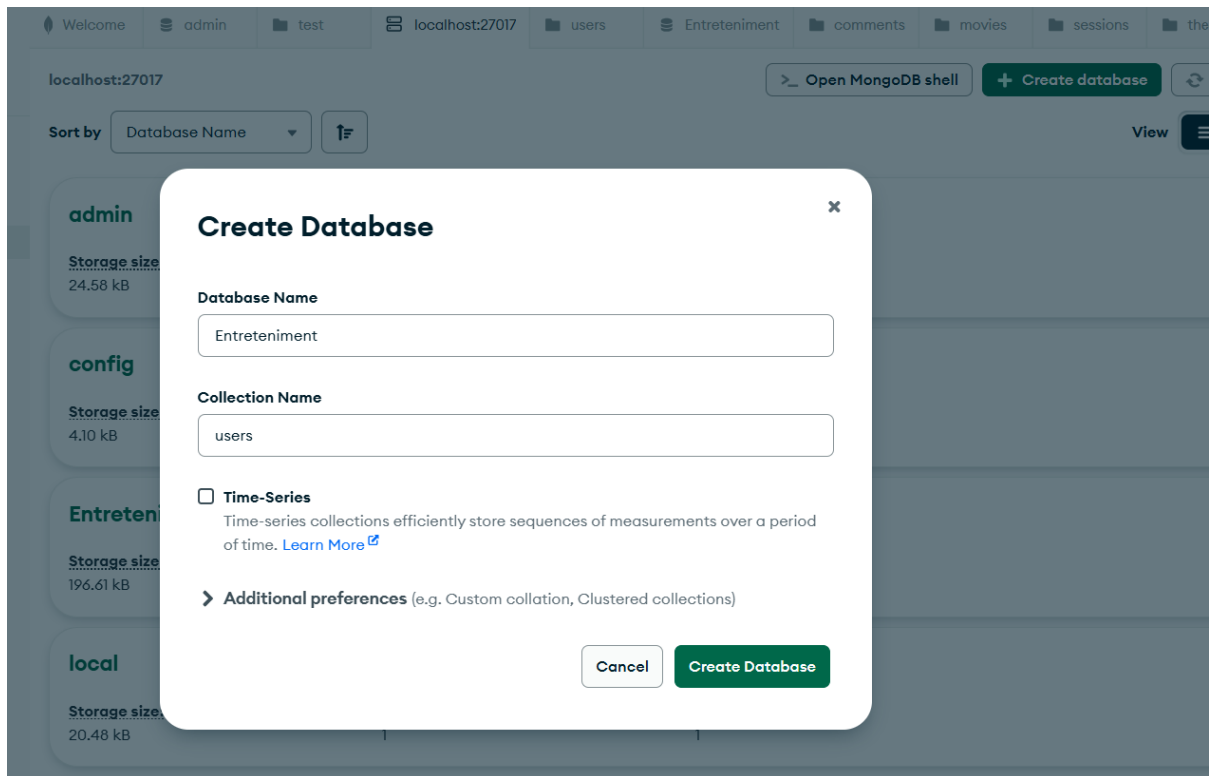


Nivell 1

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

Creemos una nueva Base de Datos, a la que le damos por nombre Entreteniment y generamos en primer lugar la colección de users



Una vez creada, vamos a la colección y añadimos los datos importando el fichero users.json

localhost:27017 > **Entertainment** > **users** Open MongoDB shell

Documents 185 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find </> Options

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - 25 of 185 ↺ ↻ ⋮ {} ⌂

```

_id: ObjectId('59b99db4cfa9a34dcd7885b6')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu"

_id: ObjectId('59b99db4cfa9a34dcd7885b7')
name: "Robert Baratheon"
email: "mark_addy@gameofthron.es"
password: "$2b$12$yGqxLG9LZpXA2xVDhuPnSOZd.VURVxz7wg0LY3pn00s7u2S1Z032y"

_id: ObjectId('59b99db5cfa9a34dcd7885b8')
name: "Jaime Lannister"
email: "nikolaj_coster-waldau@gameofthron.es"
password: "$2b$12$6vz7wiw0.EI5Rilvq1zUc./9480gb1uPtXcahDxIadgyC3PS8XCUK"

_id: ObjectId('59b99db5cfa9a34dcd7885b9')
name: "Catelyn Stark"
email: "michelle_fairley@gameofthron.es"
password: "$2b$12$fiTH5Sh1zKNFX2i/FTereWGjxoJxvmV7XL.q1fqCr8Cw0xK.mZWS"

```

Y repetimos el proceso de crear colección y cargar datos para el resto de colecciones (users, theatres, sessions, movies, comments). Obteniendo así la base de datos con todas las colecciones.

MongoDB Compass - localhost:27017/Entertainment

Connections Edit View Help

Compass My Queries

Connections (1) Search connections

localhost:27017

- Entertainment
 - comments
 - movies
 - sessions
 - theatres
 - users
- admin
 - test
- config
- local

localhost:27017 > **Entertainment** Open MongoDB shell Create collection Refresh

Sort by Collection Name ⌵

comments

Storage size: 4.10 kB	Documents: 50 K	Avg. document size: 284.00 B	Indexes: 1	Total index size: 4.10 kB
-----------------------	-----------------	------------------------------	------------	---------------------------

movies

Storage size: 4.10 kB	Documents: 24 K	Avg. document size: 1.60 kB	Indexes: 1	Total index size: 4.10 kB
-----------------------	-----------------	-----------------------------	------------	---------------------------

sessions

Storage size: 4.10 kB	Documents: 1	Avg. document size: 540.00 B	Indexes: 1	Total index size: 4.10 kB
-----------------------	--------------	------------------------------	------------	---------------------------

theatres

Storage size: 4.10 kB	Documents: 1.6 K	Avg. document size: 223.00 B	Indexes: 1	Total index size: 4.10 kB
-----------------------	------------------	------------------------------	------------	---------------------------

users

Storage size: 4.10 kB	Documents: 185	Avg. document size: 159.00 B	Indexes: 1	Total index size: 4.10 kB
-----------------------	----------------	------------------------------	------------	---------------------------

Exercici 1

- Mostra els 2 primers comentaris que hi ha en la base de dades.
Entramos en la colecció comments y en MongoDB shell indicamos la agregación LIMIT 2, para que nos devuelva solo los 2 primeros registros

The screenshot shows the MongoDB Shell interface with the following commands and results:

```
> use Entreteniment
switched to db Entreteniment
> db["comments"].aggregate([{$limit:2}])
{
  "_id": ObjectId("5a9427648b0beebe69579cc"),
  "name": "Andrea Le",
  "email": "andrea_le@fakegmail.com",
  "movie_id": ObjectId("573a1390f29313caabcd418c"),
  "text": "Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibu",
  "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 ve",
  "date": 1987-02-10T00:29:36.000Z
}
```

The MongoDB Compass interface shows the 'Aggregations' tab for the 'comments' collection. The aggregation pipeline is defined as:

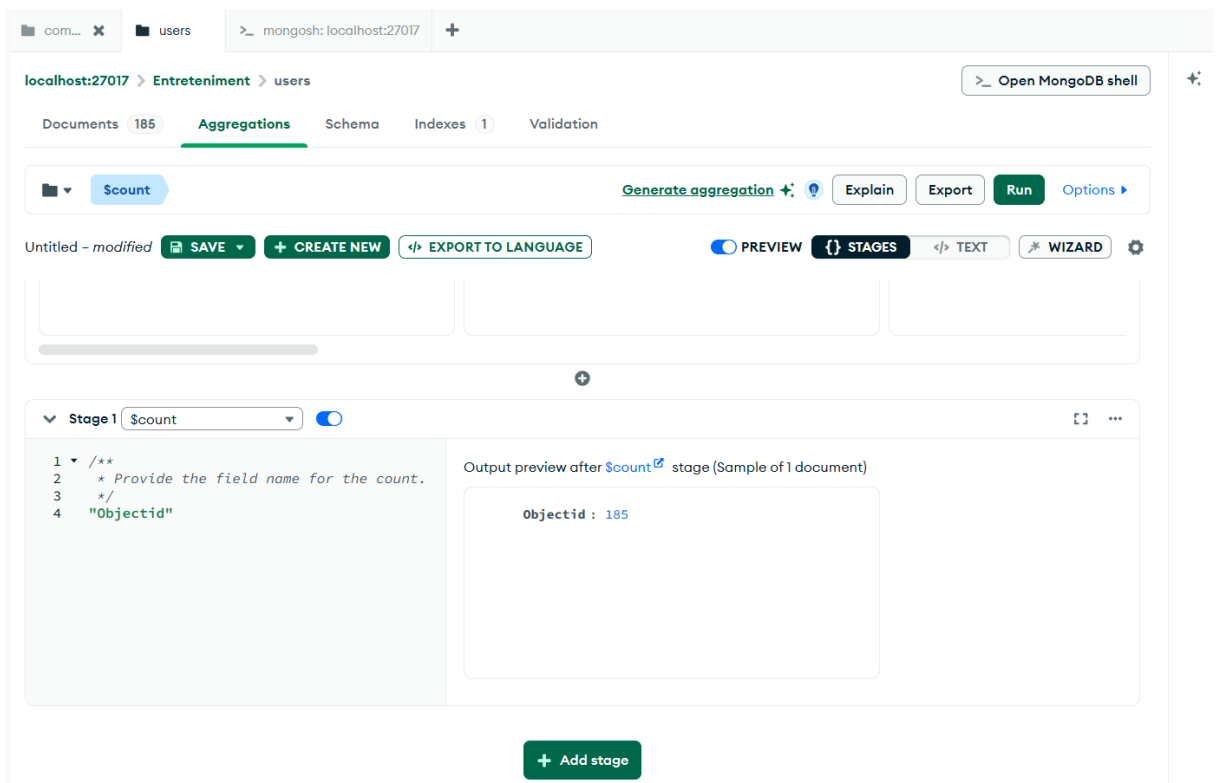
```
[
  {
    "$limit": 2
  }
]
```

The output preview shows the first two documents:

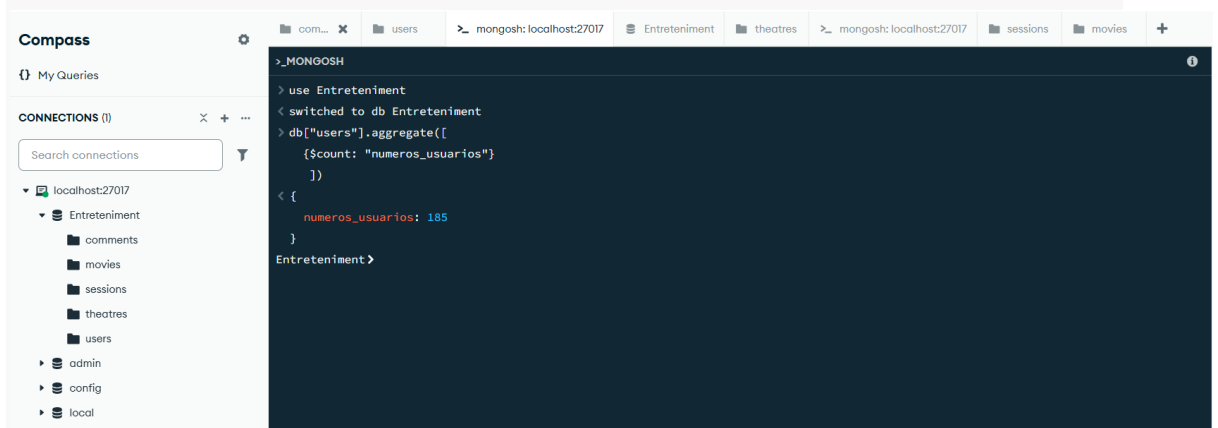
```
{
  "_id": "5a9427648b0beebe69579cc",
  "name": "Andrea Le",
  "email": "andrea_le@fakegmail.com",
  "movie_id": "573a1390f29313caabcd418c",
  "text": "Rem officiis eaque repellendus amet eos doloribus. Porro dolor volupta...",
  "date": "2012-03-26T23:20:16.000+00:00"
}
```

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

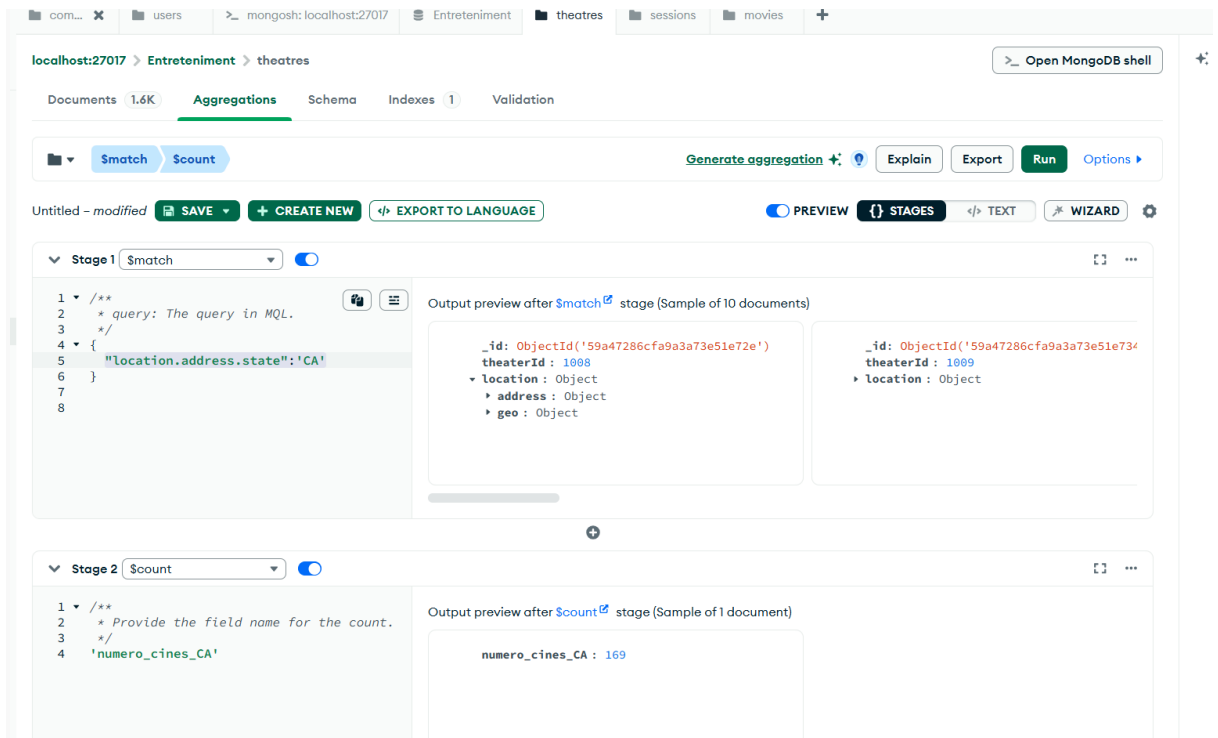
- Quants usuaris tenim registrats?
Para obtener los usuarios registrados utilizamos el aggregate \$count e indicamos que cuente los diferentes "ObjectId".



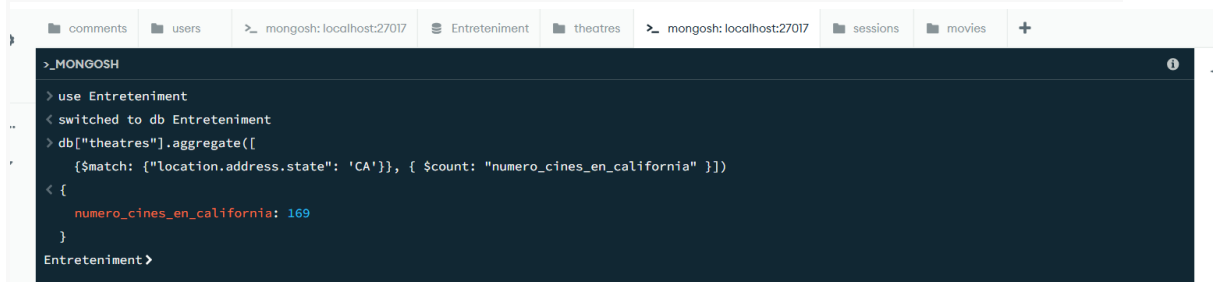
Si el cálculo lo hacemos desde PowerShell, indicamos que cuente los usuarios e indique el total en el campo “numero_usuarios”.



- Quants cinemes hi ha en l'estat de Califòrnia?
Para obtener este dato en primer lugar tenemos que filtrar, utilizando la funcion aggregate \$match indicando que el estado es = CA. Como el campo stat está dentro del grupo location y adress, tenemos que indicarlo entre comillas y separando los campos con puntos **"location.address.state":'CA'**. Posteriormente haremos el count para saber los cines que hay . En Compass hacemos la consulta en 2 passos,, utilizando 2 stages, en el primero haremos el match y en el segudno el count



En Shell hacemos toda la consulta a la vez:



- Quin va ser el primer usuari/ària en registrar-se?
 Como la colección user no tiene campo fecha de registro, hacemos la consulta sobre la colección comments.
 En Compass, en primer lugar creamos una nueva agregación, para no hacer la consulta sobre el resultado anterior y procedemos a ordenar por date, de forma ascendente y seleccionamos el primer valor

localhost:27017 > Entreteniment > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

Sort Limit Generate aggregation Explain Export Run Options

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

Stage 1 \$sort

```

1 /**
2  * Provide any number of field/order pair
3  */
4 {
5   date: 1
6 }

```

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

```

_id: ObjectId('5a9427648b0beebe695e67e')
name: "Mercedes Tyler"
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId('573a1399f29313caabcec3c...')
text: "Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cum..."
date: 1970-01-01T01:07:09.000+00:00

```

Stage 2 \$limit

```

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

```

Output preview after \$limit stage (Sample of 1 document)

```

_id: ObjectId('5a9427648b0beebe695e67e')
name: "Mercedes Tyler"
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId('573a1399f29313caabcec3c...')
text: "Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cum..."

```

En Shell hacemos la misma consulta , pero toda a la vez, ordenando en primer lugar y solicitando el primer registro en segunda instancia

com... x mongosh: localhost:27017 users Entreteniment theatres mongosh: localhost:27017 sessions movies +

```

>_MONGOSH
> use Entreteniment
< switched to db Entreteniment
> db["comments"].aggregate([
  { $sort:{date : 1}},
  {$limit: 1}}]
< {
  _id: ObjectId('5a9427648b0beebe695e67e'),
  name: 'Mercedes Tyler',
  email: 'mercedes_tyler@fakegmail.com',
  movie_id: ObjectId('573a1399f29313caabcec3ce'),
  text: 'Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cumque animi. Quia nam doloribus corrupti id voluptate esse. Tempore m
  date: 1970-01-01T01:07:09.000Z
}
Entreteniment>

```

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

Para saberlo, en primer lugar filtramos, con la agregación match, para obtener los registros que son de tipo películas ('movie') y que su género es la comedia. Y una vez tenemos este stage hecho hacemos un count para obtener las películas de comedia que hay en la base de datos.

The screenshot shows the MongoDB Compass interface for a database named 'Entreteniment' and a collection named 'movies'. The 'Aggregations' tab is active, showing a pipeline with two stages:

- Stage 1 (\$match):** The query is `{ genres: "Comedy", type: "movie" }`. The output preview shows a sample of 10 documents, including details like `_id`, `plot`, `genres`, `runtime`, `cast`, `num_mflix_comments`, and `poster`.
- Stage 2 (\$count):** The query is `'Pellicules de comedia'`. The output preview shows a single document with the result: `'Pellicules de comedia' : 7002`.

Para hacer la consulta en MongoDB Shell, hacemos la doble consulta a la vez:

```
>_MONGOSH
> use Entreteniment
< switched to db Entreteniment
> db["movies"].aggregate([
  {
    $match: { genres: "Comedy",
              type: "movie" },
    { $count: "pellicules de comedia" }
  ]
)
{
  'pellicules de comedia': 7002
}
Entreteniment>
```

Exercici 2

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

En este caso hago hacemos toda la consulta en un mismo stage. Una agregación para filtrar con la condición de que muestre los documentos sin el género es drama o el idioma es en francés, y de los resultados obtenidos que se muestren solo los que el año de la película sea 1932 .

The screenshot shows the MongoDB Compass interface for a database named 'movies'. The 'Aggregations' tab is active, and a pipeline with one stage, '\$match', is defined. The query is:

```

1 // **
2 * query: The query in MQL.
3 */
4 {
5   $or: [
6     {genres: 'Drama'},
7     {languages: 'French'}
8   ],
9   year: 1932
10 }

```

The output preview shows two sample 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": [
    "Enrique Rivero",
    "Elizabeth Lee Miller",
    "Pauline Carton",
    "Odette Talazac"
  ],
  "num_mflix_comments": 1,
  "poster": "https://m.media-amazon.com/images/M/MV5BYWY3ODZSWZEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTxxXkEyXkFqcGdeQXVyNzI4MDMyMTU@._V1_...",
  "title": "The Blood of a Poet",
  "lastupdated": "2015-09-16 13:13:05.537000000"
}

```

En Shell hacemos la misma consulta que en Compass, dentro del match indicamos en primer lugar, que busque que las películas donde el género sea drama o la lengua francés y que de los resultados seleccione los que sean del año 1932.

The screenshot shows the MongoDB Shell interface. The same aggregation pipeline is entered:

```

> db["movies"].aggregate([
  { $match:
    {
      $or: [
        {genres: 'Drama'},
        {languages: 'French'}
      ],
      year: 1932
    }
  ]
})

```

The output shows a single document:

```

{
  "_id": "ObjectId('573a1391f29313caabcd9458')",
  "plot": "A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing comes into life and starts talking.",
  "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/MV5BYWY3ODZSWZEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTxxXkEyXkFqcGdeQXVyNzI4MDMyMTU@._V1_...",
  "title": "The Blood of a Poet",
  "lastupdated": "2015-09-16 13:13:05.537000000"
}

```

En Shell también tenemos la opción de realizar la consulta mediante find


```
movies  mongosh: localhost:27017  +
>_MONGOSH
}
> db["movies"].find({
  $or: [
    {genres : ['Drama']},
    {languages: ['French']}
  ],
  year: 1932})
< {
  _id: ObjectId('573a1391f29313caabcd9458'),
  plot: 'A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing comes into life and starts talking.',
  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/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0VTYxXkEyXkFqcGdeQXVyNzI4MDMyMTU0._V1_
  title: 'The Blood of a Poet',
  lastupdated: '2015-09-16 13:13:05.537000000',
  languages: [
    'French'
  ],
  released: 2010-05-20T00:00:00.000Z,
  directors: [
```

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.

Generamos una búsqueda donde indicamos en primer lugar el país tienes que ser 'USA', en segundo lugar la condición que los premios ganados sean mayor o igual que 5 y menores o iguales que 9 y, por último, indicamos que el año tiene que ser mayor o igual a 2012 y menor o igual que 2014.

En este caso realizamos la misma consulta tanto en Compass como Shell

movies

localhost:27017

+

localhost:27017

Entertainment

movies

Open MongoDB shell

Documents 23.5K

Aggregations

Schema

Indexes 1

Validation

{

countries: 'USA',

awards.wins: {

\$gte: 5,

\$lte: 9},

year: {

\$gte: 2012,

\$lte: 2014}

}

Generate query

Explain

Reset

Find

Options

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 25 of 166

```

_id: ObjectId('573a13acf29313caabd29366')
Fullplot: "The manager of the negative assets sector of Life magazine, Walter Mit..."
imdb: Object
  year: 2013
  plot: "When his job along with that of his co-worker are threatened, Walter t..."
genres: Array (3)
  rated: "PG"
  metacritic: 54
  title: "The Secret Life of Walter Mitty"
  lastupdated: "2015-08-31 00:10:51.747000000"
languages: Array (3)
writers: Array (3)
  type: "movie"
tomatoes: Object
  poster: "https://m.media-amazon.com/images/M/MV5BODYwNDYxNDk1L5BM15BanBnXkFtZT..."
  num_flix_comments: 1
  released: 2013-12-25T00:00:00.000+00:00
awards: Object
countries: Array (2)
cast: Array (4)
directors: Array (1)
runtime: 114

```

```

_id: ObjectId('573a13b5f29313caabd45772')
plot: "After their cave is destroyed, a caveman family must trek through an u..."
genres: Array (3)
runtime: 98

```

Nivell 2

Exercici 1

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

Escribe un total de 22841 comentarios.

Lo calculamos haciendo una búsqueda, que cuente aquellos documentos de email que coincida con el string “gameofthron.es”. Indicamos que además no tenga en cuenta mayúsculas o minúsculas, para obtener todas las respuestas.

The screenshot shows the MongoDB Compass interface. At the top, there are tabs for 'movies', 'comments', 'theatres', and 'validation'. The 'comments' tab is selected. Below the tabs, there is a search bar with the query: `{email: {$regex: /gameofthron.es/, $options: 'i' }}`. The results are displayed in a list, showing the first four documents. Each document contains fields: `_id`, `name`, `email`, `movie_id`, `text`, and `date`. The total number of results is 22841, which is highlighted in a green box. The interface also includes buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'.

localhost:27017 > Entreteniment > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

{email: {\$regex: /gameofthron.es/, \$options: 'i' }}

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 25 of 22841

date: 1998-09-04T04:41:51.000+00:00

_id: ObjectId('5a9427648b0beebe6957a08')
name: "Meera Reed"
email: "ellie_kendrick@gameofthron.es"
movie_id: ObjectId('573a1390f29313caabcd4964')
text: "Harum porro ad dolorum repellendus. Nihil natus aspernatur quaerat ape..."
date: 1971-08-31T07:24:20.000+00:00

_id: ObjectId('5a9427648b0beebe6957a09')
name: "Daario Naharis"
email: "michiel_huisman@gameofthron.es"
movie_id: ObjectId('573a1390f29313caabcd4aa2')
text: "Distinctio commodi autem amet molestias. Dolorem numquam vitae volupta..."
date: 1979-07-06T20:36:21.000+00:00

_id: ObjectId('5a9427648b0beebe6957a0e')
name: "Viserys Targaryen"
email: "harry_lloyd@gameofthron.es"
movie_id: ObjectId('573a1390f29313caabcd4b1b')
text: "Itaque minima enim mollitia rerum hic. Doloribus dolor nobis corporis ..."
date: 2010-03-20T02:11:01.000+00:00

_id: ObjectId('5a9427648b0beebe6957a0f')
name: "Melisandre"

```
movies  comments  >_ mongosh: localhost:27017  +

>_MONGOSH

> use Entreteniment
< switched to db Entreteniment
> db["comments"].countDocuments({
  email:{$regex: /@gameofthron.es/, $options: 'i'}
})
< 22841
Entreteniment>
```

Exercici 2

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

Para obtener los cines que hay en el estado de Washington DC, hacemos una agregación `$match`, para indicar que solo queremos aquellos documentos donde el estado sea "DC". Y una vez los tenemos agrupamos por código postal y los contamos.

localhost:27017 > Entreteniment > theatres

Documents1.6K

Aggregations

Schema

Indexes1

Validation

>_ Open MongoDB shell

▼

\$match

\$group

Generate aggregation

⚡

ⓘ

Explain

Export

Run

Options ▶

Untitled – modified

SAVE

+ CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

WIZARD

⚙

▼ Stage 1 | \$match

ⓘ

⋮

1 {

2 "location.address.state": 'DC'

3 }

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

_id: ObjectId('59a47286cfa9a3a73e51e785')

theaterId: 1092

location: Object

_id: ObjectId('59a47286cfa9a3a73e51e785')

theaterId: 801

location: Object

+

▼ Stage 2 | \$group

ⓘ

⋮

1 {

2 _id: "\$location.address.zipcode",

3 codis_postals: {

4 \$sum: 1

5 }

6 }

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

_id: "20016"

codis_postals: 1

_id: "20010"

codis_postals: 1

localhost:27017 > Entreteniment > theatres

>_ Open MongoDB shell

Documents1.6K

Aggregations

Schema

Indexes1

Validation

▼

\$match

\$group

Edit

ⓘ

Explain

Export

Run

Options ▶

ALL RESULTS

Showing 1 – 3 count results

<

>

▼

☰

{ }

_id: "20016"

codis_postals: 1

_id: "20010"

codis_postals: 1

_id: "20002"

codis_postals: 1

```
theatres  >_ mongosh: localhost:27017  Entreteniment  +

>_MONGOSH

> use Entreteniment
< switched to db Entreteniment
> db["theatres"].aggregate([
  $match:
    {'location.address.state':'DC'}
  },
  {$group:{
    _id: '$location.address.zipcode',
    codis_postals:{$sum:1}
  }}
])
< {
  _id: '20002',
  codis_postals: 1
}
{
  _id: '20010',
  codis_postals: 1
}
{
  _id: '20016',
  codis_postals: 1
}
Entreteniment >
```

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.

Hacemos una búsqueda por las películas que el director sea John Landis y añadimos la condición que la puntuación de imdb esté entre 7.5 y 8.

theatres movies mongosh: localhost:27017 mongosh: localhost:27017 Entreteniment +

localhost:27017 > Entreteniment > movies Open MongoDB shell

Documents 23.5K Aggregations Schema Indexes 1 Validation

{directors: 'John Landis', "imdb.rating": {\$gte: 7.5, \$lte: 8} [Generate query](#) Explain Reset Find </> Options

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - 4 of 4

```

_id: ObjectId('573a1397f29313caabce6d94')
fullplot: "Faber College has one frat house so disreputable it will take anyone. ..."
imdb: Object
  year: 1978
  plot: "At a 1962 college, Dean Vernon Wormer is determined to expel the entire..."
  genres: Array (1)
  rated: "R"
  metacritic: 82
  title: "Animal House"
  lastupdated: "2015-09-13 00:02:47.803000000"
  languages: Array (2)
  writers: Array (3)
  type: "movie"
  tomatoes: Object
    poster: "https://m.media-amazon.com/images/M/MV5BM2M2ZDA4MTYtOGRjMi00OTg5LWI1ZT..."
    num_mflix_comments: 1
    released: 1978-07-28T00:00:00.000+00:00
  awards: Object
  countries: Array (1)
  cast: Array (4)
  directors: Array (1)
  runtime: 109

```

```

_id: ObjectId('573a1397f29313caabce76f7')

```

```

> MONGOSH
> use Entreteniment
< switched to db Entreteniment
> db["movies"].find({directors: 'John Landis',
                    "imdb.rating": {$gte: 7.5, $lte: 8}
                    })
< {
  _id: ObjectId('573a1397f29313caabce6d94'),
  fullplot: "Faber College has one frat house so disreputable it will take anyone. It has a second one full of white, anglo-saxon
  imdb: {
    rating: 7.6,
    votes: 84834,
    id: 77975
  },
  year: 1978,
  plot: 'At a 1962 college, Dean Vernon Wormer is determined to expel the entire Delta Tau Chi Fraternity, but those trouble-make
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
  title: 'Animal House',
  lastupdated: '2015-09-13 00:02:47.803000000',
  languages: [
    'English',
    'Italian'
  ],
}

```

La consulta la podemos realizar también mediante agregación con un \$match:

Documents 23.5K Aggregations Schema Indexes 1 Validation

Smatch

Generate aggregation

Explain

Export

Run

Options

Untitled - modified

SAVE

+ CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

WIZARD



Stage 1 Smatch



```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   directors: 'John Landis',
6   "imdb.rating": {$gte: 7.5, $lte: 8}
7 }
```

Output preview after \$match stage (Sample of 4 documents)

```
_id: ObjectId('573a1397f29313caabce7d06')
fullplot: "Two American college students
are on a walking tour of Britain
and are..."
imdb: Object
year: 1981
plot: "Two American college students on a
walking tour of Britain are attacke..."
genres: Array (2)
```

```
_id: ObjectId('573a
plot: "A snobbish i
street con ar
positions ..."
genres: Array (1)
runtime: 116
metacritic: 66
rated: "R"
cast: Array (4)
```

Documents 23.5K Aggregations Schema Indexes 1 Validation

Smatch

Edit

Explain

Export

Run

Options

ALL RESULTS

Showing 1 - 4 count results



```
_id: ObjectId('573a1397f29313caabce6d94')
fullplot: "Faber College has one frat house so disreputable it will take anyone. ..."
imdb: Object
year: 1978
plot: "At a 1962 college, Dean Vernon Wormer is determined to expel the entire ..."
genres: Array (1)
rated: "R"
metacritic: 82
title: "Animal House"
lastupdated: "2015-09-13 00:02:47.803000000"
languages: Array (2)
writers: Array (3)
type: "movie"
tomatoes: Object
poster: "https://m.media-amazon.com/images/M/MV5BM2M2ZDA4MTYtOGZjMi00Tg5LWI1ZT..."
num_mflix_comments: 1
released: 1978-07-28T00:00:00.000+00:00
awards: Object
countries: Array (1)
cast: Array (4)
directors: Array (1)
runtime: 109
```



```
theatres movies >_ mongosh: localhost:27017 >_ mongosh: localhost:27017 Entreniment +
>_MONGOSH
> db["movies"].aggregate([
  $match:
    {directors: 'John Landis',
      "imdb.rating": {$gte: 7.5, $lte: 8}
    }
  ])
< {
  _id: ObjectId('573a1397f29313caabce6d94'),
  fullplot: "Faber College has one frat house so disreputable it will take anyone. It has a second one full of white, anglo-saxon,
  imdb: {
    rating: 7.6,
    votes: 84834,
    id: 77975
  },
  year: 1978,
  plot: 'At a 1962 college, Dean Vernon Wormer is determined to expel the entire Delta Tau Chi Fraternity, but those trouble-maker
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
  title: 'Animal House',
  lastupdated: '2015-09-13 00:02:47.803000000',
  languages: [
    'English',
```

Exercici 2

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

Analizamos el schema, de la colección theatres, y desplegamos el bloque location, donde nos aparece el mapa de la ubicación de cada uno de los cines,

theatres +

localhost:27017 > Entreteniment > theatres

[Open MongoDB shell](#)

Documents 1.6K Aggregations Schema Indexes 1 Validation

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

Reset

Analyze



Options

[EXPORT SCHEMA](#)

This report is based on a sample of 1000 documents. [Learn more](#)

document

geo

coordinates

