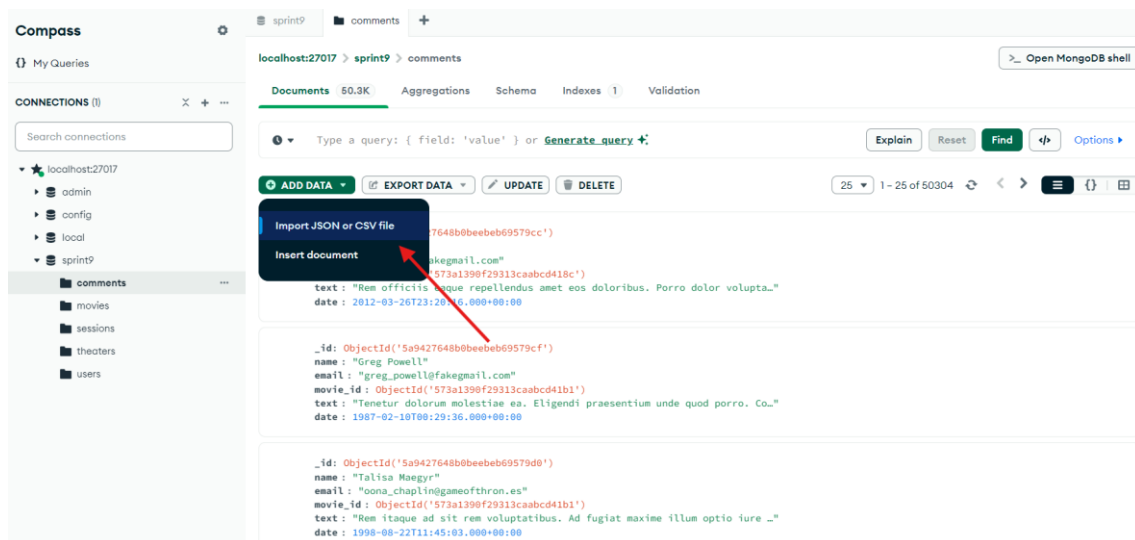
NIVEL 1

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

El primer paso es crear dentro de nuestro localhost la base de datos clicando en las indicaciones de la captura de pantalla adjunta.



Una vez creada la base de datos iremos creando las colecciones y es ahí donde cargaremos los archivos JSON.

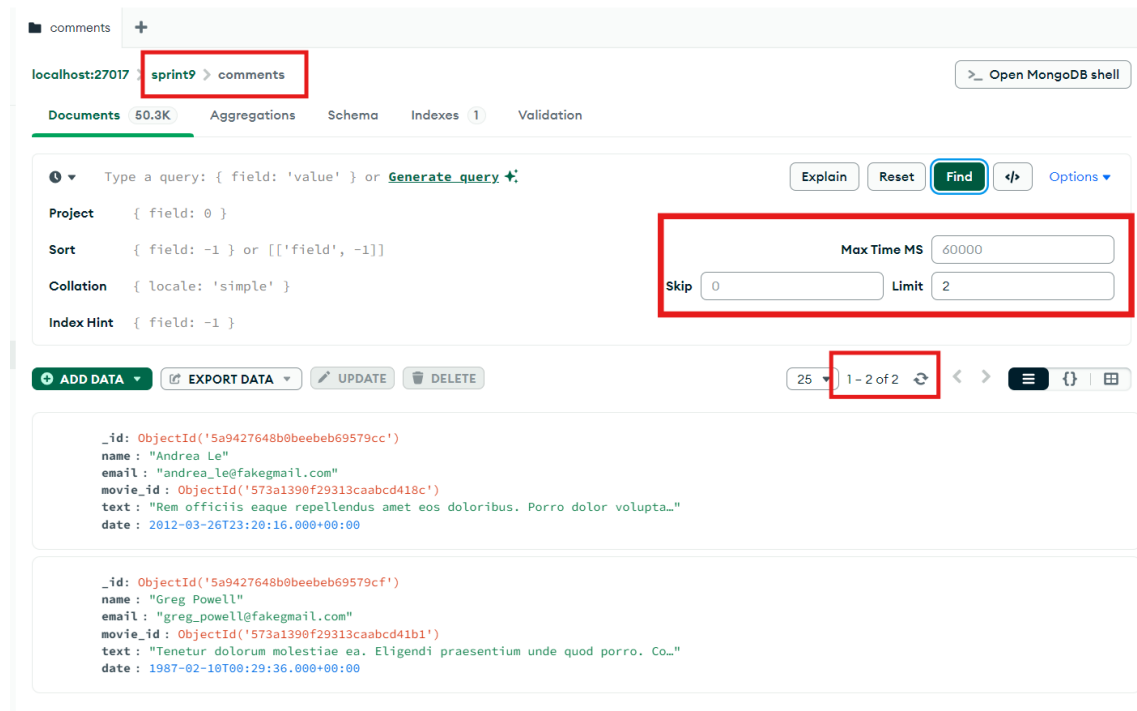


Sprint 9. Sara Gutierrez Amigo

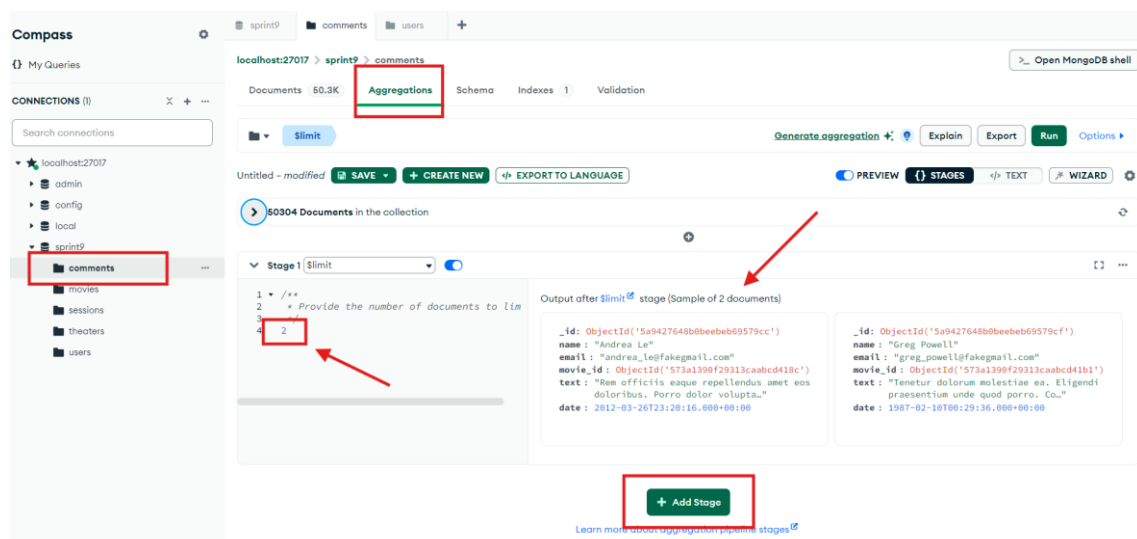
- Ejercicio 1

Mostra els 2 primers comentaris que hi ha en la base de dades.

Podemos hacerlo de dos maneras:



Seleccionamos primero la colección en la que queremos aplicar el filtro, en este caso será la de Colecciones. En el apartado de búsqueda (Find) ponemos que nos muestre como límite 2.

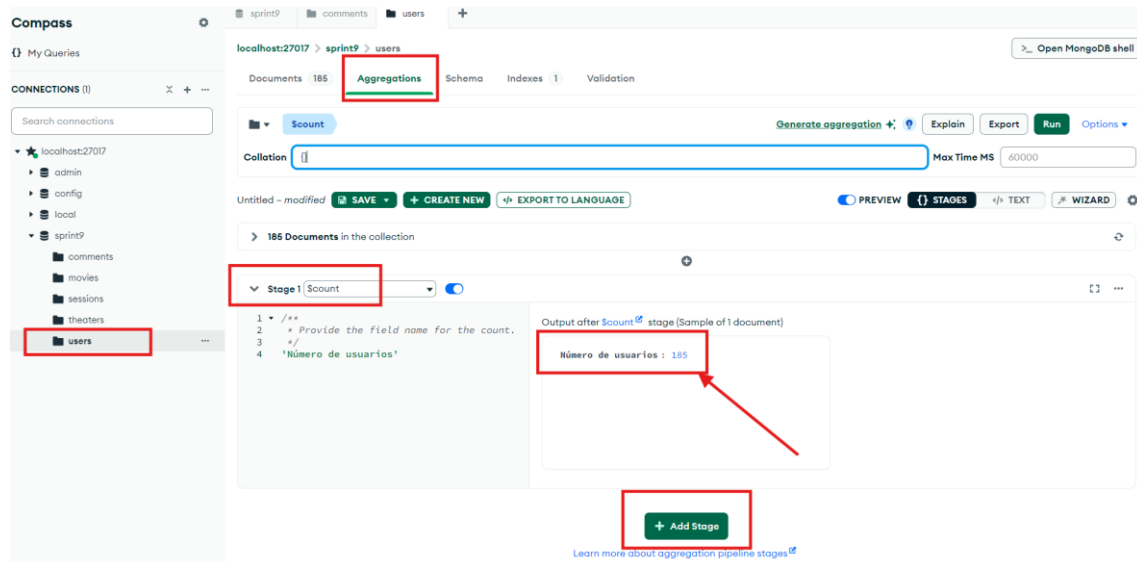


Si preferimos también podemos hacer la selección mediante una función de agregación, en este caso \$limit. Ya estamos en la colección de comentarios, pero esta vez vamos a la pestaña de Aggregations, ahí añadiremos una nueva etapa(+ Add Stage) y en la pantalla

Sprint 9. Sara Gutierrez Amigo

de comandos que nos aparece diremos que queremos ver las dos primeras líneas, simplemente poniendo el número 2. En la pantalla del Output de la derecha nos aparece el resultado.

Quants usuaris tenim registrats?

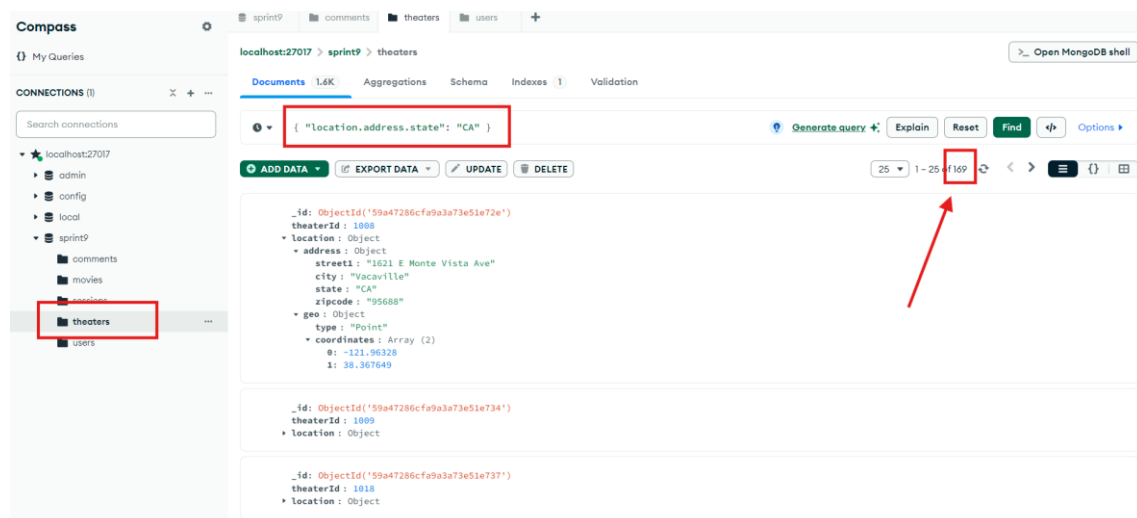


El número total de usuarios es 185.

Primero seleccionamos la colección desde donde queremos obtener información, en este caso Users. Después nos iremos al apartado de Agregaciones como en el ejercicio anterior y luego con Add Stage le añadimos la función de contaje(\$count). En la pantalla de comando podemos añadir el título del contaje que hemos realizado. El resultado nos aparece en el Output.

Quants cinemes hi ha en l'estat de Califòrnia?

Como en el primer ejercicio también tenemos dos opciones:



Sprint 9. Sara Gutierrez Amigo

Podemos hacerlo aplicando el filtro directamente y como siempre trabajando desde la colección que queremos saber la información. Para luego fijarnos en cuantas filas nos ha devuelto (el número que indica la flecha)

O bien con funciones de agregación y aplicando en este caso dos Stages. Uno inicial donde le indicaremos la condición desde donde queremos que empiece a trabajar, con \$match y después una segunda función de contaje (\$count) para que nos indique la cantidad de aquello que hemos filtrado previamente:

The screenshot shows the MongoDB Compass interface. The left sidebar shows the database structure with collections: admin, config, local, sprint9, comments, movies, sessions, theaters, and users. The main panel is set to 'localhost:27017 > sprint9 > theaters'. The 'Aggregations' tab is active, showing a pipeline with two stages:

- Stage 1 \$match**: The query is `{ "location.address.state": "CA" }`. The output shows a sample of 10 documents.
- Stage 2 \$count**: The query is `{ "Numero de cines en el estado de California": 1 }`. The output shows a sample of 1 document with the result: `Número de cines en el estado : 169 de California`. A red arrow points to the number 169.

El número total de cines en el estado de California es 169.

Quin va ser el primer usuari/ària en registrar-se?

The screenshot shows the MongoDB Compass interface. The left sidebar shows the database structure with collections: admin, config, local, sprint9, comments, movies, sessions, theaters, and users. The main panel is set to 'localhost:27017 > sprint9 > users'. The 'Documents' tab is active, showing a query: `{ field: 'value' }` or `Generate query`. The query is `{ "_id": 1 }`. The 'Sort' field is set to `{ "_id": 1 }`. The 'Limit' field is set to `1`. The output shows a sample of 1 document with the result: `{ "_id": ObjectId("59b99db4cf9a34dc7885b6"), "name": "Ned Stark", "email": "sean_bean@gameofthron.es", "password": "52b5125UREFwsR0yF8CRqGK0Lz00HN/jLHgUNN139R3AqHUQ74cr131Vu" }`.

Ned Stark es el primer usuario que se inscribió.

Primero hemos ordenado de manera ascendente el campo de usuario, ya que eso es lo que nos va a permitir saber cual fue el primero en inscribirse y luego hemos limitado esa búsqueda a 1.

Sprint 9. Sara Gutierrez Amigo

Quantes pel·lícules de comèdia hi ha en la nostra base de dades?

The screenshot shows the MongoDB Compass interface. In the left sidebar, the 'movies' collection is selected. The main panel displays an aggregation pipeline with two stages:

- Stage 1 (\$match):** A query is provided to filter documents where the 'genres' array contains 'Comedy'.

```
1 /*  
2  * query: The query in MQL.  
3  */  
4 {  
5   "genres": "Comedy",  
6 }  
7
```
- Stage 2 (\$count):** A query is provided to count the documents.

```
1 /*  
2  * Provide the field name for the count.  
3  */  
4 'Películas de comedia'
```

The output of the \$count stage is shown as 'Películas de comedia: 7024', which is highlighted with a red box and an arrow.

Hay 7024 películas de comedia.

Hacemos como en el ejercicio anterior. En el apartado de Aggregations aplicamos dos funciones. Primero filtramos el género para que nos seleccione solo las de comedia y luego hacemos un count de ese filtro aplicado.

- Exercici 2

Mostra'm tots els documents de les pel·lícules produïdes en 1932, però que el gènere sigui drama o estiguin en francès.

The screenshot shows the MongoDB Compass interface. A query is entered in the main panel:

```
{  
  "$and": {  
    "year": 1932,  
    "$or": {  
      "genres": "Drama",  
      "languages": "French"  
    }  
  }  
}
```

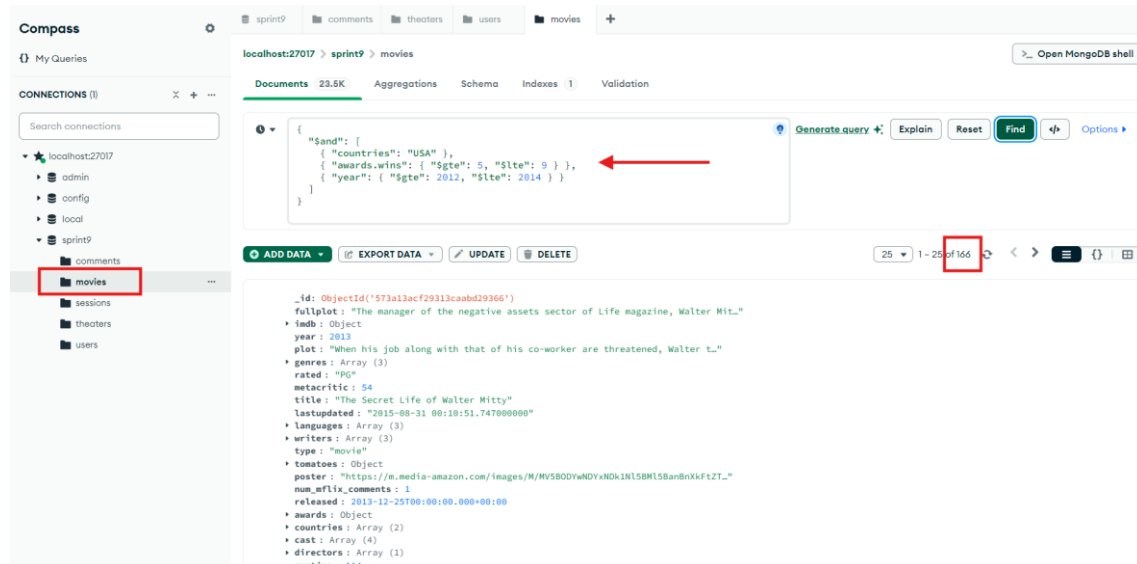
Red arrows point to the '\$and' and '\$or' operators. The query is executed, and the results are shown in a list. The first document is highlighted, showing details like 'The Blood of a Poet'.

En este caso no trabajamos con agregaciones. Hacemos la consulta directamente y le aplicamos las condiciones que necesitamos. Tal y como pone en el ejercicio con un OR (\$or) y un AND(\$and).

Sprint 9. Sara Gutierrez Amigo

- Exercici 3

Mostra'm tots els documents de pel·lícules estatunidenques que tinguin entre 5 i 9 premis que van ser produïdes entre 2012 i 2014.



The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' panel lists the database 'sprint9' and its collections: 'comments', 'movies', 'theaters', and 'users'. The 'movies' collection is selected. The main panel displays a query in the 'Documents' tab:

```
{
  "$and": [
    { "countries": "USA" },
    { "awards.wins": { "$gte": 5, "$lte": 9 } },
    { "year": { "$gte": 2012, "$lte": 2014 } }
  ]
}
```

A red arrow points to the query. Below the query, the '1 - 25 of 166' result count is shown, with '166' highlighted by a red box. The first document is expanded, showing details like title, year, genres, and awards.

Nos muestra 166 documentos que cumplen esas condiciones.

Volvemos a trabajar haciendo la consulta directamente. Como país le diremos que filtre los de USA y para el filtro tanto de premios como de año de producción utilizaremos los comandos \$gte y \$lte ("greater than or equal to" y "less than or equal to", mayor o igual que y menor o igual que).

NIVEL 2

- Exercici 1

Compte quants comentaris escriu un usuari/ària que utilitza "GAMEOFTHRON.ES" com a domini de correu electrònic.

Sprint 9. Sara Gutierrez Amigo

The screenshot shows the MongoDB Compass interface. On the left, the 'collections' list on the 'sprint9' database has 'comments' highlighted. The main panel shows the 'Aggregations' tab for the 'comments' collection. The aggregation pipeline consists of two stages:

- Stage 1: \$match** (highlighted with a red box). The query is:

```
{ "email": { "$regex": "@gameofthron\\.es$" } }
```

 (The entire stage configuration is also highlighted with a red box). The output shows a sample of 10 documents.
- Stage 2: \$count** (highlighted with a red box). The query is:

```
'Número de comentarios'
```

 (The entire stage configuration is also highlighted with a red box). The output shows a sample of 1 document:

```
{ "count": 22841 }
```

 (The output value is highlighted with a red box).

Volvemos a trabajar desde agregaciones. Primero aplicamos un primer Stage con \$match donde le decimos que nos seleccione los emails con el dominio correspondiente. Para ello, utilizamos la función \$regex ("email": { "\$regex": "@gameofthron\\.es\$" })). Le estamos pidiendo que nos busque los documentos que en el email tenga ese texto específico, en concreto los que acaben por este texto que le pasamos. Eso se lo indicamos con la "\$" final. Si quisiéramos que encontrara un texto que empiece por X, pondríamos un "^" al inicio del texto. Ponemos las dos "\\" delante del punto, para que considere el punto de manera literal.

Por último, creamos el Stage del conteo donde nos indica que han sido 22.841 comentarios.

- Exercici 2

Quants cinemes hi ha en cada codi postal situats dins de l'estat Washington D. C. (DC)?

The screenshot shows the MongoDB Compass interface. On the left, the 'collections' list on the 'sprint9' database has 'theaters' highlighted. The main panel shows the 'Aggregations' tab for the 'theaters' collection. The aggregation pipeline consists of two stages:

- Stage 1: \$match** (highlighted with a red box). The query is:

```
{ "location.address.state": "DC" }
```

 (The entire stage configuration is also highlighted with a red box). The output shows a sample of 3 documents.
- Stage 2: \$group** (highlighted with a red box). The query is:

```
{ "_id": "$location.address.zipcode", "NúmeroDeCines": { "$sum": 1 } }
```

 (The entire stage configuration is also highlighted with a red box). The output shows a sample of 3 documents, where the 'NúmeroDeCines' field is highlighted with a red box.

Sprint 9. Sara Gutierrez Amigo

Volvemos a trabajar con agregaciones, aplicamos el primer filtro donde le decimos que el estado sea DC y luego añadimos un nuevo Stage donde hacemos un GroupBy diciéndole que nos lo agrupe por código postal de manera que empieza sumando 1 a cada uno de esos grupos.

The screenshot shows the MongoDB Compass interface. On the left, the aggregation pipeline is defined in the 'AGGREGATION' tab:

```
1 {
2   {
3     $match:
4     /**
5      * query: The query in MQL.
6      */
7     {
8       "location.address.state": "DC"
9     }
10  },
11  {
12    $group:
13    /**
14     * _id: The id of the group.
15     * fieldN: The first field name.
16     */
17    {
18      _id: "$location.address.zipcode",
19      NúmeroDeCines: {
20        $sum: 1
21      }
22    }
23  }
24 }
```

On the right, the 'PIPELINE OUTPUT' tab shows a sample of 3 documents:

```
{ "_id": "20010", "NúmeroDeCines": 1 }
{ "_id": "20002", "NúmeroDeCines": 1 }
{ "_id": "20016", "NúmeroDeCines": 1 }
```

El output que nos devuelve es que hay 1 cine en el CP 20010, uno en el CP 20002 y también sólo un cine en el CP 20016.

NIVEL 3

- Exercici 1

Troba totes les pel·lícules dirigides per John Landis amb una puntuació IMDb (Internet Movie Database) d'entre 7,5 i 8.

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' panel shows the 'movies' collection selected. In the center, the 'QUERY' tab shows the following query:

```
{
  "$and": [
    { "directors": "John Landis" },
    { "imdb.rating": { "$gte": 7.5, "$lte": 8 } }
  ]
}
```

On the right, the 'RESULTS' tab shows a table of 4 movies:

#	movies	_id	ObjectId	fullplot	String	imdb	Object	year	Int32	plot	String	genres	Ar
1		ObjectId('573a1397f29313c...		"Faber College has one fr...		()	3 fields	1978		"At a 1962 college, Dean ...	()	1 elem	
2		ObjectId('573a1397f29313c...		"After the release of Jak...		()	3 fields	1980		"Jake Blues, just out fro...	()	3 elem	
3		ObjectId('573a1397f29313c...		"Two American college stu...		()	3 fields	1981		"Two American college stu...	()	2 elem	
4		ObjectId('573a1398f29313c...		"Louis Winthorpe is a bus...		()	3 fields	1983		"A snobbish investor and ...	()	1 elem	

Las películas que cumplen estas condiciones son 4.

Volvemos a actuar haciendo la consulta directamente, utilizando un AND y \$gte, \$lte para indicar el rango de IMDb.

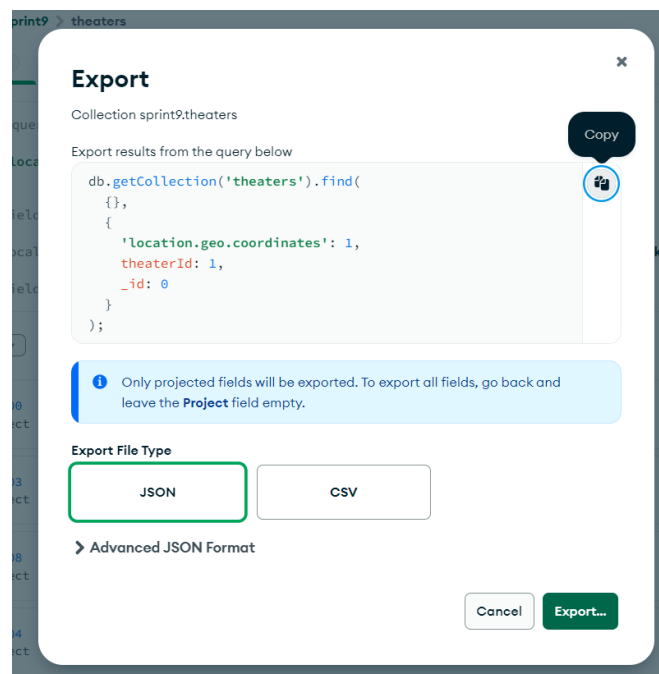
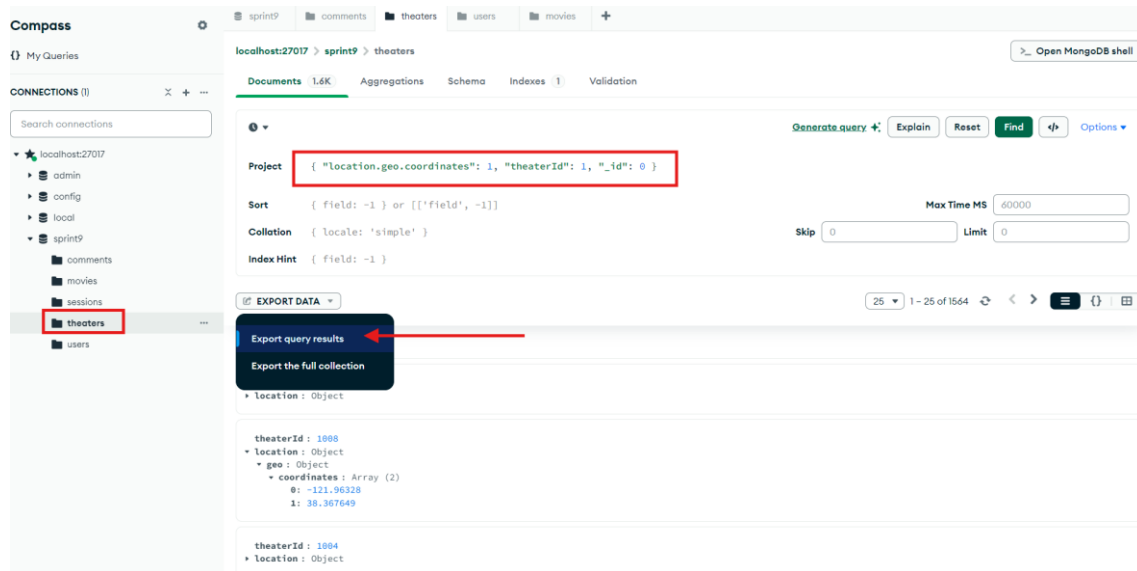
Sprint 9. Sara Gutierrez Amigo

- Exercici 2

Mostra en un mapa la ubicació de tots els teatres de la base de dades.

Vamos a hacerlo a través de Power BI.

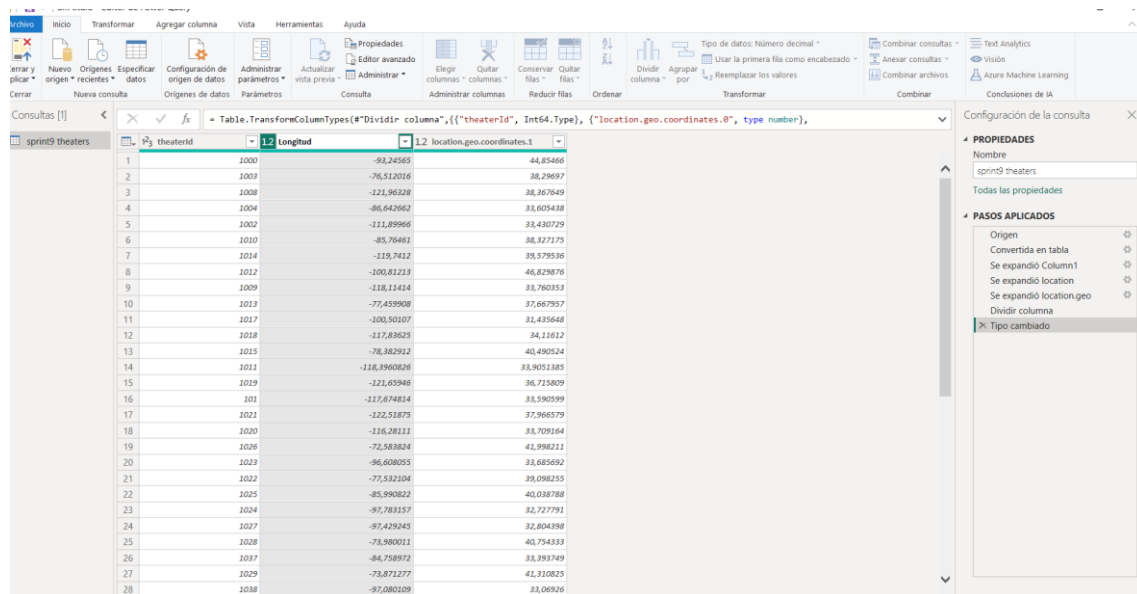
Para ello primero vamos a crear un archivo JSON con la data que necesitamos para este gráfico. Aplicaremos primero el filtro: en este archivo queremos que nos incluya las coordenadas y el id de los cines.



Una vez descargado empezamos el proceso en Power BI, obteniendo los datos desde el archivo JSON.

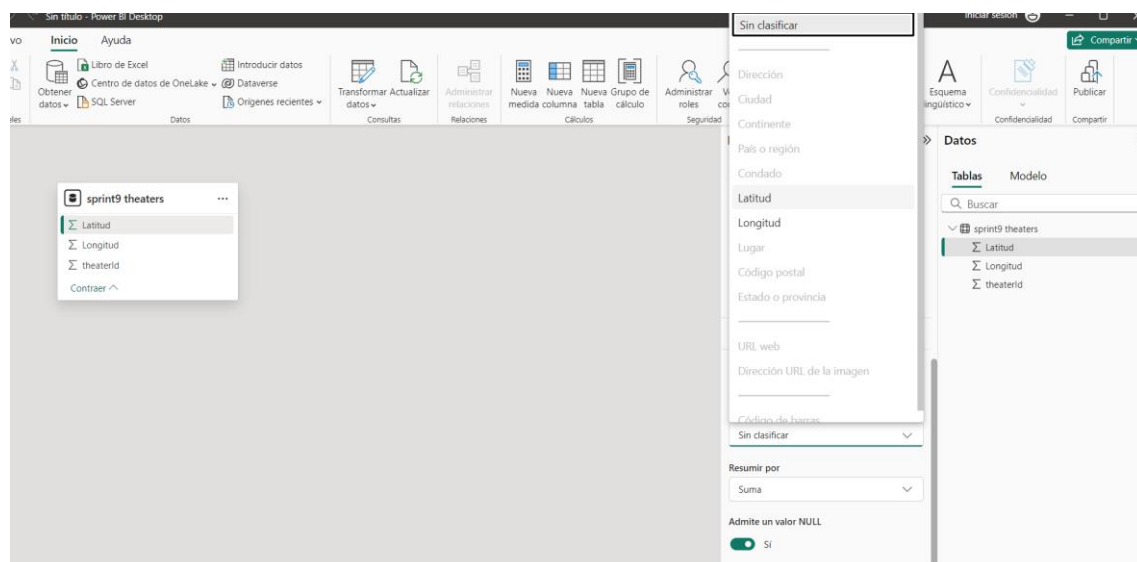
Sprint 9. Sara Gutierrez Amigo

NOTA: Lo hemos probado exportando el archivo como CSV, pero los datos no se exportan correctamente y da muchos problemas al intentar transformarlos en Power BI.



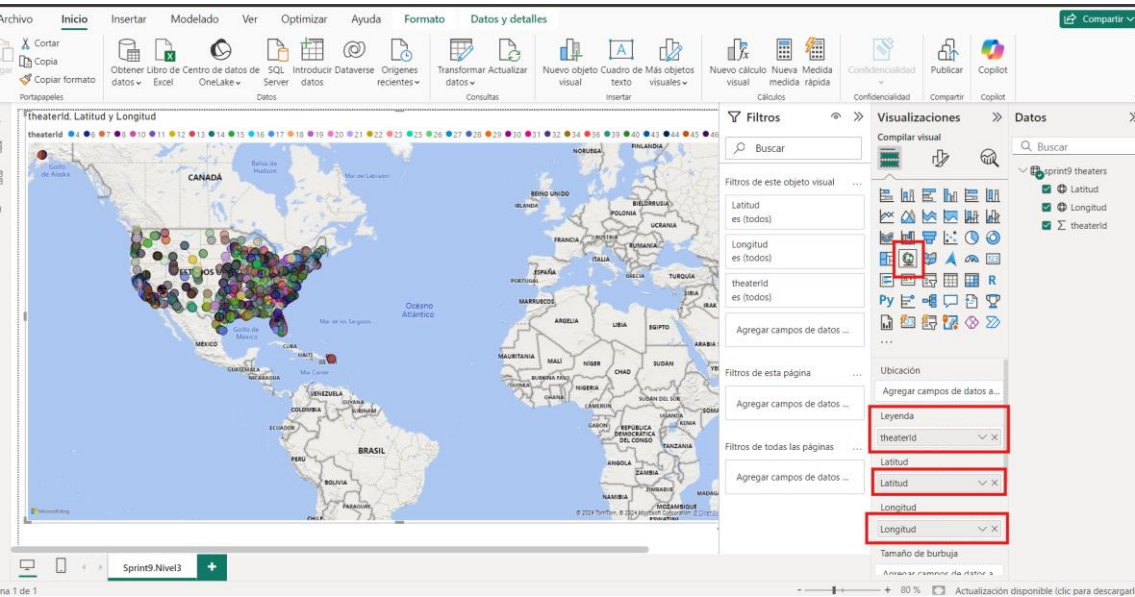
	theaterid	Longitud	location.geo.coordinates.1
1	1000	-91.24585	44.85466
2	1003	-76.512016	38.29697
3	1008	-121.96328	38.367649
4	1004	-86.642662	33.605438
5	1002	-111.89966	33.430729
6	1010	-85.76481	38.327175
7	1014	-119.7412	35.579536
8	1012	-100.81213	46.829876
9	1009	-118.11414	33.760353
10	1013	-77.459908	37.667957
11	1017	-100.50107	31.435648
12	1018	-117.83625	34.11612
13	1015	-78.382912	40.480524
14	1011	-118.3960826	33.9051385
15	1019	-121.65946	36.715809
16	101	-117.674814	33.590599
17	1021	-122.51875	37.966579
18	1020	-116.28111	33.709164
19	1026	-72.583824	41.998211
20	1023	-96.608055	33.685692
21	1022	-77.532104	39.082555
22	1025	-85.990822	40.038788
23	1024	-97.783157	32.727791
24	1027	-97.429245	32.804398
25	1028	-73.980011	40.754333
26	1037	-84.758972	33.383749
27	1029	-73.871277	41.310825
28	1038	-97.080109	33.06926

Hacemos los cambios necesarios a los datos. En este caso averiguamos que campo corresponde la latitud y cual a la longitud y renombramos las columnas y aplicamos la categoría de los datos que le corresponde:



Escogemos como gráfico un mapa donde establecemos las variables de latitud, longitud y como leyenda el id de los cines por ser lo que se ve más claro visualmente:

Sprint 9. Sara Gutierrez Amigo



Finalmente, el mapa que obtenemos es este:

