

Tarea S5.01. Consultas con MongoDB

Patricia Proniewska

Descripción

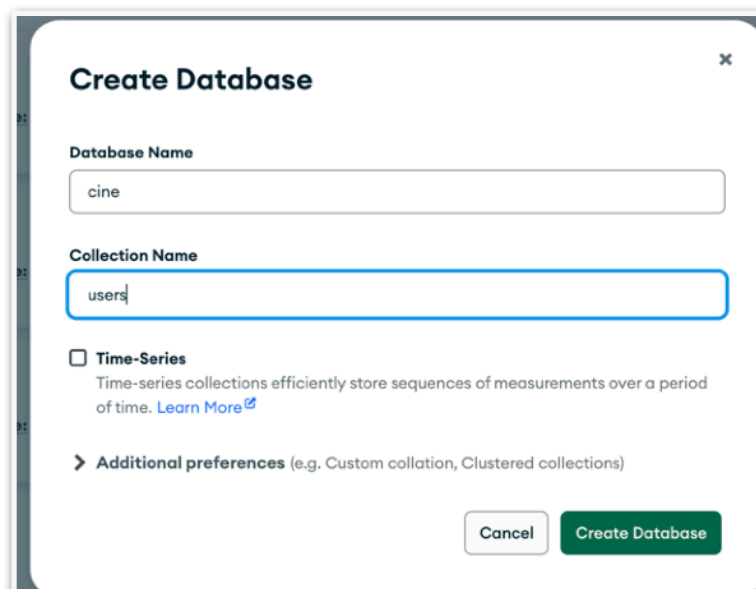
Trabajaremos con una base de datos que contiene colecciones relacionadas con una aplicación de entretenimiento cinematográfico:

- users: Almacena información de usuarios/as, incluyendo nombres, emails y contraseñas cifradas.
- theatres: Contiene datos de cines, como ID, ubicación (dirección y coordenadas geográficas).
- sesiones: Guarda sesiones de usuario, incluyendo ID de usuario y tokens JWT para la autenticación.
- movies: Incluye detalles de películas, como trama, géneros, duración, elenco, comentarios, año de lanzamiento, directores, clasificación y premios.
- comments: Almacena comentarios de usuarios/as sobre películas, con información del autor/a del comentario, ID de la película, texto del comentario y la fecha.

Realizarás algunas consultas que te pide el cliente/a, quien está midiendo si serás capaz o no de hacerte cargo de la parte analítica del proyecto vinculado con su base de datos.

Nivel 1

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



The screenshot shows the 'Create Database' dialog box in MongoDB. It has a title bar with a close button (X). The form contains two input fields: 'Database Name' with the value 'cine' and 'Collection Name' with the value 'users'. Below these fields is a checkbox labeled 'Time-Series' which is unchecked. A tooltip for the 'Time-Series' checkbox explains that time-series collections efficiently store sequences of measurements over a period of time and provides a 'Learn More' link. At the bottom, there is a link for 'Additional preferences' (e.g. Custom collation, Clustered collections) and two buttons: 'Cancel' and 'Create Database'.

Create Database

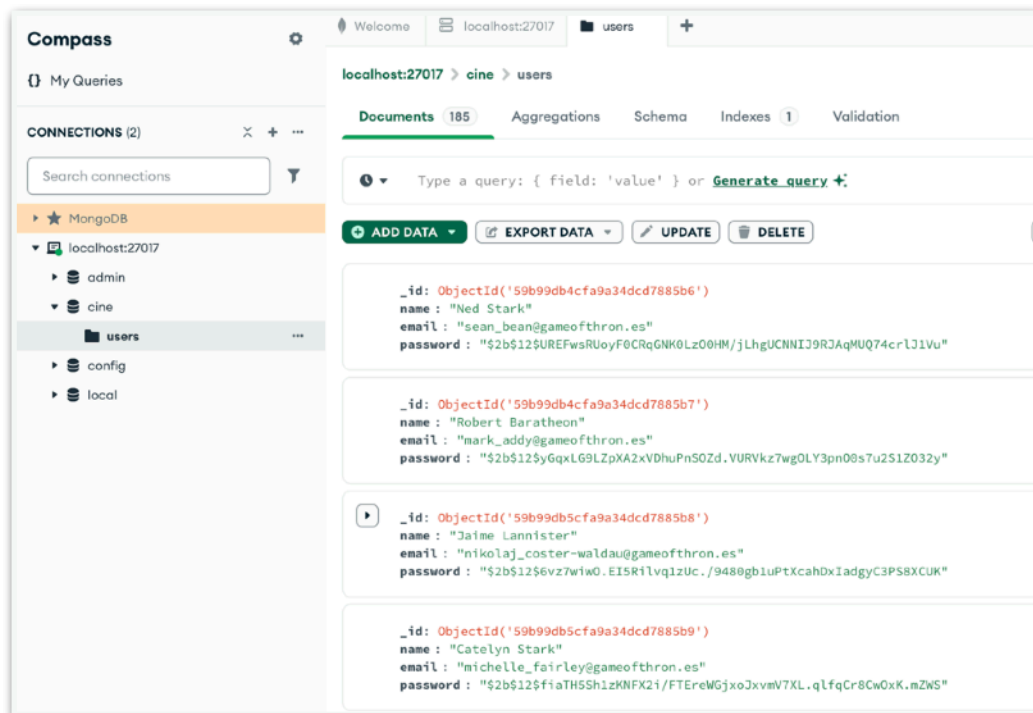
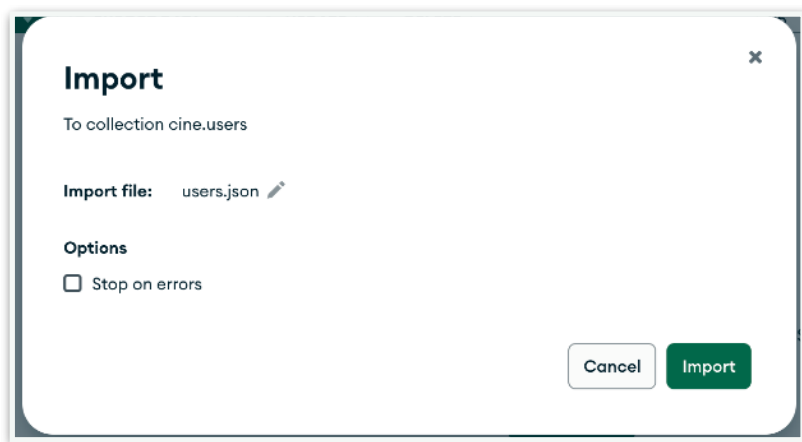
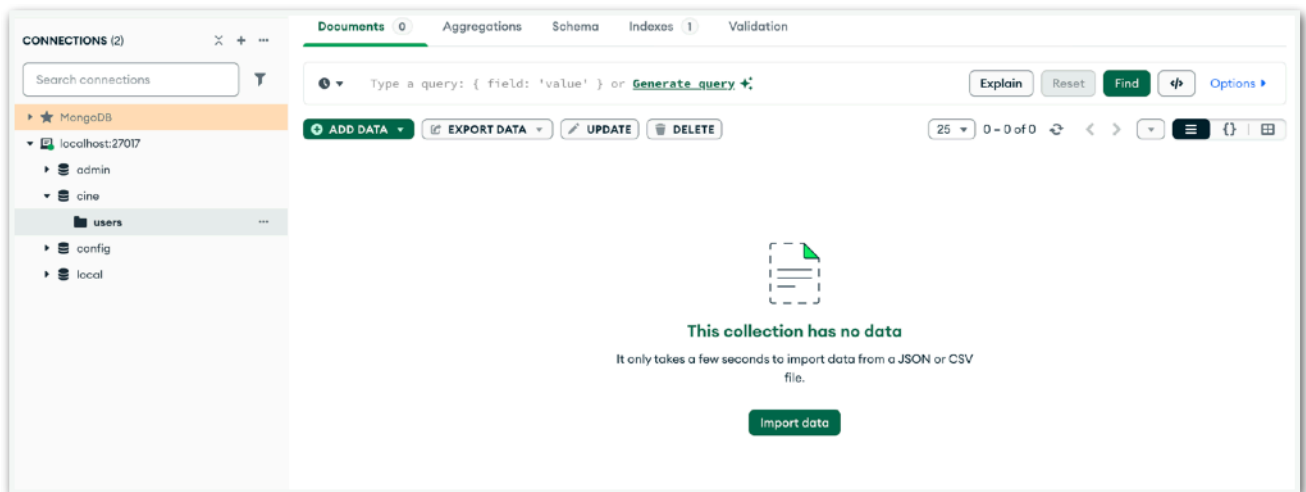
Database Name
cine

Collection Name
users

☐ **Time-Series**
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> **Additional preferences** (e.g. Custom collation, Clustered collections)

Cancel Create Database



Create Collection

Collection Name

theatres

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences (e.g. Custom collation, Clustered collections)

CancelCreate Collection

Compass

My Queries

CONNECTIONS (2)

Search connections

MongoDB

localhost:27017

admin

cine

theatres

users

config

local

localhost:27017 > cine > theatres

Documents 0

Aggregations

Schema

Indexes 1

Validation

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

ADD DATA

EXPORT DATA

UPDATE

DELETE

25 0 -

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Import data

Compass

My Queries

CONNECTIONS (2)

Search connections

MongoDB

localhost:27017

admin

cine

theatres

users

config

local

localhost:27017 > cine > theatres

Documents 1.6K

Aggregations

Schema

Indexes 1

Validation

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

ADD DATA

EXPORT DATA

UPDATE

DELETE

25 1 - 25 of 1564

_id: ObjectId('59a47286cfa9a3a73e51e72c')

theaterId: 1000

location: Object

_id: ObjectId('59a47286cfa9a3a73e51e72d')

theaterId: 1003

location: Object

_id: ObjectId('59a47286cfa9a3a73e51e72e')

theaterId: 1008

location: Object

_id: ObjectId('59a47286cfa9a3a73e51e72f')

theaterId: 1004

location: Object

_id: ObjectId('59a47286cfa9a3a73e51e730')

theaterId: 1002

location: Object

Object

Create Collection

Collection Name

sessions

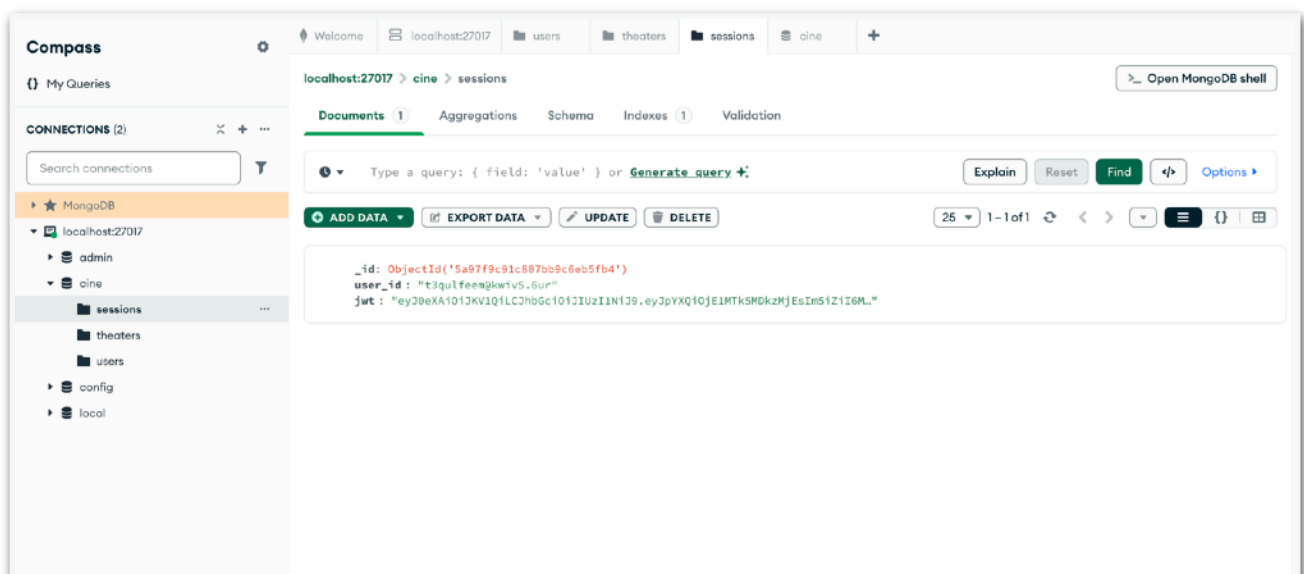
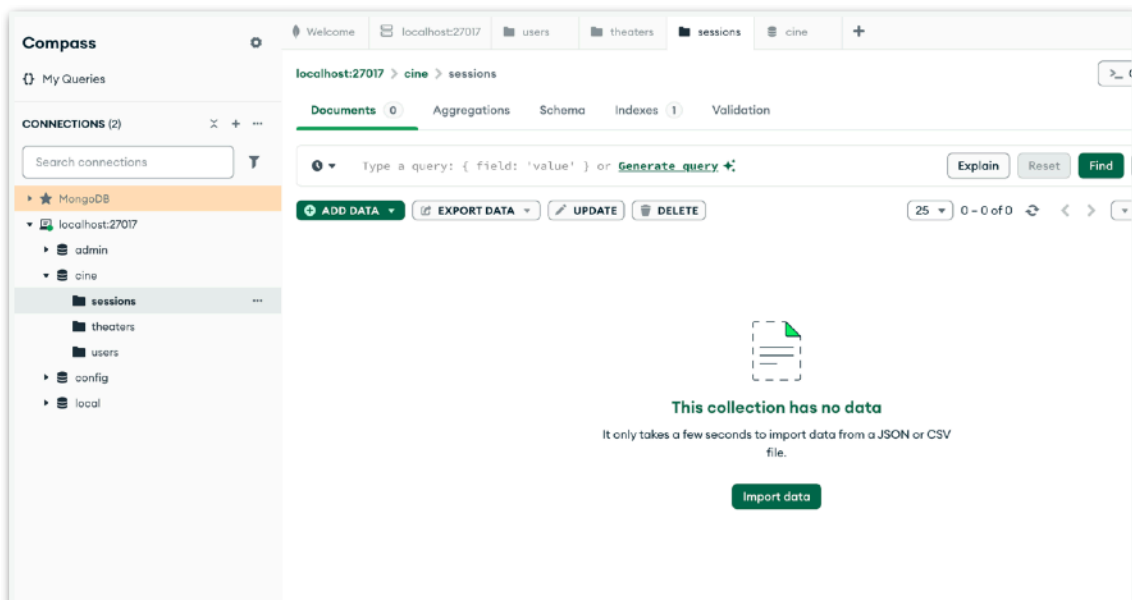
☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

[Additional preferences](#) (e.g. Custom collation, Clustered collections)

Cancel

Create Collection



Create Collection

Collection Name

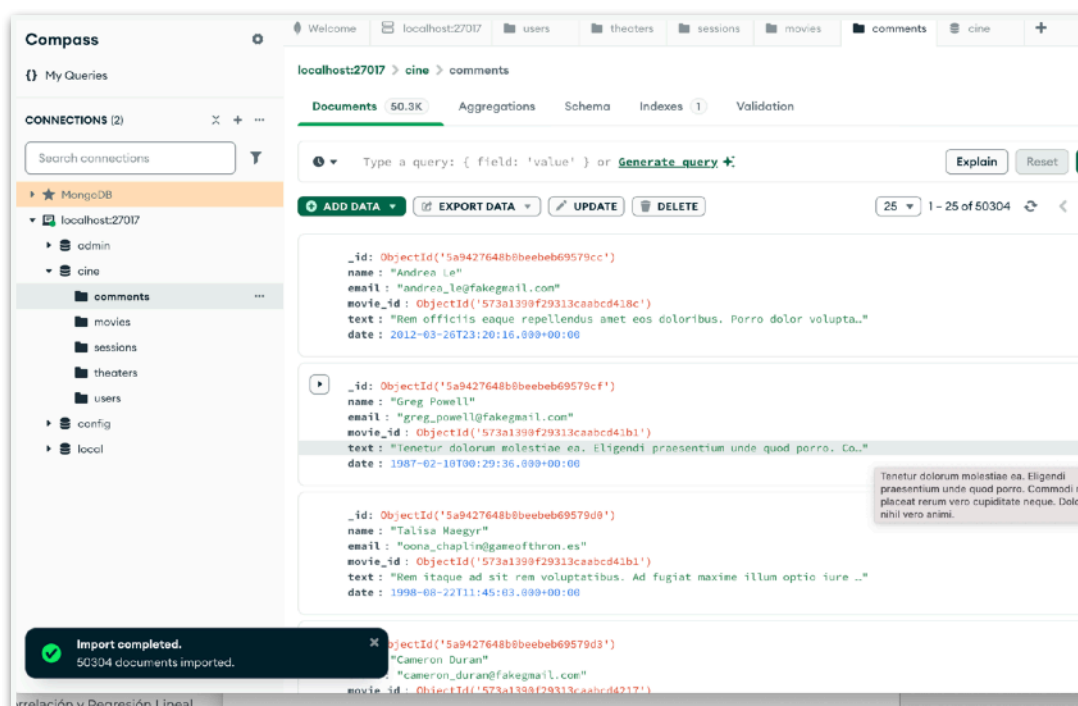
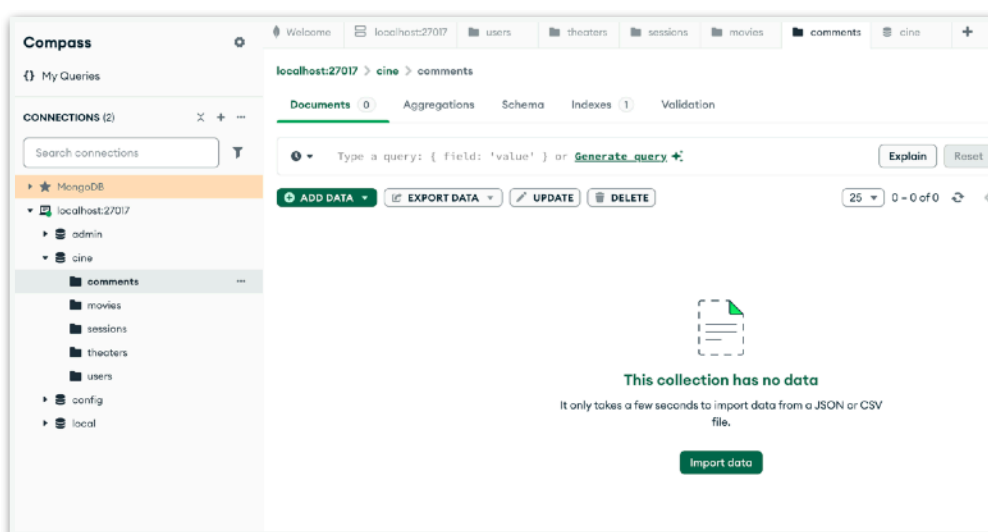
comments

☐ Time-Series

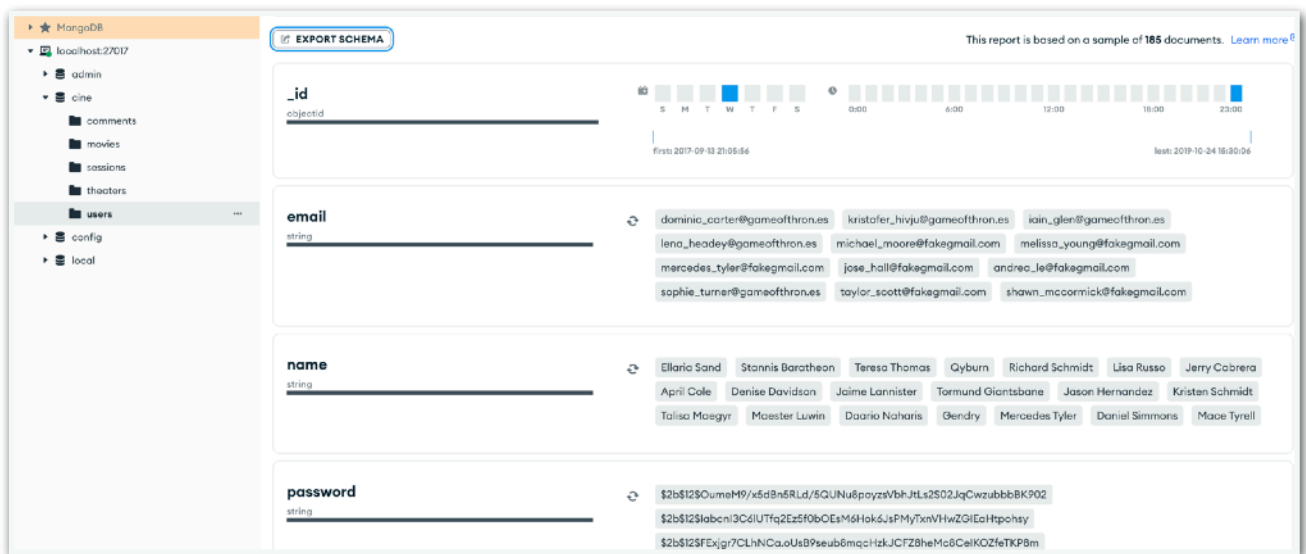
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

➤ Additional preferences (e.g. Custom collation, Clustered collections)

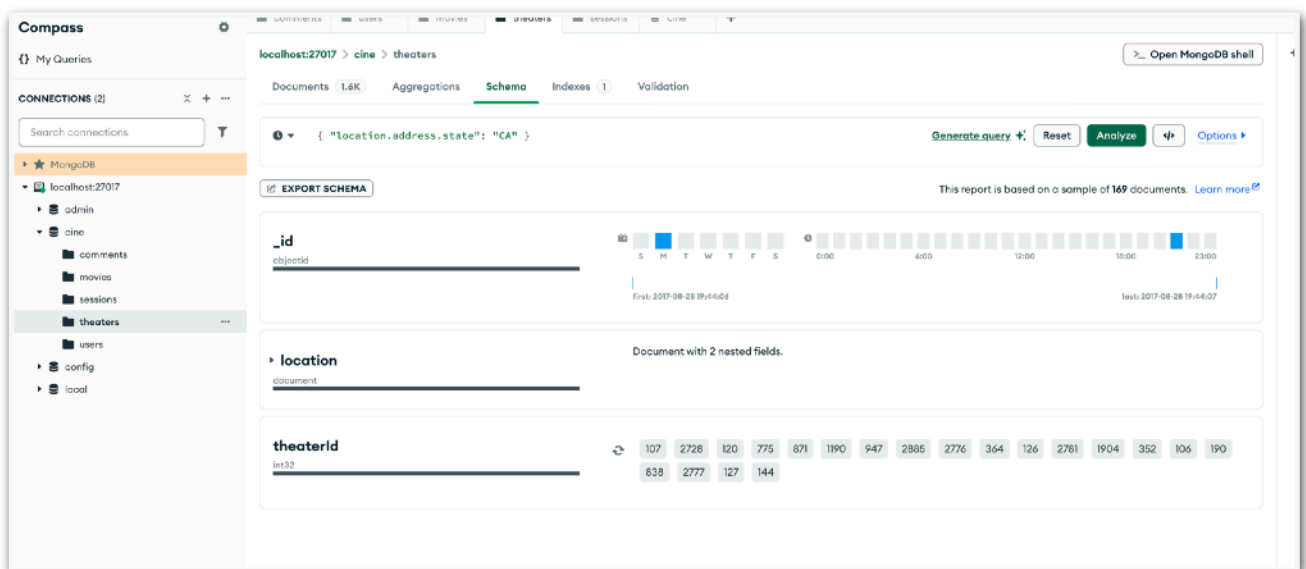
CancelCreate Collection



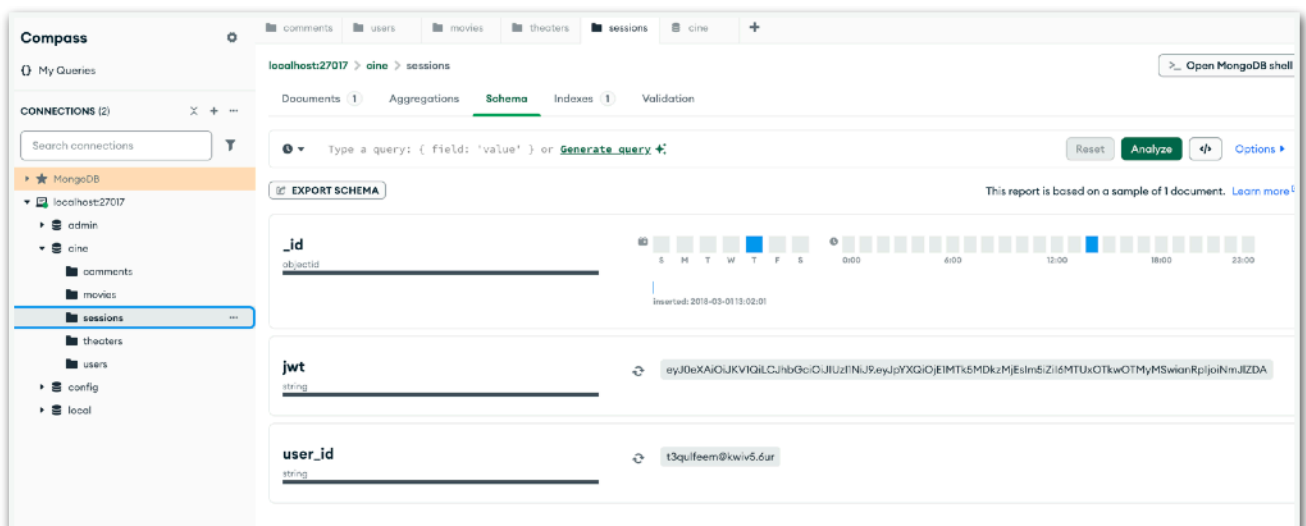
Schema users:



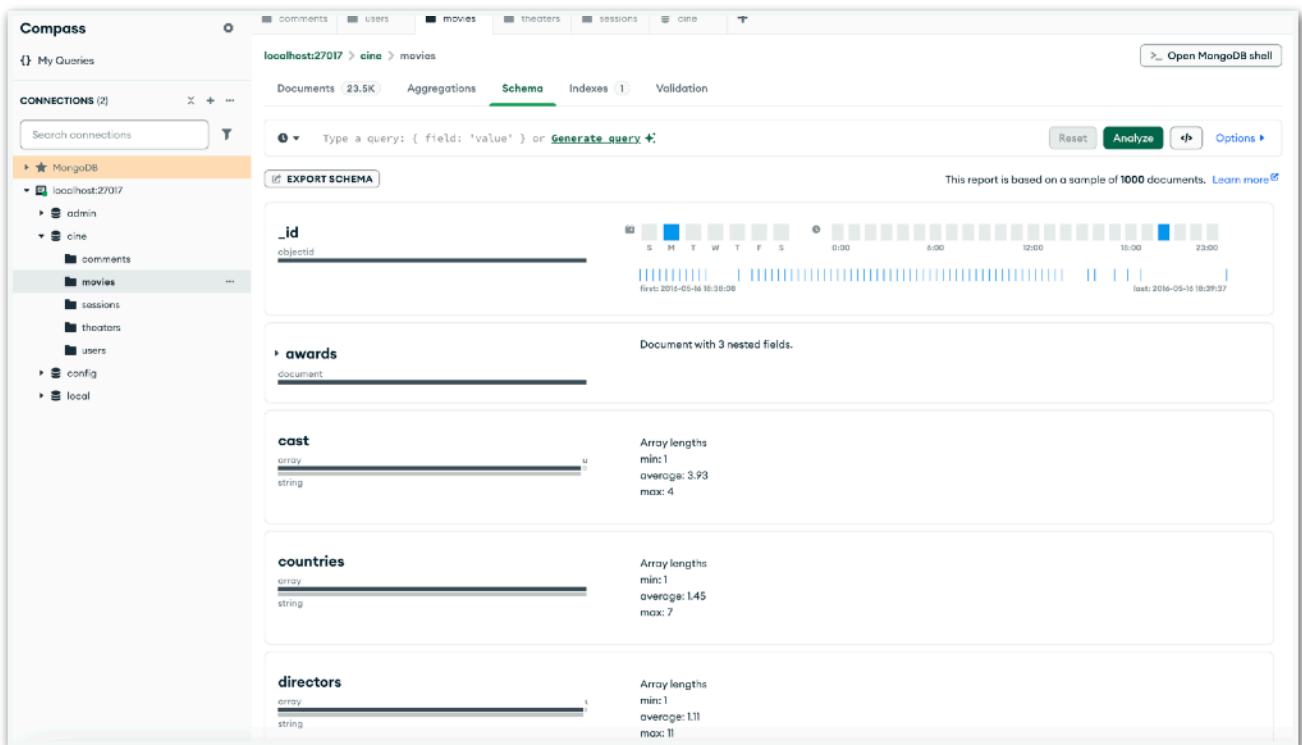
Schema theaters:



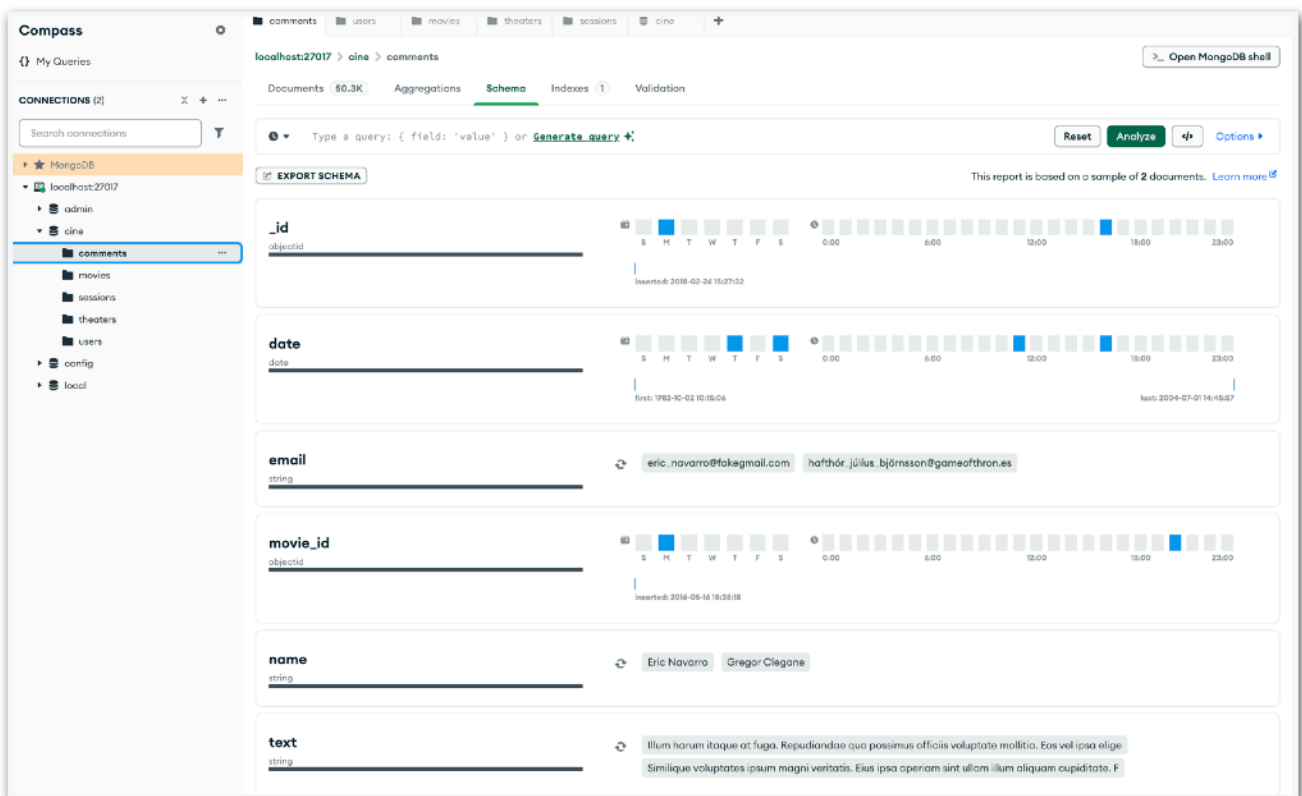
Schema sessions:



Schema movies:

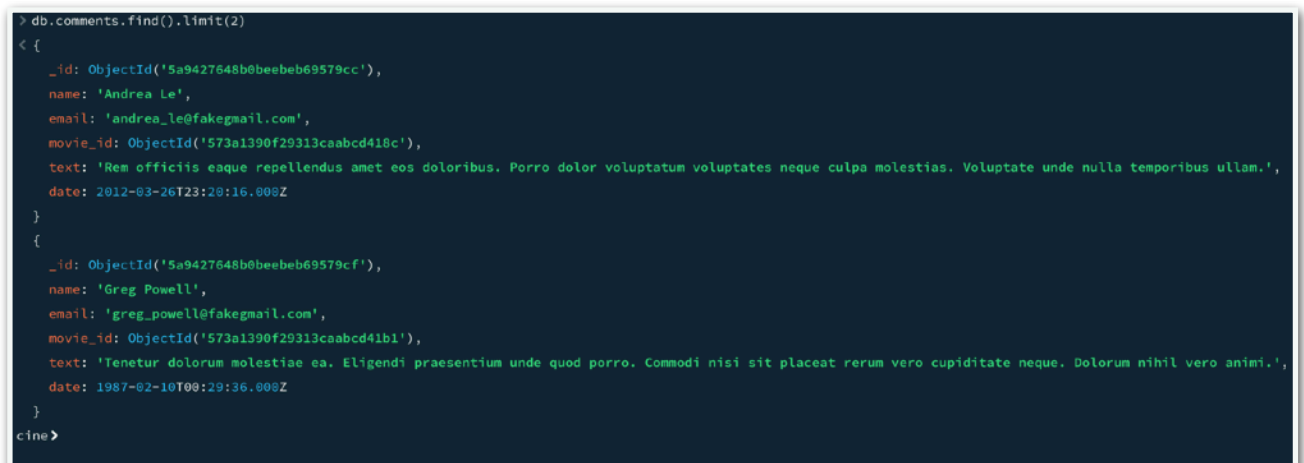
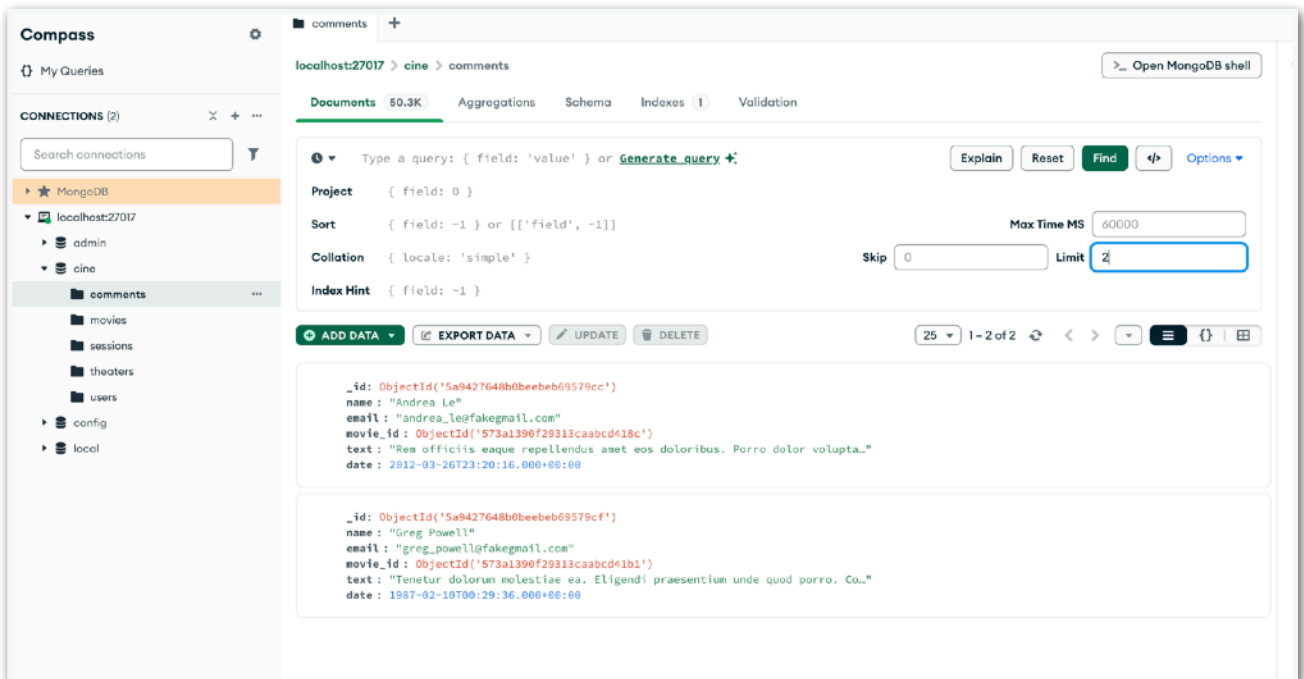


Schema comments:

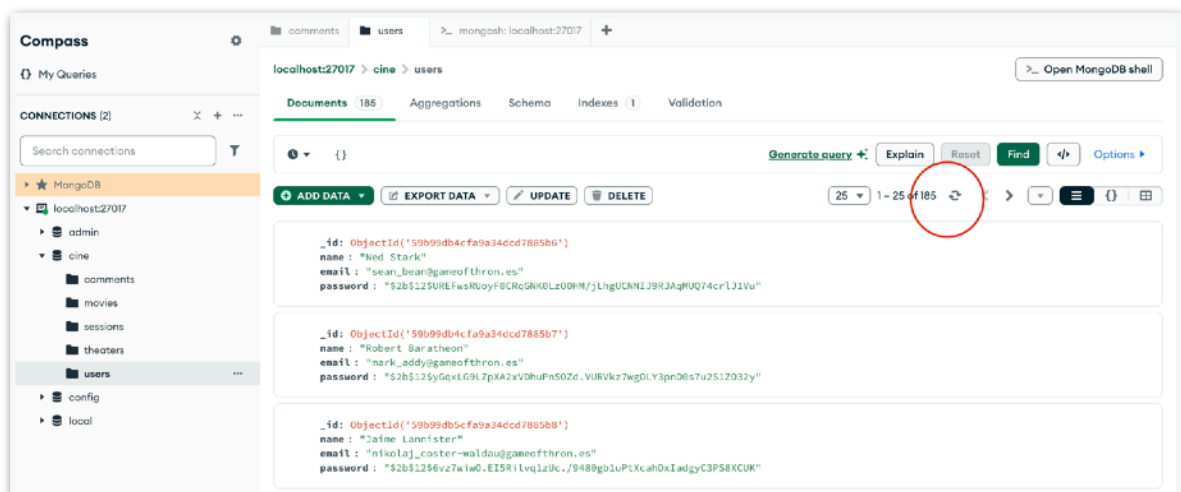


Ejercicio 1

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



- ¿Cuántos usuarios tenemos registrados? - **185 usuarios**



Documents 185 Aggregations Schema Indexes 1 Validation

Generate aggregation + Explain Export Run Options

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

185 Documents in the collection

Preview of documents

```
{ "_id": ObjectId("59b99db4cf9a34dcd7885b6"), "name": "Ned Stark", "email": "sean_bean@gameofthron.es", "password": "$2b$12$UREFwsR0yF8CRqGNK0Lz00H..." }
{ "_id": ObjectId("59b99db4cf9a34dcd7885b7"), "name": "Robert Baratheon", "email": "mark_addy@gameofthron.es", "password": "$2b$12$yGqXL69LZpXA2xVDhuPnS0Zd..." }
{ "_id": ObjectId("59b99db5cf9a34dcd7885b8"), "name": "Jaime Lannister", "email": "nikolaj_coster-waldau@gameofthron.es", "password": "$2b$12$6vz7wiw0.EI5Rilvq1zUc./9..." }
```

Stage 1 \$count

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

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

```
string: 185
```

MongoDB Compass - localhost:27017/Shell

comments users mongosh: localhost:27017 +

```
> _MONGOSH
> db["users"].find()
<
> db.users.countDocuments()
< 0
> use cine
< switched to db cine
> db.users.countDocuments()
< 185
cine>
```

- ¿Cuántos cines existen en el estado de California? - 169 cines

Documents 1.6K Aggregations Schema Indexes 1 Validation

Generate aggregation + Explain Export Run

Untitled SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT

Stage 1 \$match

```
1 {
2   "location.address.state": "CA"
3 }
```

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

```
{ "_id": ObjectId("59a47286cfa9a3a73e51e72e"), "theaterId": 1008, "location": Object }
{ "_id": ObjectId("59a47286cfa9a3a73e51e734"), "theaterId": 1009, "location": Object }
```

Stage 2 \$count

```
1 /**
2  * Provide the field name for the count.
3  */
4 "numero cines California"
```

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

```
numero cines California: 169
```

```

already on db cine
db.theaters.aggregate([
  { $match: { "location.address.state": "CA" } },
  { $count: "total_cines" }
])
{
  total_cines: 169
}
ne>

```

- ¿Cuál fue el primer usuario en registrarse? - **Ned Stark**

The screenshot shows the MongoDB Compass web interface. On the left, the 'CONNECTIONS' panel shows a connection to 'localhost:27017' with a database named 'cine'. The main panel displays an aggregation pipeline with two stages: 'Stage 1' with '\$sort' and 'Stage 2' with '\$limit'. The output preview for Stage 1 shows a sample of 10 documents, with the first document being Ned Stark. The output preview for Stage 2 shows a sample of 1 document, which is Ned Stark.

```

> db.users.aggregate([ { '$sort': { _id: 1 } }, { '$limit': 1 } ])
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu'
}
cine>

```

- ¿Cuántas películas de comedia existen en la base de datos? - 7024

The screenshot shows the MongoDB Compass interface with the Aggregations tab selected. The pipeline consists of two stages:

- Stage 1: \$match**
Query:

```
{  
  genres: "Comedy"  
}
```


Output preview shows 10 documents, including one with `genres: ["Comedy"]`.
- Stage 2: \$count**
Query:

```
{  
  "numero peliculas Comedy": 7024  
}
```


Output preview shows 1 document with the result: `numero peliculas Comedy: 7024`.

```
> db.movies.aggregate([  
  { $match: { genres: "Comedy" } },  
  { $count: "numero peliculas Comedy" }  
)  
< {  
  'numero peliculas Comedy': 7024  
}  
cine>
```

Ejercicio 2

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

Documents 23.5K 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    year: 1932
6  }
7

```

Output preview 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-

_id: ObjectId('573a1392f29313caabcd99a3')
plot: "Junta is hated by the people in the village where she lives, especiall..."
genres: Array (3)
runtime: 85
cast: Array (4)
poster: "https://m.media-amazon.com/images/H/HV5BNTQ1NTHzMT_...
title: "The Blue Light"

```

Stage 2 \$match

```

1 {
2   $or: [
3     { genres: "Drama" },
4     { languages: "French" }
5   ]
6 }
7
8

```

Output preview 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)
  0: "Enrique Rivero"
  1: "Elizabeth Lee Miller"

_id: ObjectId('573a1392f29313caabcd99a3')
plot: "Junta is hated by the people in the village where she lives, especiall..."
genres: Array (3)
  0: "Drama"
  1: "Fantasy"
  2: "Mystery"
runtime: 85
cast: Array (4)

```

Documents 23.5K Aggregations Schema Indexes 1 Validation

[\\$match](#) [\\$match](#) [Edit](#) [Explain](#) [Export](#) [Run](#) [Options](#)

ALL RESULTS Showing 1 - 18 count results [<](#) [>](#) [≡](#) [{}](#)

```

_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/H/HV5BYWY30DE5ZWEtYjIwY0BjA4LTk4ZW..."
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)
runtime: 85
cast: Array (4)
poster: "https://m.media-amazon.com/images/H/HV5BNTQ1NTHzMTQtODIyY500MTxLWE1INT..."
title: "The Blue Light"
fullplot: "Junta is hated by the people in the village where she lives, especiall..."
languages: Array (2)
  released: 1934-05-08T00:00:00.000+00:00
directors: Array (2)
writers: Array (2)
awards: Object
  lastupdated: "2015-08-03 01:02:12.350000000"
  year: 1932
imdb: Object

```

Export Pipeline To Language

My Pipeline

Shell

```

        year: 1932
      }
    },
    {
      $match: {
        $or: [
          {
            genres: "Drama"
          },
          {
            languages: "French"
          }
        ]
      }
    }
  ]

```

☐ Include Import Statements
 ☐ Include Driver Syntax

Exported Pipeline

Python

```

[
  {
    '$match': {
      'year': 1932
    }
  }, {
    '$match': {
      '$or': [
        {
          'genres': 'Drama'
        },
        {
          'languages': 'French'
        }
      ]
    }
  }
]

```

Close

```

_MONGOSH
> use cine
< switched to db cine
> db.movies.aggregate([
  {
    $match:
      {
        year: 1932
      }
    },
    {
      $match: {
        $or: [
          {
            genres: "Drama"
          },
          {
            languages: "French"
          }
        ]
      }
    }
  ]
])
< {
  _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. The artist tries to wipe it',
  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/MV5BYWY3ODU5ZWUyYjI4LTk4ZWYtMzBhZDZlMjY0YTYxXkEyXkFqcGdeQXVyNzI4MDMyMTU0._V1_SX1000_SX677_AL_.jpg',

```

14

Ejercicio 3

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

The screenshot shows the MongoDB Compass interface with an aggregation pipeline for the 'movies' collection. The pipeline consists of three stages, all using the '\$match' operator.

- Stage 1:** The query is `{ countries: "USA" }`. The output preview shows documents with fields like `_id`, `plot`, `genres`, `runtime`, `cast`, `num_mflix_comments`, `title`, and `poster`.
- Stage 2:** The query is `{ year: { $in: [2012, 2013, 2014] } }`. The output preview shows documents with fields like `_id`, `fullplot`, `imdb`, `year`, `plot`, and `genres`.
- Stage 3:** The query is `{ "awards.wins": { $in: [5, 6, 7, 8, 9] } }`. The output preview shows documents with fields like `_id`, `fullplot`, `imdb`, `year`, `plot`, and `genres`.

The screenshot shows the 'ALL RESULTS' tab in MongoDB Compass, displaying the results of the aggregation pipeline. The results are shown as a list of documents, with the first two documents visible.

- Document 1:** `{ "_id": "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 08:10:51.747000000", "languages": [], "writers": [], "type": "movie", "tomatoes": { "poster": "https://m.media-amazon.com/images/M/MV5BODVwNDYxNDk1N15BNl5BanBnXkFtZT...", "num_mflix_comments": 1, "released": "2013-12-25T00:00:00.000+00:00", "awards": { }, "countries": [], "cast": [], "directors": [], "runtime": 114 } }`
- Document 2:** `{ "_id": "573a13b5f29313caabd45772", "plot": "After their cave is destroyed, a caveman family must trek through an u...", "genres": [], "runtime": 98, "metacritic": 55, "rated": "PG", "cast": [], "poster": "https://m.media-amazon.com/images/M/MV5BNTcyOTc2OTA1M15BNl5BanBnXkFtZT...", "title": "The Croods", "fullplot": "The Croods is a prehistoric comedy adventure that follows the world's ...", "languages": [], "released": "2013-03-22T00:00:00.000+00:00", "directors": [] }`

×

Export

Aggregation on cine.movies

Export results from the aggregation below

```

db.getCollection('movies').aggregate(
  [
    { $match: { countries: 'USA' } },
    {
      $match: {
        year: { $in: [2012, 2013, 2014] }
      }
    },
    {
      $match: {
        'awards.wins': { $in: [5, 6, 7, 8, 9] }
      }
    }
  ]
)

```

Export File Type

JSON

CSV

> Advanced JSON Format

Cancel

Export...

```

>_MONGOOSH
> use cine
< switched to db cine
> db.getCollection('movies').aggregate(
  [
    { $match: { countries: 'USA' } },
    {
      $match: {
        year: { $in: [2012, 2013, 2014] }
      }
    },
    {
      $match: {
        'awards.wins': { $in: [5, 6, 7, 8, 9] }
      }
    }
  ],
  { maxTimeMS: 60000, allowDiskUse: true }
);
< [
  {
    _id: ObjectId('573a13acf29313caabd29366'),
    fullplot: "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen years for the magazine and has a tedious li
    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 global journey that turns into an ad
    genres: [
      'Adventure',
      'Comedy',
      'Drama'
    ],
    rated: 'PG',
    runtime: 115
  }
]

```


Nivel 2

Ejercicio 1

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

Ejercicio 2

¿Cuántos cines existen en cada código postal situados dentro del estado Washington DC (DC)?

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.

Ejercicio 2

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