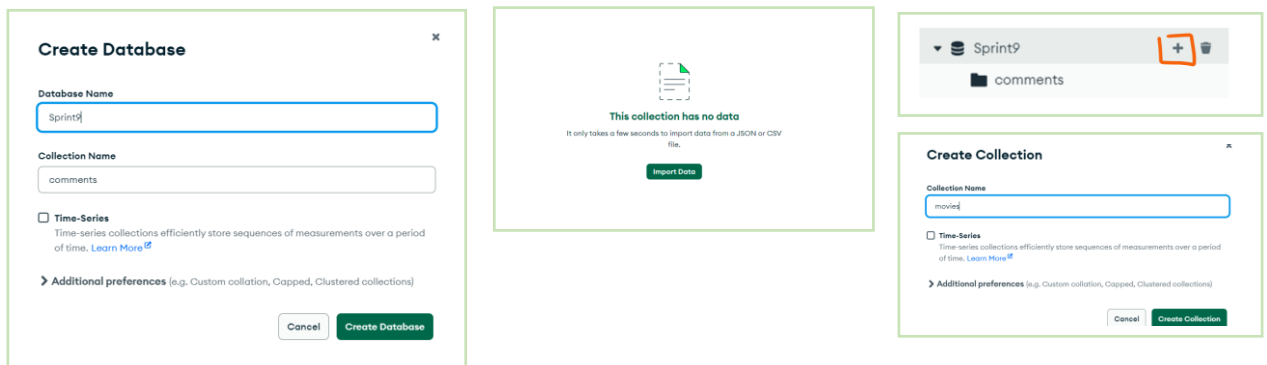


NIVEL 1

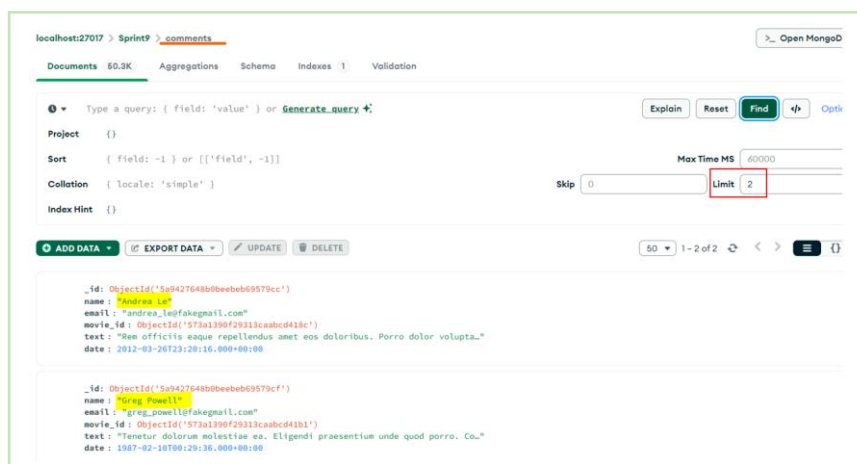
EJERCICIO 1

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

Primero, una vez en MongoDB Compass, he creado una nueva base de datos, que la he llamado Sprint9, y ponemos también el nombre de la primera colección que luego importaremos. Luego importo el archivo comments (archivo json) y el mismo procedimiento aplico para el resto de los archivos, para luego ir haciendo las consultas necesarias. Adjunto alguna de las imágenes de la secuencia que he seguido.



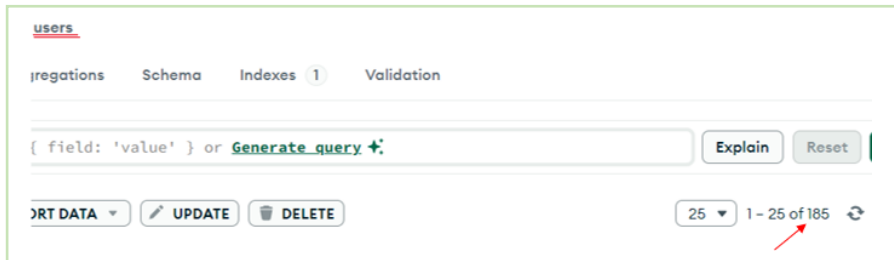
- *Muestra los 2 primeros comentarios que hay en la base de datos.*



Para este primer ejercicio, lo hice directamente desde la pantalla principal, fui a la opción **limit**, establecí el valor en 2 y con eso ya me salen los primeros dos comentarios de la base de datos. Lo hicieron **Andrea Le y Greg Powell**.

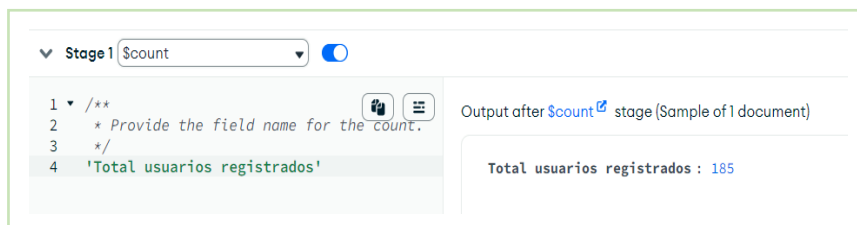
- ¿Cuántos usuarios tenemos registrados?

Respecto a este ejercicio, como primera alternativa, puedo ir a la tabla users, y en el extremo derecho me aparece el nro total de usuarios, o bien



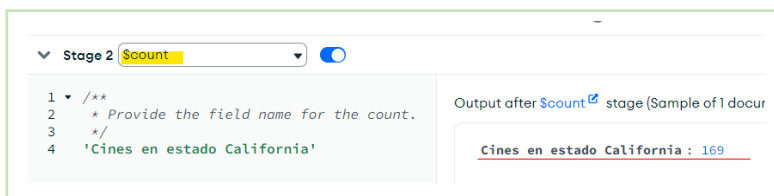
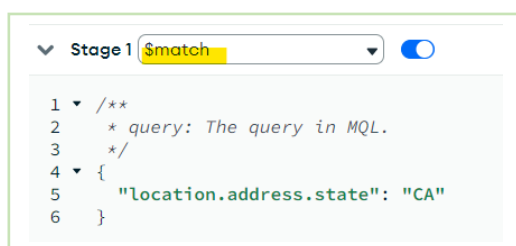
O bien, en el panel de **Aggregations**, en el primer Stage1, pongo la opción de **\$count**.

En total son **185 usuarios**.



- ¿Cuántos cines hay en el estado de California?

Para este ejercicio también lo sigo realizando por el panel de **Aggregations**, que me permite ir haciéndolo por etapas, donde cada etapa va recibiendo la salida de datos de la etapa anterior.



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

- ¿Cuál fue el primer usuario/a en registrarse?

```

1 [
2   {
3     $limit:
4     /**
5      * Provide the number of documents to limit.
6      */
7     1
8   }
9 ]

```

Stage 1 **\$limit**

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

```

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

```

```

_id: ObjectId('59b99db4cfa9a34dc7885b6')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$UREFwsRUoyF0CRqGNK0Lz08HM/jLh_

```

El primer usuario en registrarse fue Ned Stark. Seleccioné la tabla users y utilicé el operador **\$limit**: 1 para mostrar el primer usuario.

- ¿Cuántas películas de comedia hay en nuestra base de datos?

Para este caso en primer lugar he seleccionador el operador **\$match**, tomé como género Comedy, y luego utilicé el operador **\$count**, para determinar la cantidad de películas comedias en la base de datos, en total son **7024**

Stage 1 **\$match**

```

1 /**
2  * query: The query in MQL.
3  */
4  {
5    "genres": "Comedy"
6  }

```

Stage 2 **\$count**

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

```

1 /**
2  * Provide the field name for the count.
3  */
4  'Total películas de Comedia'

```

Total películas de Comedia: 7024

EJERCICIO 2

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

```

{
  $match: {
    year: 1932,
    $or: [
      {
        genres: "Drama"
      },
      {
        languages: "French"
      }
    ]
  },
  $count:
    /**
     * Provide the field name for the count.
     */
    "Total películas de 1932 que son Drama o estan en frances"
}

```

Total películas de 1932 que son Drama o estan : 18
en frances

Para este ejercicio, en primer lugar he utilizado el operador **\$match** para cerrarlo en películas de 1932. Y luego una vez filtrado por películas de 1932, utilice el operador **\$or** para que me buscara género Drama o de idioma francés. Y por último, **\$count** para contar cuántas películas eran, un total de 18.

EJERCICIO 3

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

```
[
  {
    $match: {
      countries: "USA",
      $and: [
        {
          year: {
            $gte: 2012,
            $lte: 2014
          },
          "awards.wins": {
            $gte: 5,
            $lte: 9
          }
        }
      ]
    },
    $count:
    /**
     * Provide the field name for the count.
     */
    "Películas estadounidenses producidas entre 2012 y 2014, y
  ]
```

Películas estadounidenses producidas entre 2012 y 2014, y que tienen entre 5 y 9 premios : 166

Para este ejercicio, en primer lugar he utilizado el operador **\$match** para filtrar solo aquellas películas estadounidenses. Luego el operador **\$and**, acompañados del **\$gte** y **\$lte** para delimitar tanto el año de la película como la cantidad de premios recibidos.

Las películas que cumplen con estas pautas son **166**

NIVEL 2

EJERCICIO 1

- ¿Cuenta cuántos comentarios escribe un usuario/a que utiliza "GAMEOFTHRON.ES" como dominio de correo electrónico?

Para resolver este ejercicio he comenzado por utilizar **\$match** en conjunto con el operador **\$regex**, lo que permite hacer búsqueda de un tramo de texto, es decir, en este caso que tengan el texto gameofthron.es en el campo email. Y por último una vez identificados todos aquellos que terminan con este dominio, aplicar el operador **\$count**.

```
▼ Stage 1 $match
1 ▼ /**
2   * query: The query in MQL.
3   */
4 ▼ {
5 ▼   email: {
6     $regex: "gameofthron.es"
7   }
8 }
```

▼ Stage 2 \$count

```
1 ▼ /**
2   * Provide the field name for the count.
3   */
4 Cantidad de comentarios escritos por usuar
```

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

Cantidad de comentarios escritos por usuarios con dominio gameofthron	: <u>22841</u>
-----------------------------------------------------------------------	----------------

EJERCICIO 2

- ¿Cuántos cines hay en cada código postal situados dentro del estado Washington D. C. (DC)?

Para resolver este ejercicio, he comenzado por **\$match** para acotar a aquellos que están dentro del estado de Washington (he interpretado que corresponde a las siglas DC “District Columbia”). Luego he utilizado el módulo **\$group**, seguido de **_id** (este id no hace referencia al id de la tabla, sino el que campo que definiré como criterio de agrupación, en este caso “**\$location.address.zipcode**” para poder agrupar la cantidad por zipcode y con el módulo **\$sum** poder tener el total de cines de cada zipcode del estado de Washginton.

```
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      Cantidad_Cines: {
20        $sum: 1
21      }
22    }
23  }
24 }
```

Sample of 3 documents

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

{
  "_id": "20010",
  "Cantidad_Cines": 1
}

{
  "_id": "20016",
  "Cantidad_Cines": 1
}
```

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

Para resolver este ejercicio, comencé por la opción de **Aggregations**, y una vez ahí comienzo a construir el código. En primer lugar he utilizado como opción de Stage 1, he puesto **\$match**,

sin embargo para lo que es ya el resto del código, me he sentido más cómodo continuándolo desde la opción **Text** (esto

```
1 {
2   {
3     $match: {
4       "directors.0": "John Landis"
5     },
6   }
7 }
```

también lo he aplicado para ejercicios anteriores). El resultado final de este ejercicio son **4 películas**. Para llegar a ese resultado me he valido también del **\$and** y del **\$gte**, **\$lte** para marcar los limites deseados.

oalhost:27017 > Spring > movies

Documents 25.8K Aggregations Schema Indexes 1 Validation

Generate aggregation + Explain Export

Preview - modified SAVE + CREATE NEW EXPORT TO LANGUAGE

PREVIEW STAGES TEXT

EJERCICIO 2

- *Muestra en un mapa la ubicación de todos los cines de la base de datos.*

Para este ejercicio he ido a la opción **Schema**, y dentro del campo location, está la posibilidad de ver la ubicación geográfica de cada uno de los cines. Esto es posible ya que dentro del campo location están almacenadas las coordenadas (longitud y latitud)

```
Output after $match stage (Sample of 4 documents)
```

```
{
  "_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"
  },
  "location": {
    "address": {
      "document": {
        "_id": ObjectId("573a1397f29313caabce6d94"),
        "plot": "Jake B...",
        "genres": Array,
        "runtime": 133,
        "rated": "R",
        "cast": Array,
        "num_mflix_comments": 1,
        "poster": "http:..."
      }
    }
  }
}
```

