

SPRINT5

SPRINT 5 : Bases de dades no relacionals

Elaborat per:

Vanesa Pérez Ramírez

Data: 15/10/2025

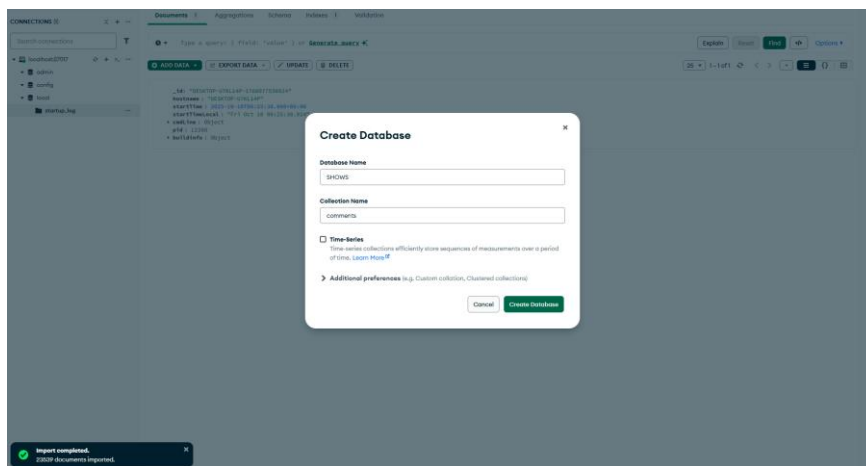
SPRINT 5 : Bases de dades no relacionals	1
Nivell 1	3
Exercici 1	5
Mostra els 2 primers comentaris que hi ha en la base de dades.	5
COMPASS:	5
MONGOSH.....	6
Quants usuaris tenim registrats?	7
A COMPASS:	7
MONGOSH.....	8
Quants cinemes hi ha en l'estat de Califòrnia?	8
COMPASS.....	8
MONGOSH.....	10
Quin va ser el primer usuari/ària en registrar-se?	11
COMPASS.....	11
Quantes pel·lícules de comèdia hi ha en la nostra base de dades?	12
COMPASS.....	12
MONGOSH.....	14
Exercici 2	14
COMPASS.....	14
MONGOSH.....	15
Exercici 3	17
COMPASS.....	17
MONGOSH.....	18
Nivell 2	19
Exercici 1	19
COMPASS.....	19
Exercici 2	21
COMPASS.....	21
Nivell 3	24
Exercici 1	24
COMPASS.....	24
Exercici 2	24
COMPASS.....	24

Nivell 1

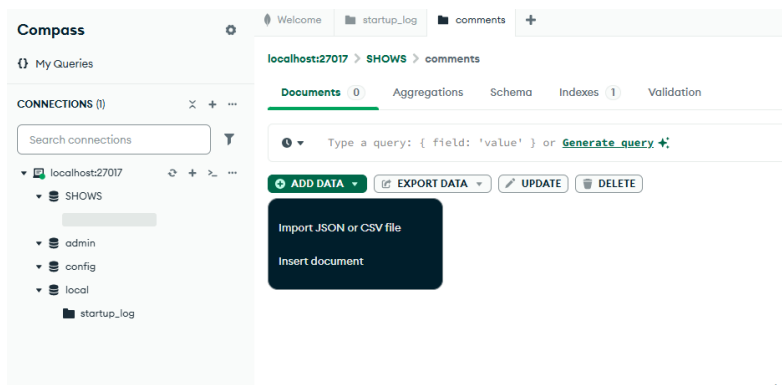
Crea una base de dades amb MongoDB utilitzant com a col·leccions els arxius adjunts.

Amb Compass:

Per crear la base de dades, clicko el símbol + que hi ha al costat de Localhost i he d'afegir el nom de la Base de dades i el nom de la primera col·lecció

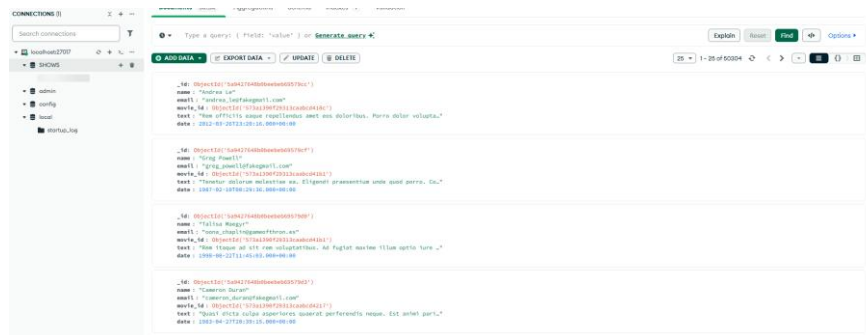


Carrego les dades de la col·lecció comments, desplegant el botó de ADD DATA i seleccionant import JSON i selecciono l'arxiu de comments, dono a acceptar i les dades estan carregades

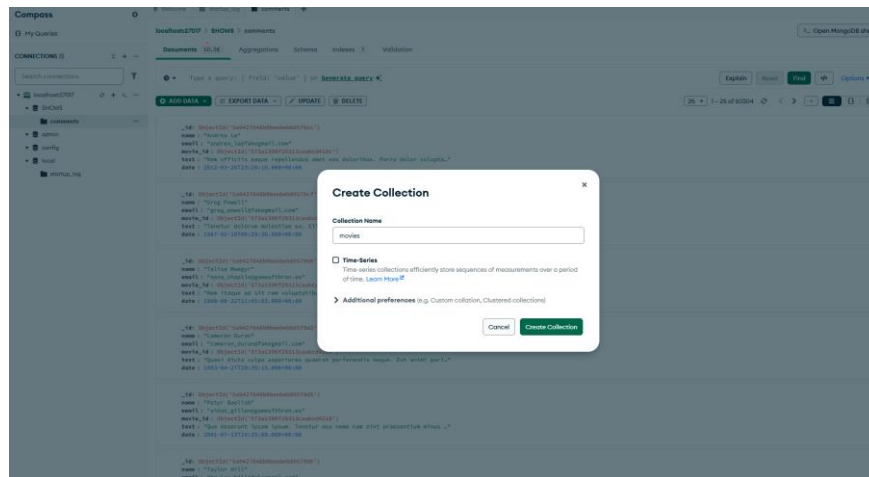


S'han carregat 50304 documents a la col·lecció

SPRINT5

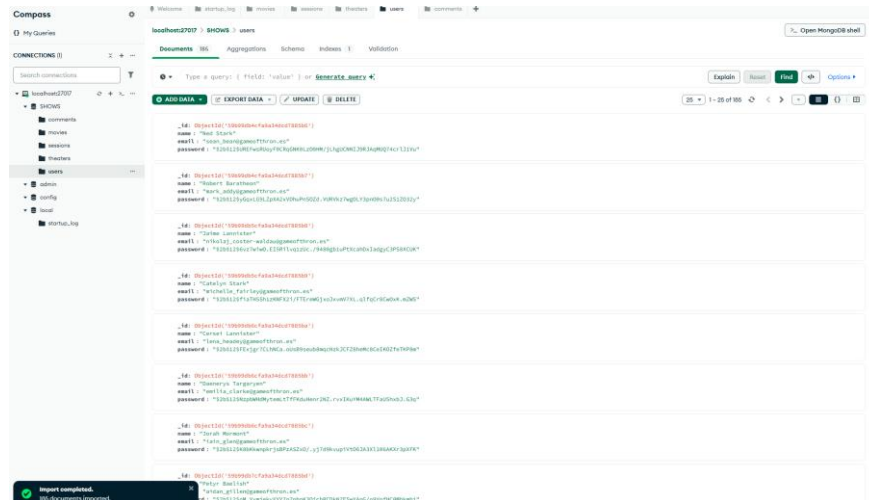


Faig el mateix a cada arxiu, però en aquests casos com la base de dades ja està creada per afegir col·leccions noves he de clicar al símbol + que hi ha al costat de la base de dades i afegir la nova col·lecció:



Ja estàn les col·leccions carregades i tots els seus documents:

SPRINT5

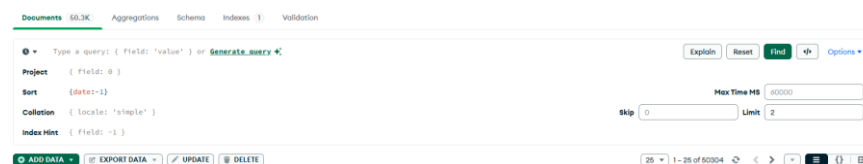


Exercici 1

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

COMPASS:

Seleccionem la col·lecció posant-nos sobre el nom de la col·lecció sobre la que volem treballar i després anem al costat de la casella del Find clickem sobre options i per fer la busqueda utilitzem el mètode sort({date:-1}) indicant que ordeni el camp date de menor a major i a més a la casella límit li posem 2, ja que només volem veure els dos primers registres:



Clickem sobre el botó Find i el resultat és:

SPRINT5

Type a query: { field: "value" } or { field: "value" : value }

Project { field: 0 }

Sort { date: -1 }

Collation { locale: 'simple' }

Index Hint { field: -1 }

Explains

Reset

Find

Options

Max Time MS 60000

Skip 0 Limit 2

25

1 - 2 of 2

<

>

⌂

⌕

⌕

ADD DATA

EXPORT DATA

UPDATE

DELETE

Introducing insights

Across Comps, you may now see icons like this to clue you in on potential areas of improvement for your data.

See insights in action

_id: ObjectId('5b7327c34a68c3ffa7bc1d3')

name: 'xkybrzdb'

email: 'v4z4prjyhgq22b.lyk'

movie_id: ObjectId('573a13eff29313caabdd82f3')

text: 'fazzlabfzle'

date: 2018-08-14T15:05:03.142+00:00

_id: ObjectId('5b7327724a68c3ffa7bc1ae')

name: 'jjjdownw'

email: 'r09xxa9h0gm3q7.7yh'

movie_id: ObjectId('573a13eff29313caabdd82f3')

text: 'fewfee'

date: 2018-08-14T15:03:14.306+00:00

Type a query: { field: "value" } or { field: "value" : value }

Project { field: 0 }

Sort { date: -1 }

Collation { locale: 'simple' }

Index Hint { field: -1 }

Explains

Reset

Find

Options

Max Time MS 60000

Skip 0 Limit 2

25

1 - 2 of 2

<

>

⌂

⌕

⌕

ADD DATA

EXPORT DATA

UPDATE

DELETE

Introducing insights

Across Comps, you may now see icons like this to clue you in on potential areas of improvement for your data.

See insights in action

_id: ObjectId('5b7327c34a68c3ffa7bc1d3')

name: 'xkybrzdb'

email: 'v4z4prjyhgq22b.lyk'

movie_id: ObjectId('573a13eff29313caabdd82f3')

text: 'fazzlabfzle'

date: 2018-08-14T15:05:03.142+00:00

_id: ObjectId('5b7327724a68c3ffa7bc1ae')

name: 'jjjdownw'

email: 'r09xxa9h0gm3q7.7yh'

movie_id: ObjectId('573a13eff29313caabdd82f3')

text: 'fewfee'

date: 2018-08-14T15:03:14.306+00:00

MONGOSH

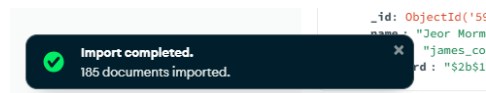
```
> MONGOOSH
> use SHOWS
< switched to db SHOWS
> db["comments"].find().sort({date:-1}).limit(2)
< {
  _id: ObjectId('5b7327c34a68c3ffa7bc1d3'),
  name: 'xkybrzdb',
  email: 'v4z4prjyhgq22b.lyk',
  movie_id: ObjectId('573a13eff29313caabdd82f3'),
  text: 'fazzlabfzle',
  date: 2018-08-14T15:05:03.142Z
}
{
  _id: ObjectId('5b7327724a68c3ffa7bc1ae'),
  name: 'jjjdownw',
  email: 'r09xxa9h0gm3q7.7yh',
  movie_id: ObjectId('573a13eff29313caabdd82f3'),
  text: 'fewfee',
  date: 2018-08-14T15:03:14.306Z
}
SHOWS >
```

SPRINT5

Quants usuaris tenim registrats?

A COMPASS:

No caldria fer cap operació ja que el mateix programa et dona la informació del número de documents (registres)

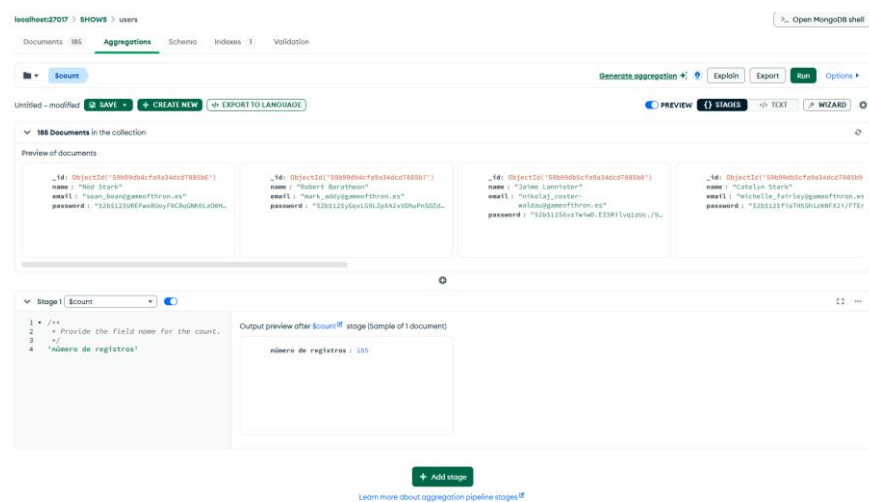


Veiem que la pestanya de documents ja indica el número 185.

Veiem que els usuaris estan indexats



I si fem la funció d'agregació count() a compass veiem que el resultat de número de documents és de 185.



SPRINT5

MONGOSH

```
> MONGOSH

> db.getCollectionNames().map( (name) => ({[name]: db[name].count()}) )
< DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
< [
  { comments: 50304 },
  { users: 185 },
  { theaters: 1564 },
  { movies: 23539 },
  { sessions: 1 }
]
SHOWS >
```

O també:

```
> db.users.countDocuments()
< 185
SHOWS >
```

Quants cinemes hi ha en l'estat de Califòrnia?

COMPASS

Primer em miro detingudament l'estructura dels documents veiem que a theaters la localització és un objecte que inclou dos objectes (address, geo), a l'objecte address veiem el parell de ciutat.

```
{
  "_id": ObjectId("59a47286cfa9a3a73e51e796"),
  "theaterId": 1120,
  "location": Object
}
```

```
{
  "_id": ObjectId("59a47286cfa9a3a73e51e797"),
  "theaterId": 1121,
  "location": Object
  ▾
    address: Object
      street1: "5101 Frederica St"
      city: "Owensboro"
      state: "KY"
      zipcode: "42301"
    geo: Object
      type: "Point"
      coordinates: Array (2)
        0: -87.127319
        1: 37.719315
  }
}
```

Per veure com es diu el State a l'arxiu fem una selecció de tots els i veiem que "CA" és California al document.

SPRINT5

Project: `{ "location.address.state": "true" }`

Sort: `{ field: -1 } or [[field, -1]]` Max Time MS:

Collection: `{ locale: 'single' }` Skip: Limit:

Index Hint: `{ field: -1 }`

EXPORT DATA 1-25 of 164

```
{
  "_id": "1",
  "location": {
    "address": {
      "state": "CA"
    }
  }
}
```

```
{
  "_id": "2",
  "location": {
    "address": {
      "state": "CA"
    }
  }
}
```

Fem el filtre per State: "CA" i ens dona els resultats i al costat de la busqueda ja ens surt el número de documents resultants del filtre que són 169:

Generate query

Project: `{ (theaterId:true) }`

Sort: `{ }` Max Time MS:

Collection: `{ locale: 'single' }` Skip: Limit:

Index Hint: `{ field: -1 }`

EXPORT DATA 1-25 of 169

```
{
  "_id": "1",
  "theaterId": 1000
}
```

```
{
  "_id": "2",
  "theaterId": 1009
}
```

```
{
  "_id": "3",
  "theaterId": 1010
}
```

```
{
  "_id": "4",
  "theaterId": 1011
}
```

Podem utilitzar funcions d'agregació:

Stage 1:

```
1 //
2 // query: The query in MQL.
3 //
4 {
5   "location.address.state": "CA"
6 }
```

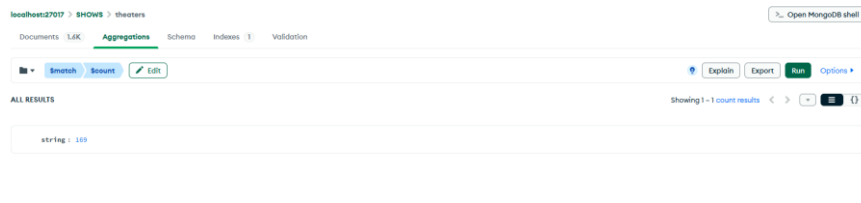
Stage 2:

```
1 //
2 // Provide the field name for the count.
3 //
4 "string"
```

[Learn more about aggregation pipeline stages](#)

Executem els stages:

SPRINT5



MONGOSH

Per veure com es diu el State a l'arxiu fem una selecció de tots els i veiem que "CA" és California a l'arxiu.

```
> _MONGOSH
> db.theaters.countDocuments( { "location.address.state": "CA" })
< 169
SHOWS >
```

```
> use SHOWS
< switched to db.SHOWS
> db["theaters"].find({}, {location:true})
< [
  {
    _id: ObjectId('59a47286cfa9a3a73e51e72c'),
    location: {
      address: {
        street1: '348 W Market',
        city: 'Bloomington',
        state: 'MI',
        zipcode: '55425'
      },
      geo: {
        type: 'Point',
        coordinates: [
          -93.24565,
          44.85466
        ]
      }
    }
  },
  {
    _id: ObjectId('59a47286cfa9a3a73e51e72d'),
    location: {
      address: {
        street1: '45235 Worth Ave.',
        city: 'California',
        state: 'MD',
        zipcode: '20619'
      },
      geo: {
        type: 'Point',
        coordinates: [
          -76.512816,
          38.29697
        ]
      }
    }
  }
]
```

```
{
  _id: ObjectId('59a47286cfa9a3a73e51e72e'),
  location: {
    address: {
      street1: '1621 E Monte Vista Ave',
      city: 'Vacaville',
      state: 'CA',
      zipcode: '95688'
    },
    geo: {
      type: 'Point',
      coordinates: [
        -121.96328,
        38.367649
      ]
    }
  }
}
```

SPRINT5

Per calcular el número de teatres que hi ha a California apliquem el següent:

```
> _MONGOSH
> db.theaters.countDocuments( { "location.address.state": "CA" })
< 169
SHOWS >
```

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

COMPASS

Com no hi ha data als documents de la col·lecció de users a l'id hi ha data per tant sol·licitarem la crear una nova columna que s'anomeni fecha_creacion y aquesta l'ordenarem en del mes antic al mes nou. Quan ho fem veiem que hi ha dos usuaris amb la mateixa data i exactament la mateixa hora, per tant decideixo fer un límit 2 per a que ens surtin el dos documents.

The screenshot shows the MongoDB Compass interface for the 'users' collection. The aggregation pipeline is defined as follows:

```
1 //
2 // newfield: the new field name,
3 // expression: the new field expression.
4 //
5 {
6   fecha_creacion: { $toDate: "$date" }
7 }
```

The output preview after the \$addFields stage shows the first 10 documents. The first two documents have the same 'fecha_creacion' value: '2017-09-13T21:05:16.800-0800'.

The second stage is \$sort, which sorts the documents by 'fecha_creacion' in ascending order. The output preview after the \$sort stage shows the first 10 documents, which are the same as the first two documents from the \$addFields stage.

SPRINT5

The screenshot shows the MongoDB Atlas pipeline editor. Stage 2 is '\$sort' and Stage 3 is '\$limit'. The output preview for Stage 2 shows 10 documents, and for Stage 3, it shows 2 documents.

```
1 //
2 // Provide any number of field/order pair
3 //
4 {
5   fecha_creacion: 1
6 }
7
```

Output preview after \$sort stage (Sample of 10 documents)

Document 1	Document 2	Document 3
<pre>{ "_id": "5909b04cfab344dcf7883b7", "name": "Robert Baratheon", "email": "robert_baratheon@fbrion.es", "password": "1205121yGxLGRzKAZxV0hPh002G...", "fecha_creacion": 2017-09-13T21:05:56.098+09:00 }</pre>	<pre>{ "_id": "5909b04cfab344dcf7883b6", "name": "Ned Stark", "email": "ned_stark@fbrion.es", "password": "1205121yGxLGRzKAZxV0hPh002G...", "fecha_creacion": 2017-09-13T21:05:56.098+09:00 }</pre>	<pre>{ "_id": "5909b04cfab344dcf7883b5", "name": "Catelyn Stark", "email": "catelyn_stark@fbrion.es", "password": "1205121yGxLGRzKAZxV0hPh002G...", "fecha_creacion": 2017-09-13T21:05:56.098+09:00 }</pre>

Output preview after \$limit stage (Sample of 2 documents)

Document 1	Document 2
<pre>{ "_id": "5909b04cfab344dcf7883b6", "name": "Ned Stark", "email": "ned_stark@fbrion.es", "password": "1205121yGxLGRzKAZxV0hPh002G...", "fecha_creacion": 2017-09-13T21:05:56.098+09:00 }</pre>	<pre>{ "_id": "5909b04cfab344dcf7883b7", "name": "Robert Baratheon", "email": "robert_baratheon@fbrion.es", "password": "1205121yGxLGRzKAZxV0hPh002G...", "fecha_creacion": 2017-09-13T21:05:56.098+09:00 }</pre>

[Learn more about aggregation pipeline stages](#)

The screenshot shows the MongoDB Atlas aggregation results. The aggregation pipeline consists of \$sort and \$limit stages. The results show two documents.

```
1 //
2 // Provide the number of documents to limit
3 //
4 2
```

Output preview after \$limit stage (Sample of 2 documents)

Document 1	Document 2
<pre>{ "_id": "5909b04cfab344dcf7883b6", "name": "Ned Stark", "email": "ned_stark@fbrion.es", "password": "1205121yGxLGRzKAZxV0hPh002G...", "fecha_creacion": 2017-09-13T21:05:56.098+09:00 }</pre>	<pre>{ "_id": "5909b04cfab344dcf7883b7", "name": "Robert Baratheon", "email": "robert_baratheon@fbrion.es", "password": "1205121yGxLGRzKAZxV0hPh002G...", "fecha_creacion": 2017-09-13T21:05:56.098+09:00 }</pre>

[Learn more about aggregation pipeline stages](#)

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

COMPASS

S'aplica el filtre fent un \$and:

Directament diu que hi han 7002 registres:

SPRINT5

Documents 23.6K Aggregations Schema Indexes Validation

Generate query [Explain](#) [Reset](#) [Find](#) [Options](#)

Project (Field: 0)

Sort (Field: -1) or [{"field": -1}]

Collection (local: 'sample')

Index Hint (Field: -1)

Max Time MS 60000

Skip 0 Limit 0

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1-26 of 7002

```
{
  "_id": "ObjectID('973a1390f29313caab04883')",
  "plot": "Cartoon figures announce, via comic strip balloons, that they will mov...",
  "genres": Array (3),
  "runtime": 7,
  "cast": Array (1),
  "num_offits_comments": 1,
  "poster": "https://m.media-amazon.com/images/M/W58Yzg2NlNhNjc0Sj0wM180ZWM4MzI3-...",
  "title": "Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving...",
  "fullplot": "Cartoonist Winsor McCay agrees to create a large set of drawings that...",
  "languages": Array (1),
  "released": "1911-04-04T04:00:00-08:00-00:00",
  "directors": Array (2),
  "writers": Array (2),
  "moods": Object,
  "lastupdated": "2015-08-29 01:09:03.030000000",
  "year": 1911,
  "tmdb": Object,
  "countries": Array (1),
  "type": "movie",
  "tomatoes": Object
}
```

```
{
  "_id": "ObjectID('973a1390f29313caab0d065')",
  "plot": "The cartoonist, Winsor McCay, brings the Dinosaur back to life in th...",
  "genres": Array (3),
  "runtime": 12,
  "cast": Array (1),
  "num_offits_comments": 1,
  "poster": "https://m.media-amazon.com/images/M/W58Yzg2NlNhNjc0Sj0wM180ZWM4MzI3-...",
  "title": "Gertie the Dinosaur",
  "fullplot": "Winsor L. McCay bets another cartoonist that he can animate a dinosaur...",
  "languages": Array (1),
  "released": "1914-09-12T04:00:00-08:00-00:00",
  "directors": Array (1),
  "writers": Array (1),
  "moods": Object,
  "lastupdated": "2015-08-18 01:03:15.313000000",
  "year": 1914,
  "tmdb": Object,
  "countries": Array (1),
  "type": "movie"
}
```

Per agregació:

Les dues passes:

Documents 23.6K Aggregations Schema Indexes Validation

[Match](#) [Count](#) [Generate aggregation](#) [Explain](#) [Export](#) [Run](#) [Options](#)

Unified - modified [SAVE](#) [CREATE NEW](#) [EXPORT TO LANGUAGE](#) [PREVIEW](#) [STAGES](#) [TEXT](#) [WIZARD](#)

Stage 1 [Match](#)

```
1 //
2 // query: The query in RQL.
3 //
4 {
5   $and: [{genres:"Comedy"}], {type:"movie"}
6 }
```

Output preview after [Match](#) stage (Sample of 10 documents)

```
{
  "_id": "ObjectID('973a1390f29313caab04883')",
  "plot": "Cartoon figures announce, via comic strip balloons, that they will mov...",
  "genres": Array (3),
  "runtime": 7,
  "cast": Array (1),
  "num_offits_comments": 1,
  "poster": "https://m.media-amazon.com/images/M/W58Yzg2NlNhNjc0Sj0wM180ZWM4MzI3-...",
  "title": "Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving...",
  "fullplot": "Cartoonist Winsor McCay agrees to create a large set of drawings that...",
  "languages": Array (1),
  "released": "1911-04-04T04:00:00-08:00-00:00",
  "directors": Array (2),
  "writers": Array (2),
  "moods": Object,
  "lastupdated": "2015-08-29 01:09:03.030000000",
  "year": 1911,
  "tmdb": Object,
  "countries": Array (1),
  "type": "movie",
  "tomatoes": Object
}
```

Stage 2 [Count](#)

```
1 //
2 // Provide the field name for the count.
3 //
4 'numero de películas que són comedies'
```

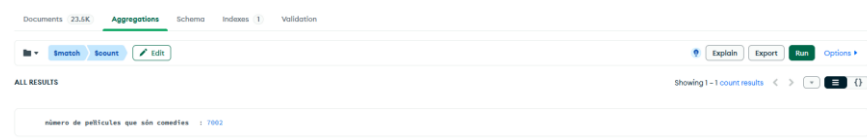
Output preview after [Count](#) stage (Sample of 1 document)

```
{
  "numero de películas que són comedies": 7002
}
```

[Add stage](#) [Learn more about aggregation pipeline stages](#)

El resultat un cop executada l'agregació:

SPRINT5



MONGOSH

Apliquem el filtre perquè sigui pel·lícula i en la descripció del gènere aparegui la paraula comedy:

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

Exercici 2

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

COMPASS

Apliquem el següent filtre:

```
{ $and: [{ $or: [{ genres: "Drama" }, { languages: "French" } ] }, { type: "movie" }, { year: 1932 } ] }
```

I el resultat és el següent:

SPRINT5

na" }, {languages:"French"}]], {type:"movie"}, {year:1932}]]

Generate query ↗

Explain

Reset

Find

Options ▶

ADD DATA

EXPORT DATA

UPDATE

DELETE

1 - 18 of 18

```
_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_flix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWVjYmY1ODNjA4LTk4ZW..."
title: "The Blood of a Poet"
lastupdated: "2015-09-16 13:13:05.537000000"
languages: Array (1)
released: 2010-05-20T00:00:00.000+00:00
directors: Array (1)
writers: Array (1)
awards: Object
year: 1932
imdb: Object
countries: Array (1)
type: "movie"
tomatoes: Object
```

```
_id: ObjectId('573a1392f29313caabcd99a3')
plot: "Junta is hated by the people in the village where she lives, especiall..."
genres: Array (3)
```

18 documents

MONGOSH

Aplico el mètode find amb el filtre or per les condicions llenguatge en Francés o sigui un drama. Y aplico el filtre perquè compleixi que està dintre de l'or i amés a més sigui movie i a més a més sigui del 1932

L'expressió per l'or és:

```
{ $or: [{ genres: "Drama" }, { languages: "French" } ] }
```

I l'expressió per l'and és:

```
{ $and: [{ $or: [{ genres: "Drama" }, { languages: "French" } ] }, { type: "movie" }, { year: 1932 } ] }
```

Al mongosh queda de la següent forma:

SPRINT5

```
> db["movies"].find({$and:[{$or:[{genres:"Drama" },{languages:"French"}]},{type:"movie"},{year:1932}}})
< {
  _id: ObjectId('573a1391f29313caabcd9450'),
  plot: 'A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing comes into life and starts t
  runtime: 55,
  rated: 'UNRATED',
  cast: [
    'Enrique Rivero',
    'Elizabeth Lee Miller',
    'Pauline Carton',
    'Odette Talazac'
  ],
  num_mflix_comments: 1,
  poster: 'https://m.media-amazon.com/images/M/MV5BYWY3ODk5ZWEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjV0VTVxXkEyXkFqcGdeQXVyNzI4MDMyMT
  title: 'The Blood of a Poet',
  lastupdated: '2015-09-16 13:13:05.537000000',
  languages: [
    'French'
  ],
  released: 2010-05-20T00:00:00.000Z,
  directors: [
    'Jean Cocteau'
  ],
  writers: [
    'Jean Cocteau'
  ]
}
```

El número de documents que s'han filtrat són:

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


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.

COMPASS

Li aplico el següent filtre:

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

Utilitzo l'operador lògic AND de les condicions amb operadors de comparació que ens han indicat:

País Estats Units: {countries:"USA"}

Premis entre 5 i 9: {"awards.wins":{\$gte:5,\$lte:9}}

Any entre 2012 i 2014: {year:{\$gte:2012,\$lte:2014}}

Quedant un llistat de 166 documents:

The screenshot shows the Compass web interface for MongoDB. At the top, a query is entered in the search bar: `{&and:[{countries:"USA"},{"awards.wins":{$gte:5,$lte:9}},{year:{$gte:2012,$lte:2014}}]}`. Below the search bar, there are buttons for 'Explain', 'Reset', 'Find', and 'Options'. The 'Find' button is highlighted in green. Below the search bar, there are fields for 'Project', 'Sort', 'Collation', and 'Index Hint'. The 'Project' field is set to `{ field: 0 }`. The 'Sort' field is set to `{ field: -1 } or [['field', -1]]`. The 'Collation' field is set to `{}`. The 'Index Hint' field is set to `{ field: -1 }`. Below these fields, there are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The 'EXPORT DATA' button is highlighted in green. Below the buttons, there is a table of results. The first result is a document with the following fields: `_id`, `fullplot`, `imdb`, `year`, `plot`, `genres`, `rated`, `metacritic`, `title`, `lastupdated`, `languages`, `writers`, `type`, `tomatoes`, `poster`, and `num_mflix_comments`. The document is for the movie 'The Secret Life of Walter Mitty'.

```
{
  "_id": "ObjectId('573a13acf29313caabd29366')",
  "fullplot": "The manager of the negative assets sector of Life magazine, Walter Mit...",
  "imdb": {
    "year": 2013,
    "plot": "When his job along with that of his co-worker are threatened, Walter t...",
    "genres": [
      "PG"
    ],
    "rated": "PG",
    "metacritic": 54,
    "title": "The Secret Life of Walter Mitty",
    "lastupdated": "2015-08-31 00:10:51.747000000"
  },
  "languages": [
    "English"
  ],
  "writers": [
    "Tom Mankiewicz"
  ],
  "type": "movie",
  "tomatoes": {
    "poster": "https://m.media-amazon.com/images/M/MV5BODYwNDYxNDk1N15BM15BanBnXkFtZT...",
    "num_mflix_comments": 1
  }
}
```

SPRINT5

MONGOSH

Aplicant la mateixa lògica que amb compass queda de la següent manera:

```
>_MONGOSH
> db.movies.find({$and:[{countries:"USA"},{"awards.wins":{$gte:5,$lte:9}},{year:{$gte:2012,$lte:2014}}])
< {
  _id: ObjectId('573a13acf29313caabd29366'),
  fullplot: "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen years for
  imdb: {
    rating: 7.4,
    votes: 211230,
    id: 359950
  },
  year: 2013,
  plot: 'When his job along with that of his co-worker are threatened, Walter takes action in the real world embarking on a g
  genres: [
    'Adventure',
    'Comedy',
    'Drama'
  ],
  rated: 'PG',
  metacritic: 54,
  title: 'The Secret Life of Walter Mitty',
  lastupdated: '2015-08-31 00:10:51.747000000',
  languages: [
    'English',
    'Spanish',
    'Icelandic'
  ]
}
```

Mirem que el número de documents sigui 166:

```
> db.movies.countDocuments({$and:[{countries:"USA"},{"awards.wins":{$gte:5,$lte:9}},{year:{$gte:2012,$lte:2014}}])
< 166
SHOWS>
```

Nivell 2

Exercici 1

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

COMPASS

El que he decidit fer es buscar el patró del domini desitjat al email amb la funció \$regex i dient-li que busqui tant majúscules com minúscules. Surten un total de comentaris de 22841 entre tots el usuaris amb aquest domini de mail.

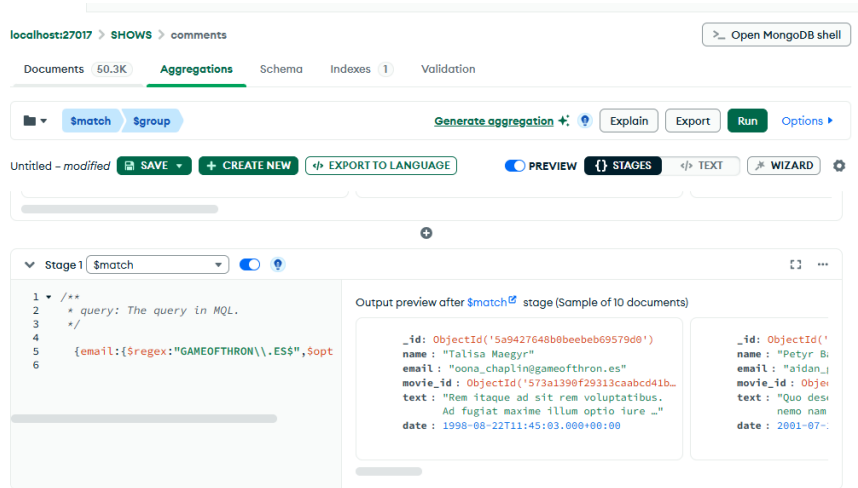
Nota: les contrabarras abans del punt són per que vegi el punt com string i el dolar final es per dir que el text termina allà.

The screenshot shows the MongoDB Compass interface. At the top, the connection is 'localhost:27017' and the database is 'SHOWS'. The collection 'comments' is selected. The search bar contains the query: `{email:{$regex:"GAMEOFTHRON\\.ES$",options:"i"}}`. Below the search bar, there are buttons for 'Generate query', 'Explain', 'Reset', 'Find', and 'Options'. The results are displayed in a list of documents. The first three documents are shown, each with fields: _id, name, email, movie_id, text, and date. The email addresses in the first three documents are: 'oona_chaplin@gameofthron.es', 'Petyr Baelish', and 'brenock_o'connor@gameofthron.es'.

_id	name	email	movie_id	text	date
ObjectId('5a9427648b0beebe69579d0')	Talisa Maegyr	oona_chaplin@gameofthron.es	ObjectId('573a1390f29313caabed41b1')	Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ..	1998-08-22T11:45:03.000+00:00
ObjectId('5a9427648b0beebe69579d5')	Petyr Baelish	saidon.gillen@gameofthron.es	ObjectId('573a1390f29313caabed4218')	Quo deserunt ipsam ipsum. Tenetur eos nemo nam sint praesentium minus ..	2001-07-13T19:25:09.000+00:00
ObjectId('5a9427648b0beebe69579db')	Ollivander	brenock_o'connor@gameofthron.es	ObjectId('573a1390f29313caabed413b')	Perspicatissimam sit pariatur quas. Perferendis officia harum ipsum deleniti ..	2005-01-04T13:49:05.000+00:00

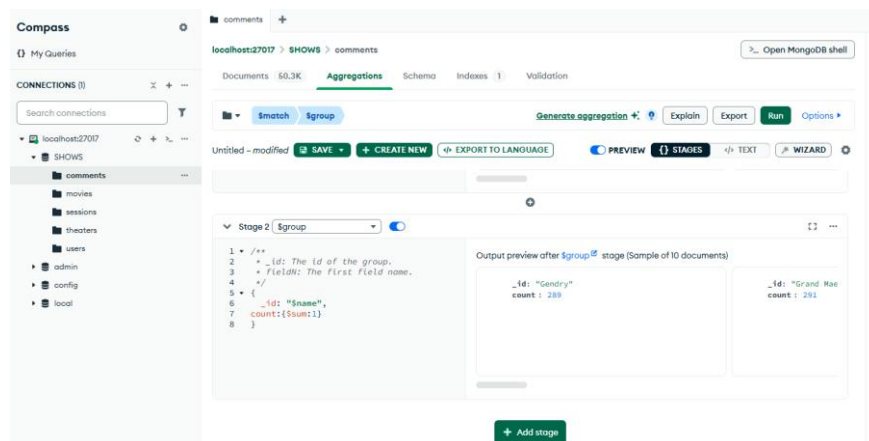
Group by para cada usuario:

SPRINT5



The screenshot shows the MongoDB Compass interface with the 'Aggregations' tab selected. The database is 'localhost:27017' and the collection is 'SHOWS'. The 'Aggregations' tab shows a single stage, '\$match', with a query in MQL: `{email:{$regex:"GAMEOFTHRON\\..ES$"},$opt`. The output preview shows a sample of 10 documents, including details for 'Talisa Maegyr' and 'Petyr B. email: "Petyr B. email: "aidan_j movie_id: ObjectId("573a1396f29313caabcd41b_ text: "Quo des nemo nam date: 2091-07-".

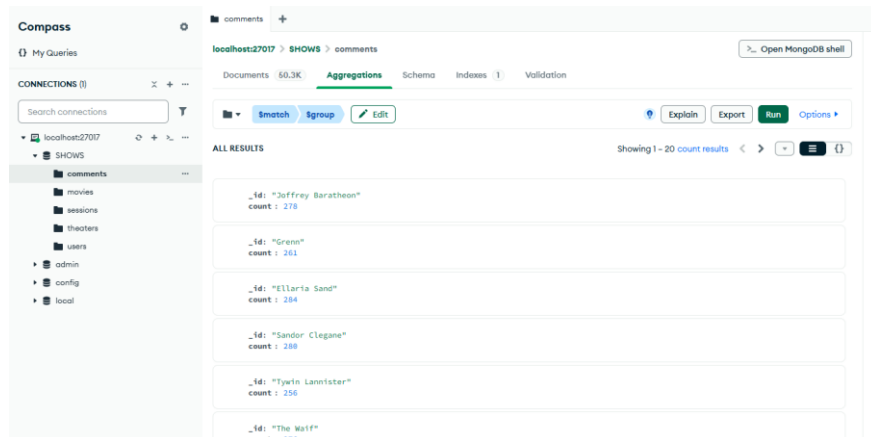
Stage 2:



The screenshot shows the MongoDB Compass interface with the 'Aggregations' tab selected. The database is 'localhost:27017' and the collection is 'SHOWS'. The 'Aggregations' tab shows two stages: '\$match' and '\$group'. The '\$group' stage has a query in MQL: `{_id: "$name", count: {$sum:1}}`. The output preview shows a sample of 10 documents, including details for 'Gendry' and 'Grand Ma count: 291'.

Run Aggregation: ens dona de cada usuari amb el domini quants comentaris ha fet.

SPRINT5



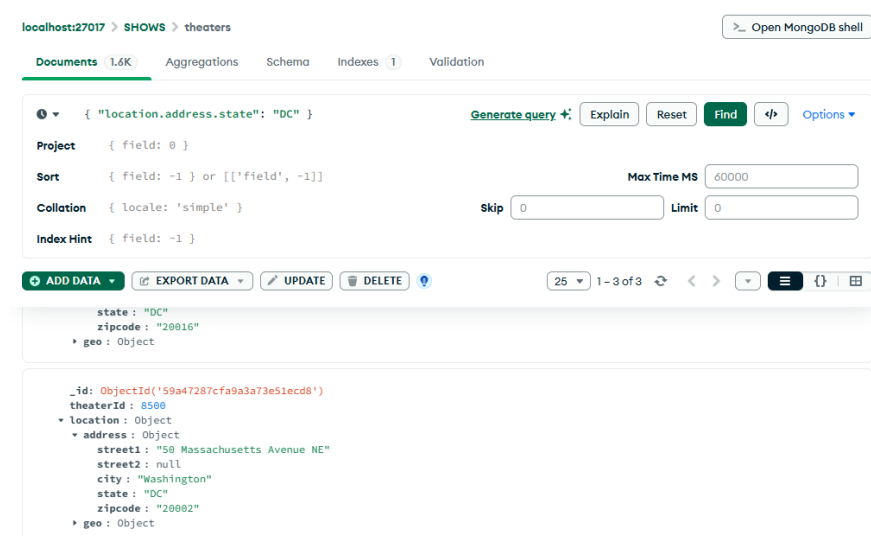
The screenshot shows the MongoDB Compass interface. On the left, the 'SHOWS' collection is expanded, showing sub-collections like 'comments', 'movies', 'sessions', 'theaters', and 'users'. The main panel displays an aggregation pipeline for the 'comments' collection. The pipeline consists of a single stage: `{ "_id": "Joffrey Baratheon" }`. The results show the count for each character ID.

Character ID	Count
"Joffrey Baratheon"	278
"Crane"	261
"Ellaria Sand"	284
"Sandor Clegane"	288
"Tywin Lannister"	256
"The Wolf"	276

Exercici 2

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

COMPASS



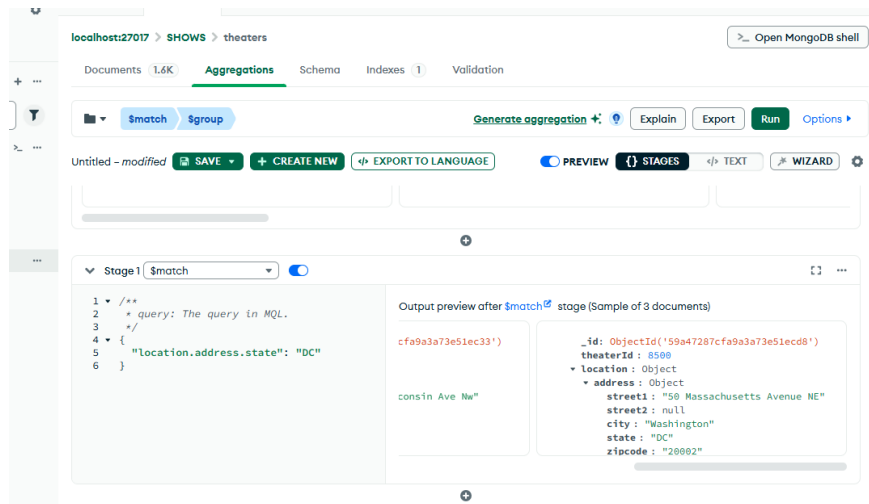
The screenshot shows the MongoDB Compass interface. The main panel displays a query for the 'theaters' collection. The query is: `{ "location.address.state": "DC" }`. The results show the count for each state code.

State Code	Count
"DC"	20016

Fem l'agregació:

Stage 1: filtrem per DC amb un match

SPRINT5



localhost:27017 > SHOWS > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

Generate aggregation + Explain Export Run Options

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

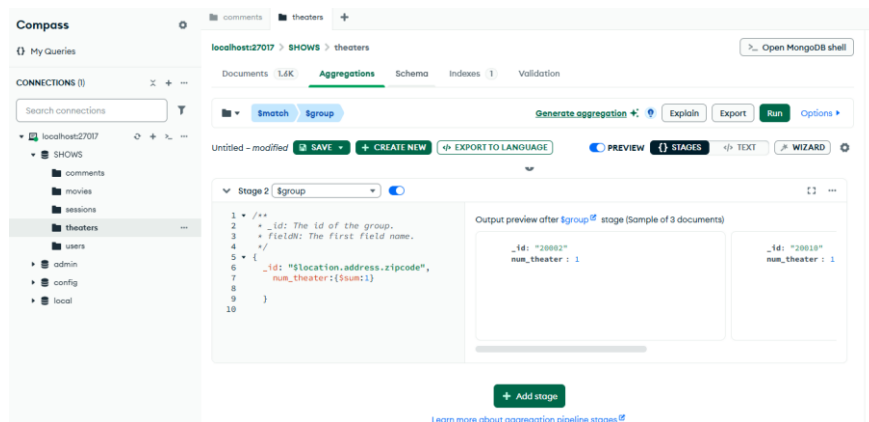
Stage 1 \$match

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "location.address.state": "DC"
6 }
```

Output preview after \$match stage (Sample of 3 documents)

```
{
  "_id": ObjectId("59a47287cfa9a3a73e51ecd8"),
  "theaterId": 8598,
  "location": {
    "address": {
      "street1": "58 Massachusetts Avenue NE",
      "street2": null,
      "city": "Washington",
      "state": "DC",
      "zipcode": "20002"
    }
  }
}
```

Fem el count , comptant els objectes:



localhost:27017 > SHOWS > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

Generate aggregation + Explain Export Run Options

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 2 \$group

```
1 /**
2  * _id: The id of the group.
3  * $first: The first field name.
4  */
5 {
6   "_id": "$location.address.zipcode",
7   "num_theater": {$sum:1}
8 }
9
10
```

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

```
{
  "_id": "20002",
  "num_theater": 1
}
```

+ Add stage

[Learn more about aggregation pipeline stages](#)

Ho executem i obtenim el desitjat:

SPRINT5

localhost:27017 > SHOWS > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

[-] \$match \$group Edit Explain Export Run Options ▾

ALL RESULTS Showing 1 – 3 count results < > [Menu] [JSON]

<pre>{ "_id": "20016", "num_theater": 1 }</pre>
<pre>{ "_id": "20010", "num_theater": 1 }</pre>
<pre>{ "_id": "20002", "num_theater": 1 }</pre>

Comentado [VP1]:

Nivell 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.

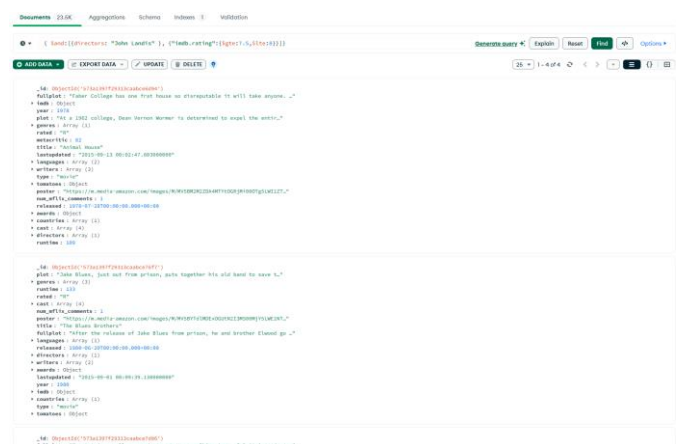
COMPASS

Faig un find amb la següent expressió:

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

Fent que compleixi les condicions amb un \$and i una de les condicions és el que en SQL seria un between.

La pantalla mostra els resultat e indica que el número de documents és 4.



Exercici 2

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

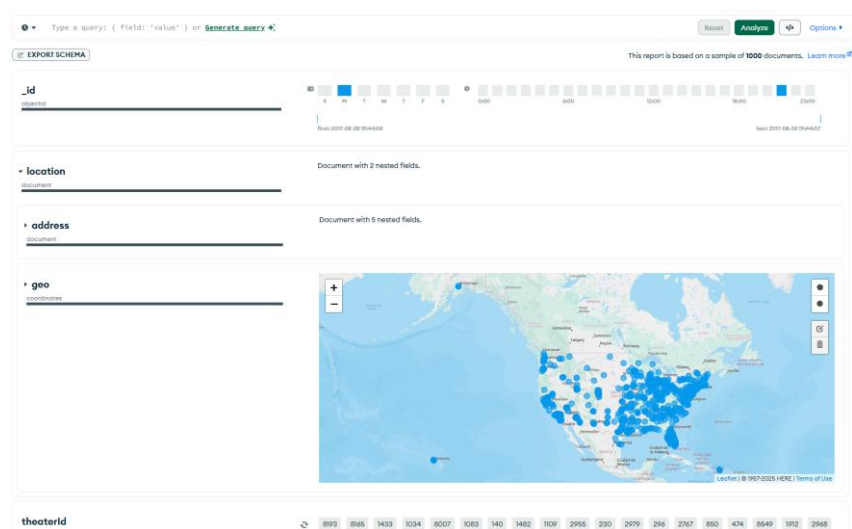
COMPASS

Per veure les ubicacions en el mapa com dintre dels documents de cada teatre hi ha les coordenades de les localitzacions. Anem a SCHEMA de la col·lecció i:

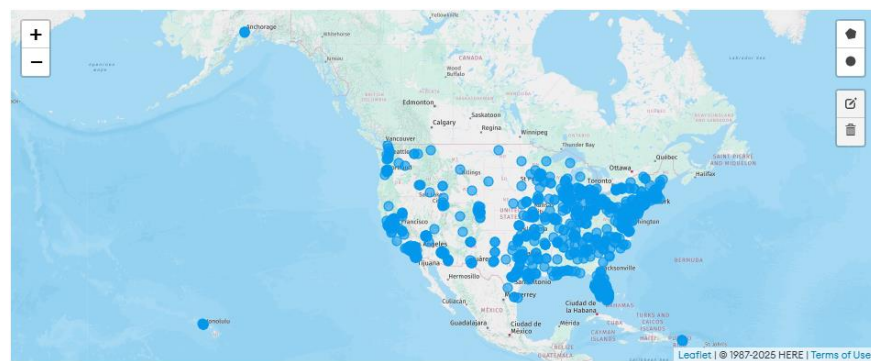
SPRINT5



A l'apartat de localització podem veure la representació gràfica



Retallo la foto



SPRINT5

