

Tasca S9.01. Consultes amb MongoDB

Descripció

Treballarem amb una base de dades que conté col·leccions relacionades amb una aplicació d'entreteniment cinematogràfic:

- **users:** Emmagatzema informació d'usuaris/es, incloent-hi noms, emails i contrasenyes xifrades.
- **theatres:** Conté dades de cinemes, com ID, ubicació (direcció i coordenades geogràfiques).
- **sessions:** Guarda sessions d'usuari, incloent-hi ID d'usuari i tokens JWT per a l'autenticació.
- **movies:** Inclou detalls de pel·lícules, com a trama, gèneres, durada, elenc, comentaris, any de llançament, directors, classificació i premis.
- **comments:** Emmagatzema comentaris d'usuaris/es sobre pel·lícules, amb informació de l'autor/a del comentari, ID de la pel·lícula, text del comentari i la data.

Duràs a terme algunes consultes que et demana el client/a, el qual està mesurant si seràs capaç o no de fer-te càrrec de la part analítica del projecte vinculat amb la seva base de dades.

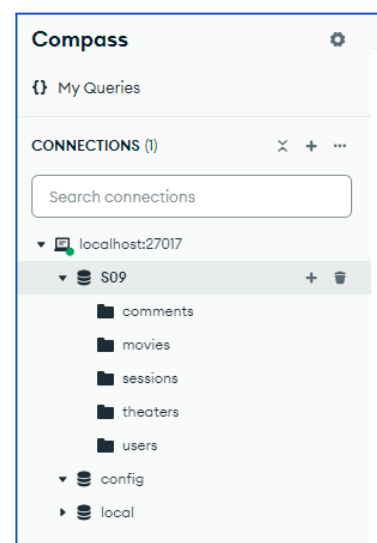
★ Nivell 1

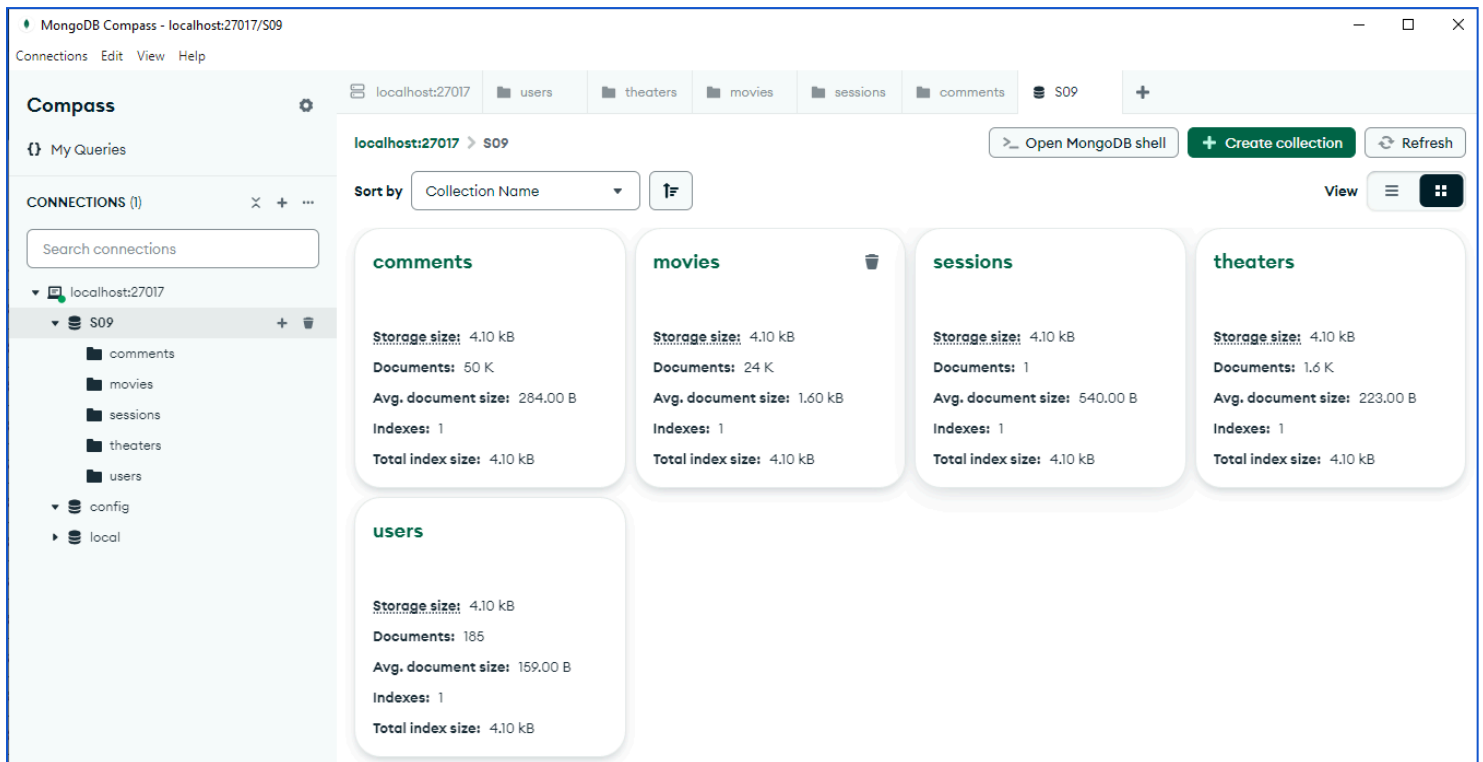
Crea una base de dades amb MongoDB utilitzant com a col·leccions els arxius adjunts.

He baixat MongoDB Compass per Windows seguint aquest tutorial:

https://www.youtube.com/watch?v=AVYKn_3rozI

He creat una nova **database** anomenada **S09** per emmagatzemar totes les taules proporcionades per aquest exercici, i les he anat guardant en noves **col·leccions** cada arxiu **.json**. Les importacions no han donat errors i han quedat així →

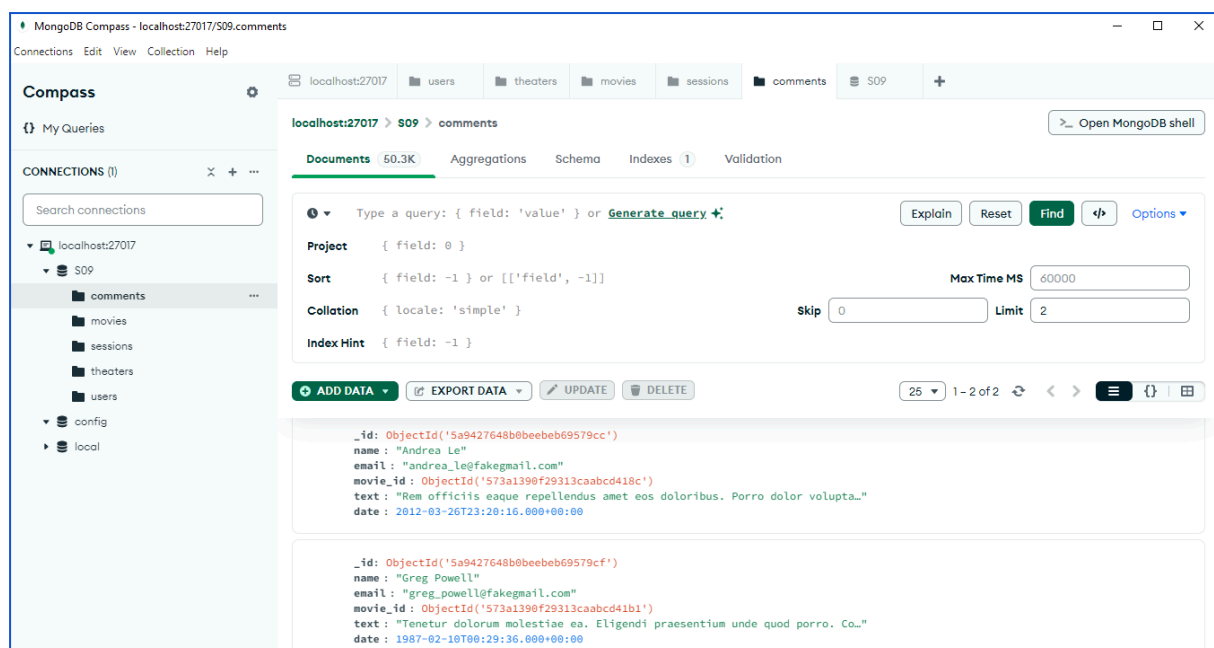




1.1. Exercici 1

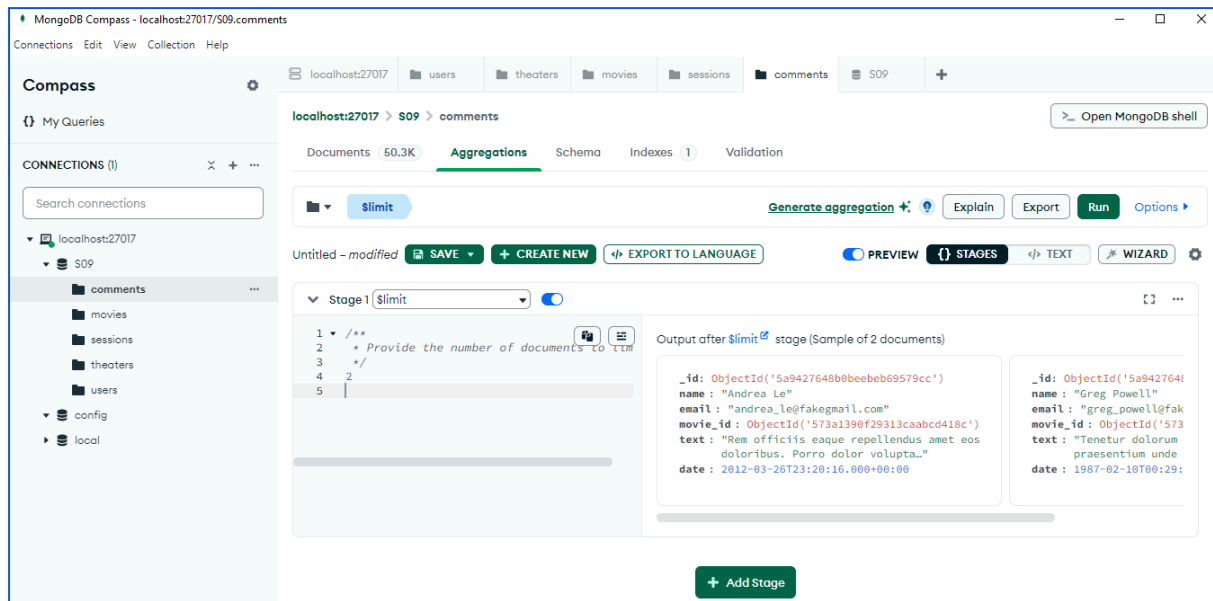
- Mostra els 2 primers comentaris que hi ha a la base de dades.

Per fer això hem d'anar a la col·lecció **comments** i a **options** filtrar **Limit 2**:



Una altra manera de trobar aquests dos primers comentaris és:

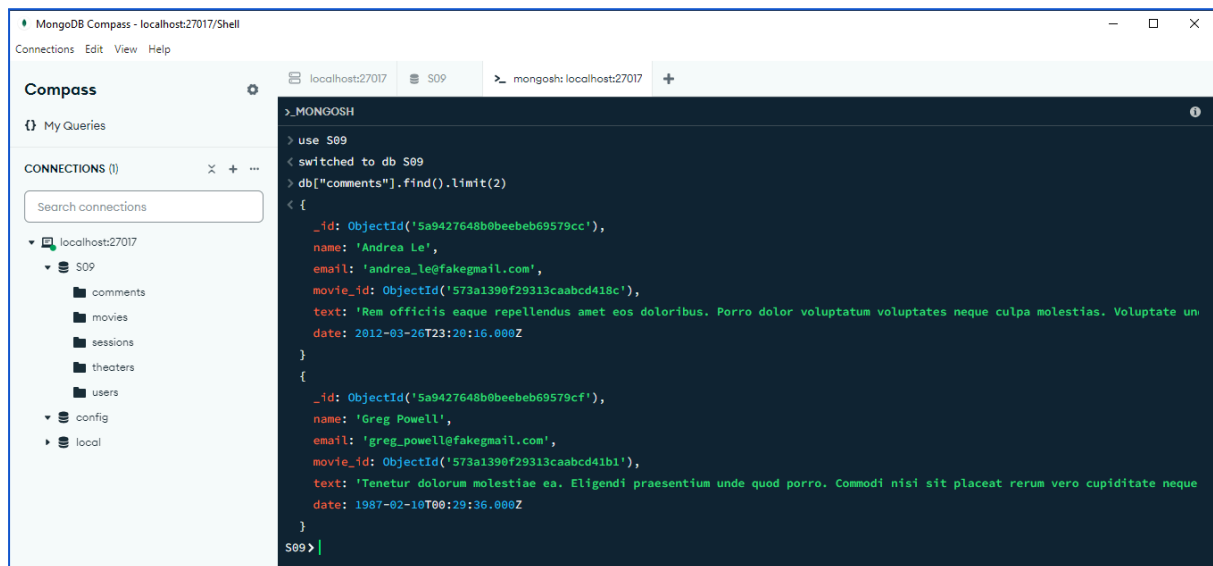
Aggregations → **Add Stage** → Stage 1: **\$limit** → (escriure: **2** allà on hi diu: *number*)



Per últim podem accedir als 2 primers comentaris a través de la Shell:

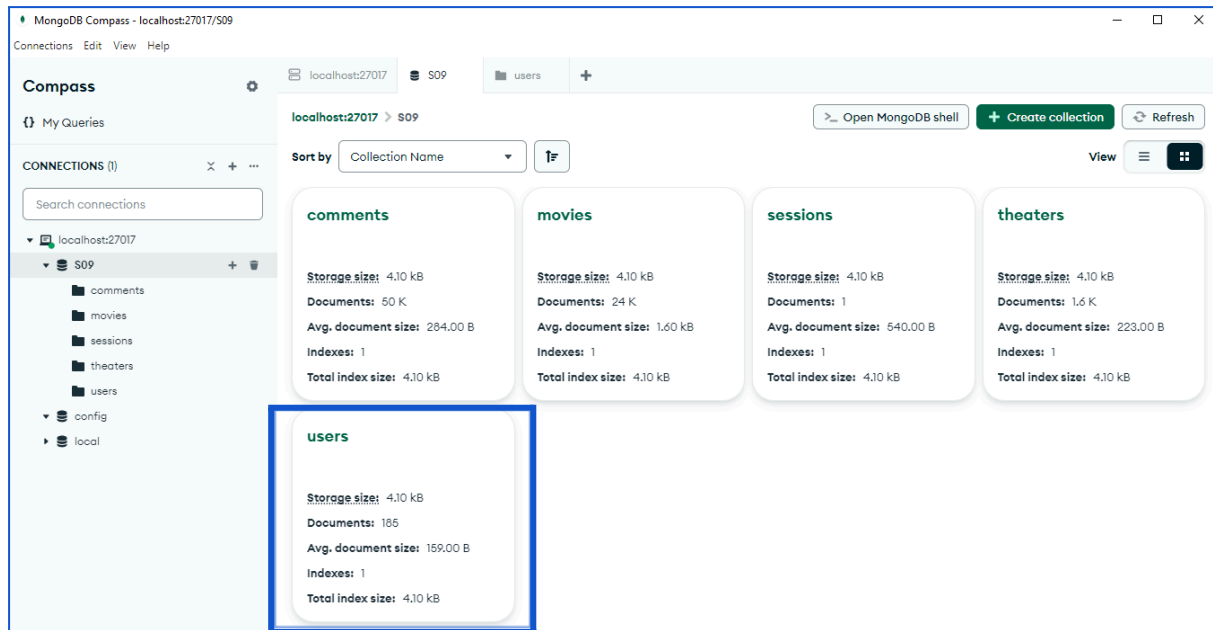


Afegint: **.limit(2)** al codi que ja venia per defecte: **db["comments"].find()**:

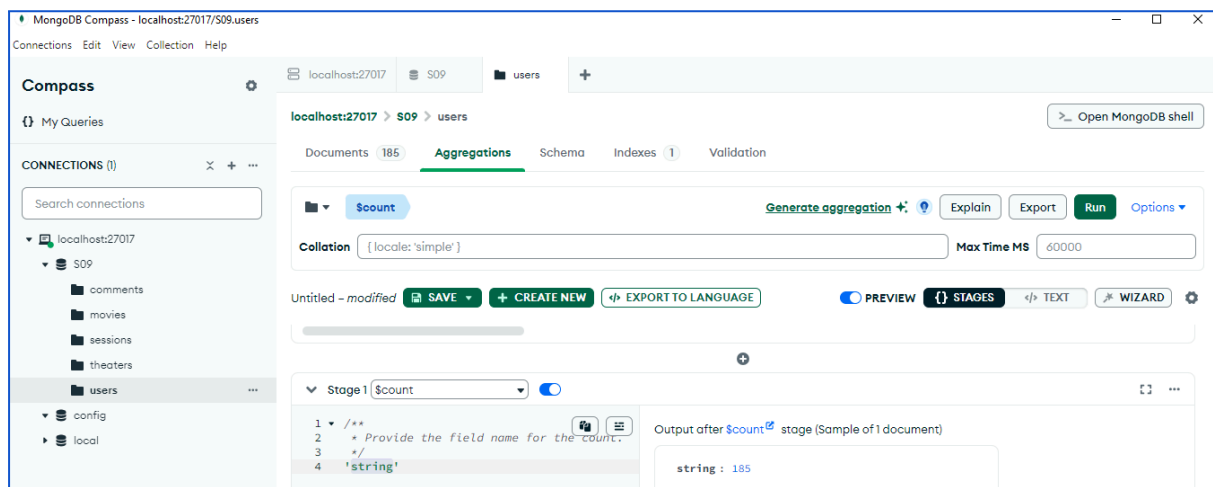


- Quants usuaris tenim registrats?

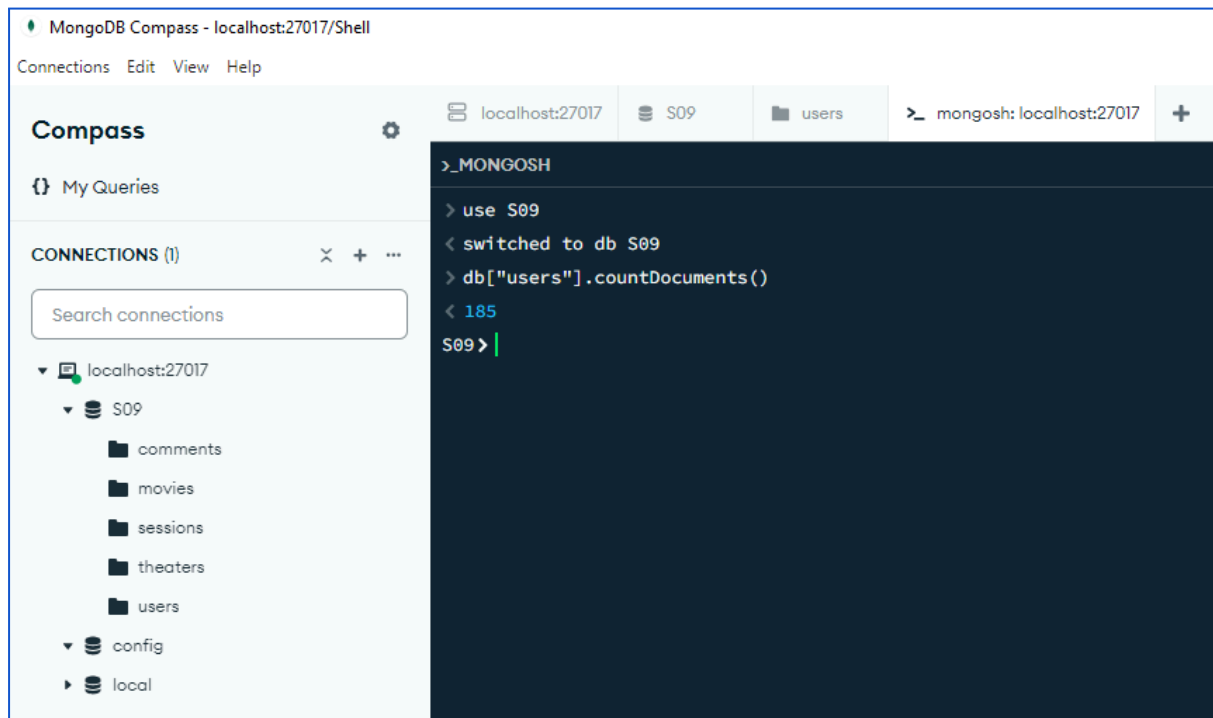
A primer cop d'ull podríem dir que n'hi ha **185**:



Si des d'**Aggregations** → **Add Stage** fem servir un **\$count** també ens diu **185**:



I si fem servir el **.countDocuments()** des de la **Shell** obtenim el mateix resultat:

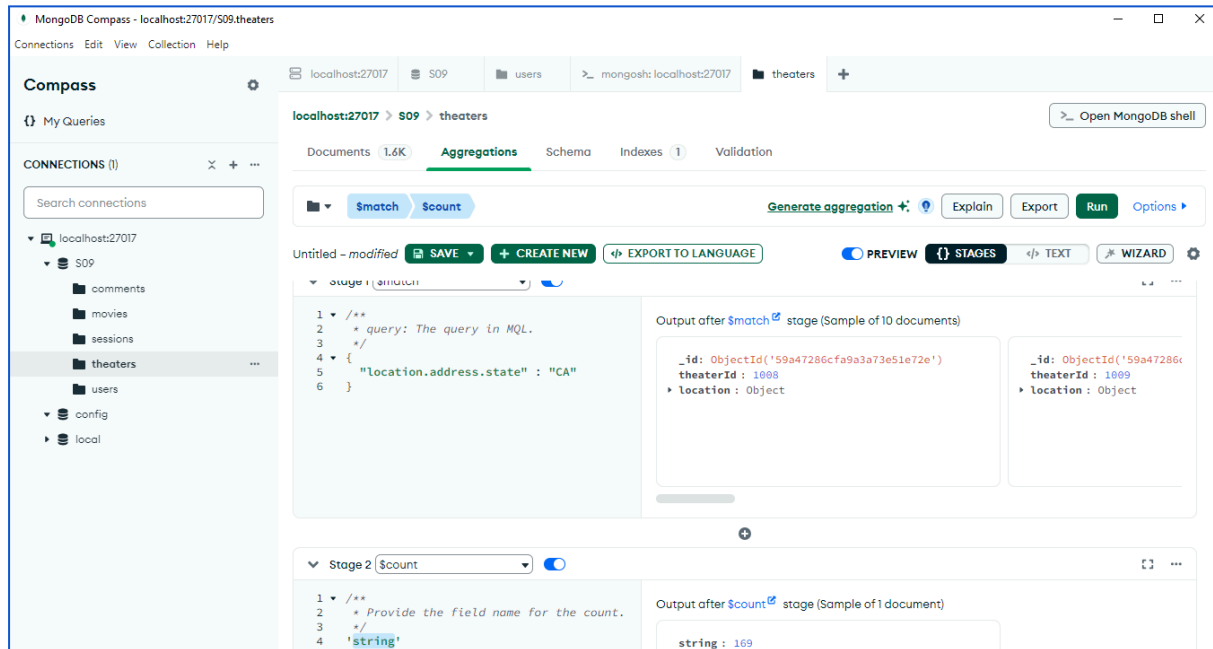


Per últim he volgut comprovar que no hi havia cap usuari repetit fent servir un **.distinct()**:

```
> db.users.distinct("email").length
< 185
```

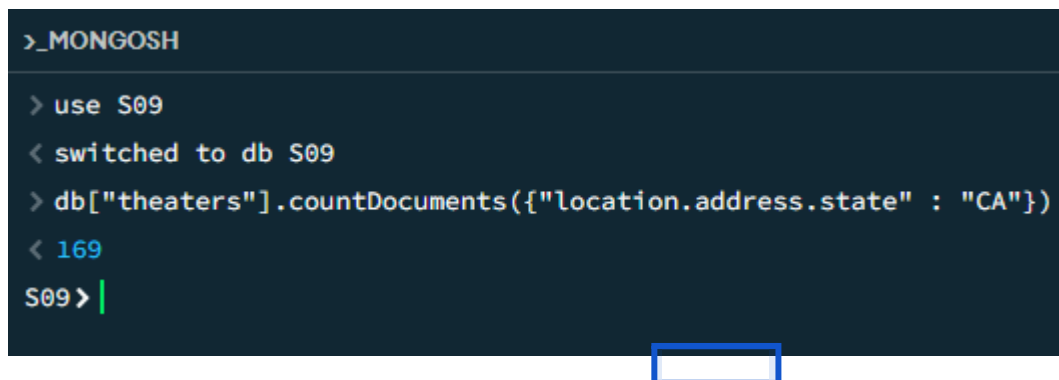
- Quants cinemes hi ha a l'estat de Califòrnia?

Si ho mirem des d'**Aggregations** → **Add Stage** i filtrem per l'estat de *Califòrnia* amb un **\$match**, i després fem un **\$count**:



i veiem que el resultat és 169.

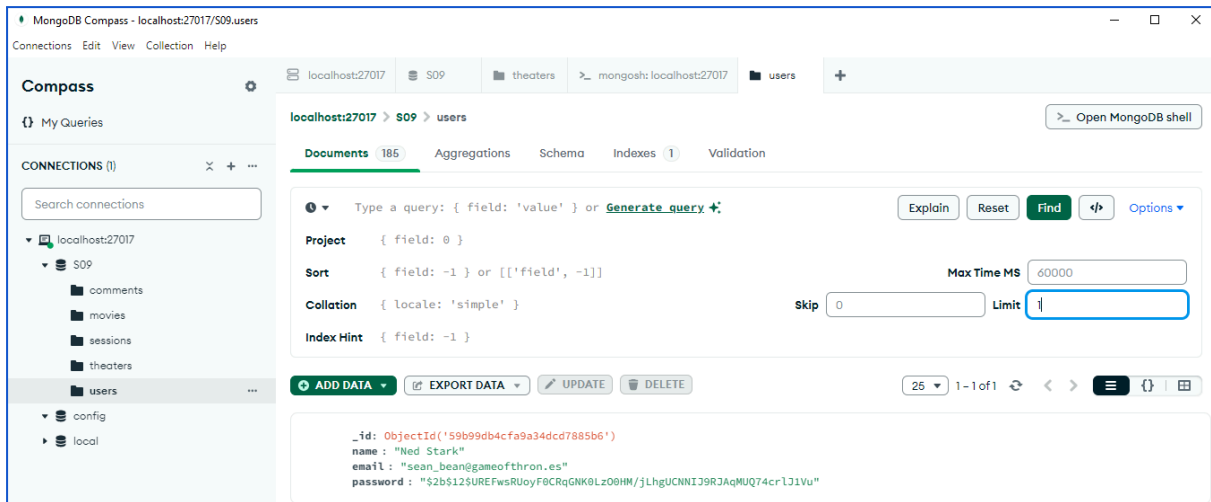
Si ho fem a través de la shell:



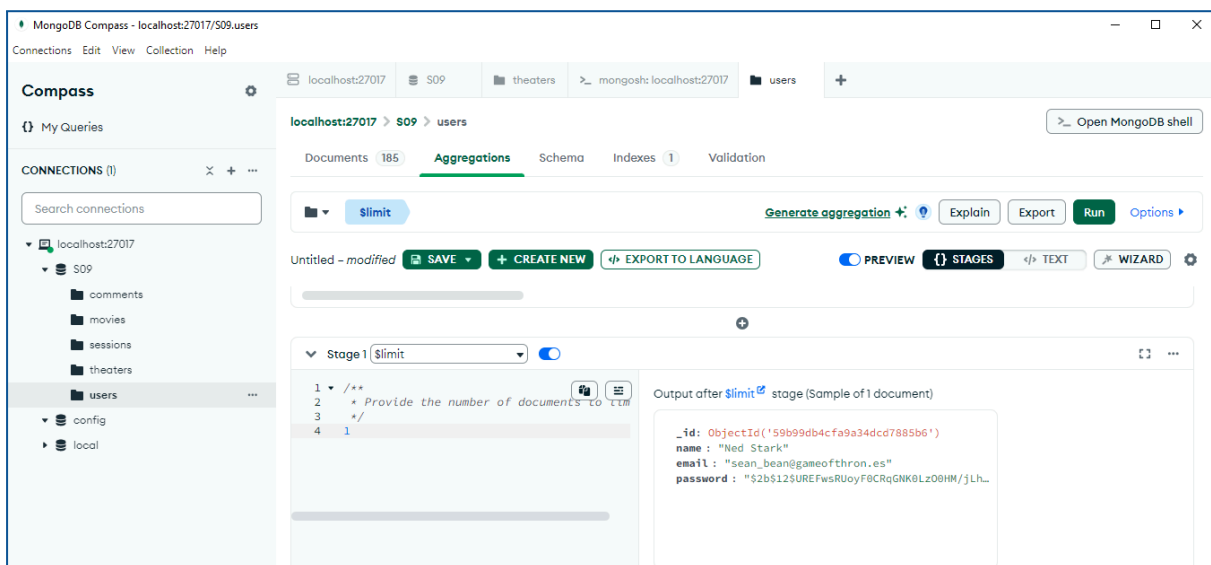
Veiem que obtenim el mateix resultat. **169**.

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

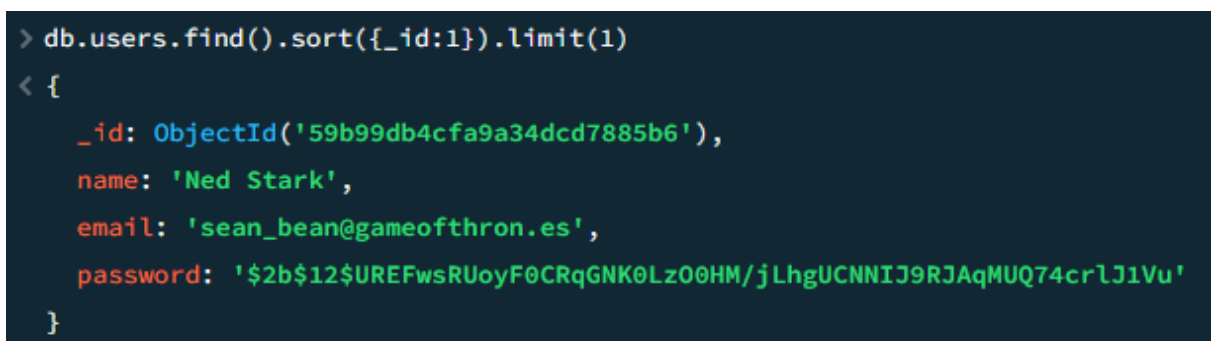
Si **limitem** els documents de la col·lecció **users** a **1**, per poder veure només el 1r, ens trobarem en **Ned Stark**:



Aquest mateix **limit** el podem posar desde **Aggregations**:



O des de la **shell**. Endreçant els documents per **ordre d'id** i **filtrant pel 1r**:




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


Si ho volem mirar des d'**Aggregations** primer haurem de **filtrar per pel·lícules**:


```
Stage1 $match   
1 ▾ /**  
2   * query: The query in MQL.  
3   */  
4 ▾ {  
5   type: "movie"  
6 }  
7
```

després per **Comèdia**:

```
Stage2 $match   
1 ▾ /**  
2   * query: The query in MQL.  
3   */  
4 ▾ {  
5   genres: "Comedy"  
6 }
```

i per últim demanar el **recompte**:

```
Stage3 $count   
1 ▾ /**  
2   * Provide the field name for the count.  
3   */  
4 'string'
```

Output after \$count  stage (Sample of 1 document)
string : 7002

Això ens aboca un resultat de **7002 pel·lícules de comèdia**.

Si ho volem mirar des de la **shell**, ho farem així:

```
> db.movies.countDocuments({type:"movie", genres:"Comedy"})  
< 7002
```

i ens donarà el mateix resultat :)

1.2. Exercici 2

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

A través d'**Aggregations** → **Stage** haurem d'aplicar diversos filtres:

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

```
1 ▾ /**
2   * query: The query in MQL.
3   */
4 ▾ {
5   year: 1932
6 }
```

```
1 ▾ /**
2   * query: The query in MQL.
3   */
4 ▾ {
5   $or:
6   [
7     { languages: "French" },
8     { genres: "Drama" }
9   ]
10 }
```

Output after \$match stage (Sample of 10 documents)

```
_id: ObjectId('573a1391f29313caabcd9458')
plot: "A young artist draws a face at a canvas on his easel. Suddenly the mou..."
runtime: 55
rated: "UNRATED"
cast: Array (4)
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWYtY..."
```

```
_id: ObjectId('573a13921
plot: "Junta is hated b
village where she
genres: Array (3)
runtime: 85
cast: Array (4)
poster: "https://m.medi
amazon.com/imag
title: "The Blue Light"
```

i si fem scroll a la dreta els podrem veure tots.

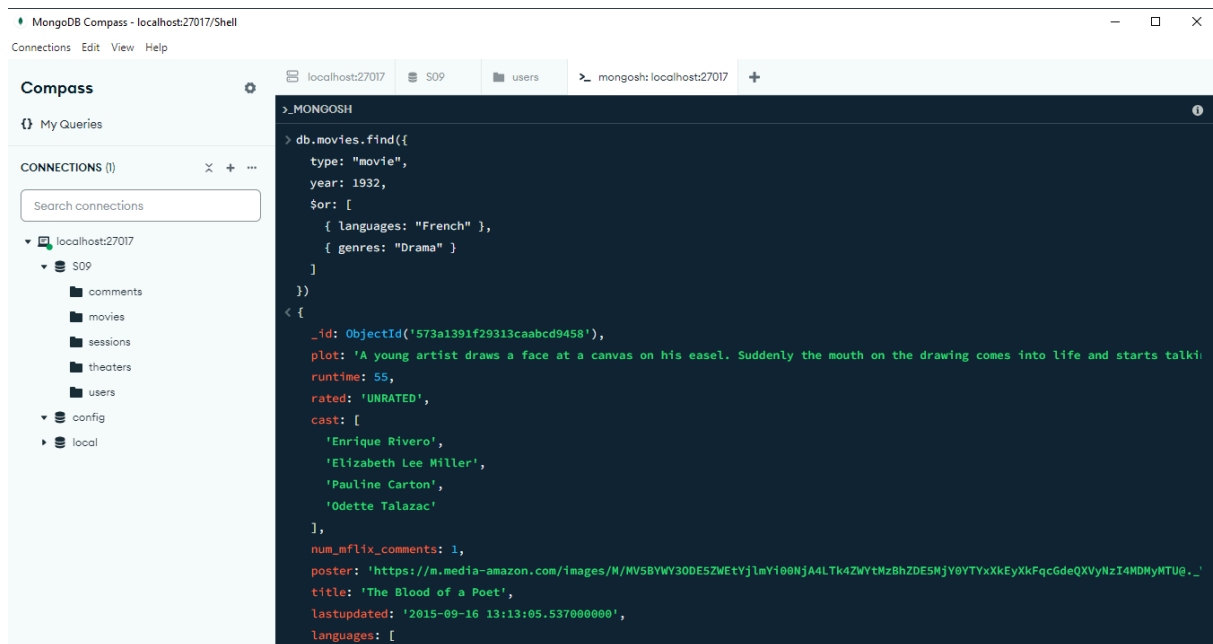
En total són 18. L'exercici no demana el recompte però està bé saber quants són.

```
1 ▾ /**
2   * Provide the field name for the count.
3   */
4   'string'
```

Output after \$count stage (Sample of 1 document)

```
string: 18
```

Per veure els documents a través de la **shell** haurem de demanar-ho així:

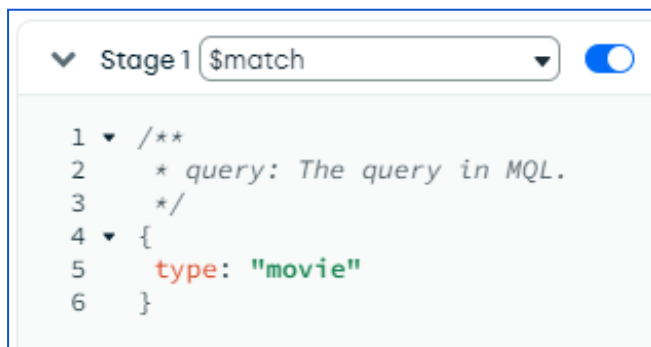


i si fem scroll els podrem veure tots.

1.3. 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.

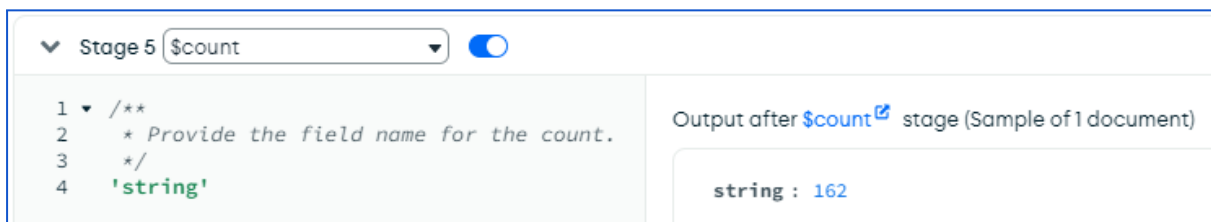
Tal i com venint fent, si volem veure'ls a través d'**Aggregations** → **Stage** haurem d'aplicar diversos filtres:



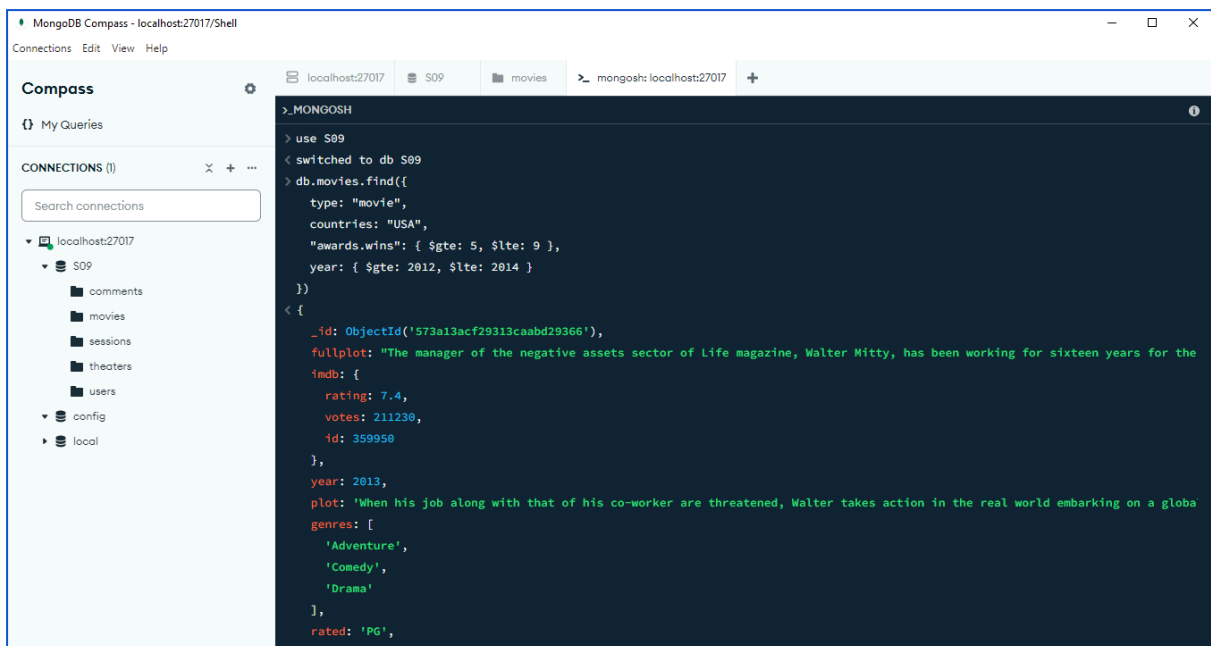


i si fem scroll a la dreta els podrem veure tots.

En total són 162. L'exercici no ho demana però està bé saber quantes són.



Per veure els documents a través de la **shell** haurem de demanar-ho així:



Si per un casual volguessim veure els títols, hauríem de fer-ho així:

```
> db.movies.find(
  {
    type: "movie",
    countries: "USA",
    "awards.wins": { $gte: 5, $lte: 9 },
    year: { $gte: 2012, $lte: 2014 }
  },
  {
    title: 1
  })
< {
  _id: ObjectId('573a13acf29313caabd29366'),
  title: 'The Secret Life of Walter Mitty'
}
{
  _id: ObjectId('573a13b5f29313caabd45772'),
  title: 'The Croods'
}
{
  _id: ObjectId('573a13b9f29313caabd4dd15'),
  title: 'The Book Thief'
}
```

```
{
  _id: ObjectId('573a13b9f29313caabd4de39'),
  title: 'World War Z'
}
{
  _id: ObjectId('573a13b9f29313caabd4f21f'),
  title: 'Godzilla'
}
{
  _id: ObjectId('573a13bbf29313caabd5271d'),
  title: 'Mr Hockey: The Gordie Howe Story'
}
{
  _id: ObjectId('573a13bcf29313caabd57341'),
  title: 'Hitchcock'
}
{
  _id: ObjectId('573a13bef29313caabd5d9c2'),
  title: 'Lone Survivor'
}
```

No es demana, però pot resultar útil.

☆☆ Nivell 2

2.1. Exercici 1

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

Desde la col·lecció **comments** de la nostra base de dades, cal utilitzar la funció **find** amb una expressió regular (**\$regex**) per buscar tots els documents on el camp **email** acaba amb "@gameofthron.es". L'expressió regular **/@gameofthron\.es\$/i** assegura que el domini sigui insensible a majúscules i minúscules (case-insensitive). Finalment, hem utilitzat la funció **count** per comptar el nombre de documents que compleixen aquesta condició.

```
>_MONGOSH
> use S09
< switched to db S09
> db.comments.find({
  email: { $regex: /@gameofthron\.es$/i }
}).count()
< 22841
```

El resultat és que aquest usuari ha escrit **22.841** comentaris.

2.2. Exercici 2

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

Des de la **shell**:

\$match: Aquesta etapa filtra els documents per assegurar-se que només es consideren els cinemes situats a Washington D.C. ("**location.address.state**": "DC").

\$group: Aquesta etapa agrupa els documents per codi postal (**_id**: "**location.address.zipcode**") i compta quants documents hi ha en cada grup utilitzant **\$sum: 1**, que incrementa el comptador per cada document que compleix la condició.

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

Per tant hi ha 1 cinema a cada codi postal de Washington DC. En total 3 a tot l'estat.

☆☆☆ Nivell 3

3.1. Exercici 1

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

Per trobar les pel·lícules que ha rodat en John Landis haurem de filtrar per **directors**, i per veure la puntuació a IMDb haurem de filtrar per **imdb.rating**.

Per altra banda demanem que es vegin els títols per consola (**title: 1**) però NO els IDs (**_id: 0**).

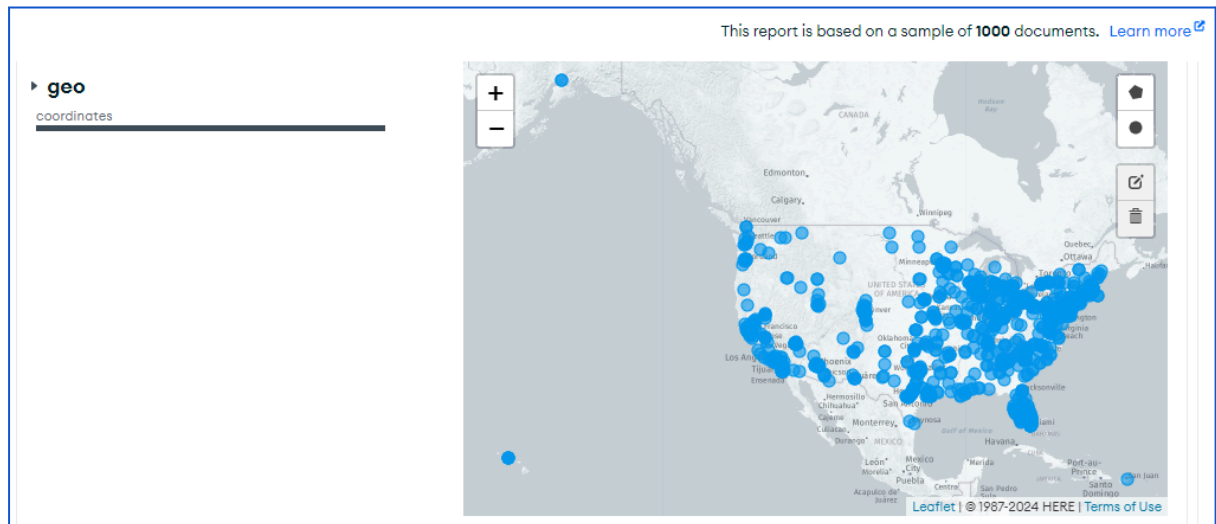
```
> db["movies"].find
(
  {
    directors: "John Landis",
    "imdb.rating": { $gte: 7.5, $lte: 8 }
  },
  { title: 1, _id: 0 }
)
< {
  title: 'Animal House'
}
{
  title: 'The Blues Brothers'
}
{
  title: 'An American Werewolf in London'
}
{
  title: 'Trading Places'
}
```

Així veiem que en John Landis té 4 pel·lícules en aquesta base de dades.

3.2. Exercici 2

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

Per veure la ubicació dels teatres en un mapa només cal estar sobre la *col·lecció theaters*, prémer sobre l'opció de **Schema**, obrir la opció de **location**, i després la de **geo**:



Tot i que el mapa només mostra 1000 registres i la taula theaters en té 1564.