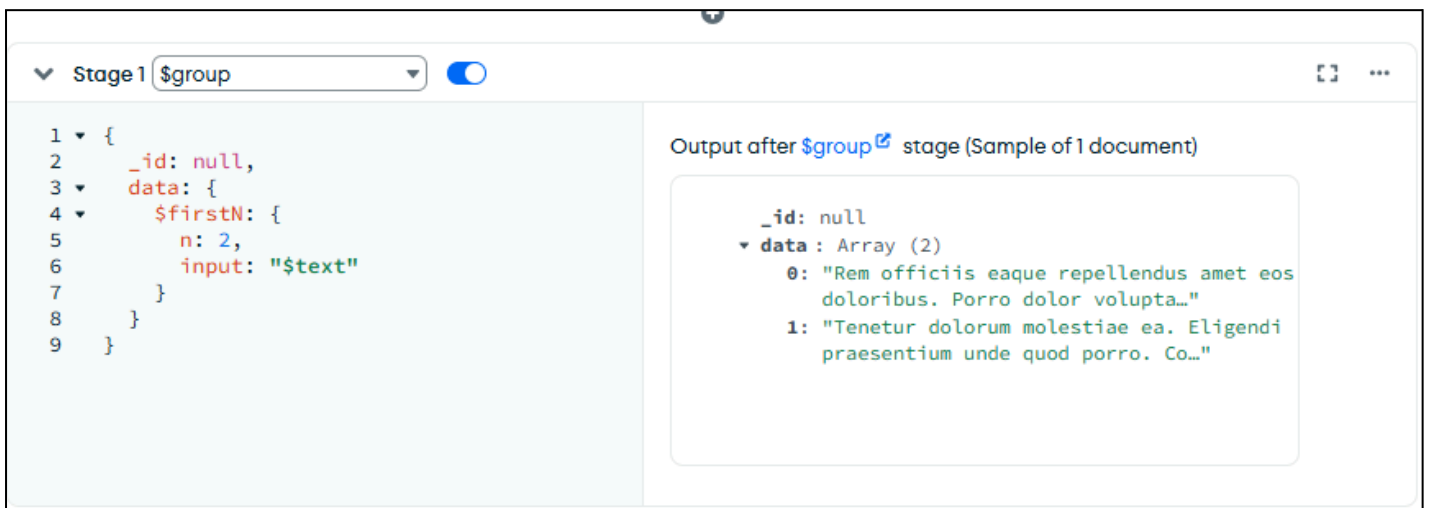


Nivell 1

- Exercici 1

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



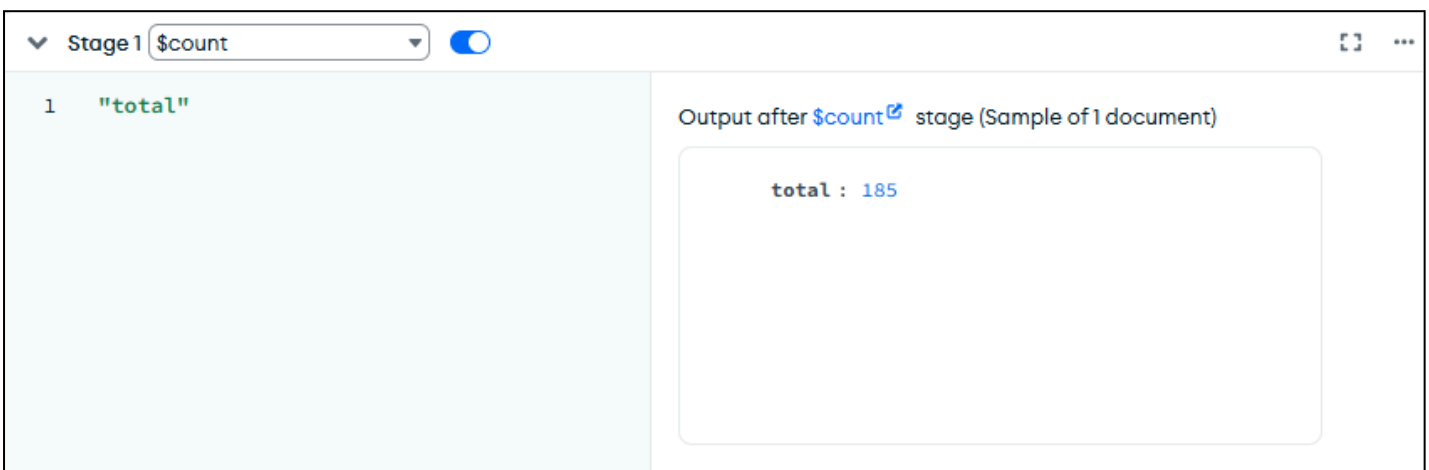
The screenshot shows a MongoDB aggregation pipeline stage configuration. The stage is named "Stage1" and uses the "\$group" operator. The configuration is as follows:

```
1 {  
2   _id: null,  
3   data: {  
4     $firstN: {  
5       n: 2,  
6       input: "$text"  
7     }  
8   }  
9 }
```

The output after the "\$group" stage (Sample of 1 document) is shown as follows:

```
{  
  "_id": null,  
  "data": Array (2)  
    0: "Rem officiiis eaque repellendus amet eos  
        doloribus. Porro dolor volupta..."  
    1: "Tenetur dolorum molestiae ea. Eligendi  
        praesentium unde quod porro. Co..."  
}
```

- Quants usuaris tenim registrats?



The screenshot shows a MongoDB aggregation pipeline stage configuration. The stage is named "Stage1" and uses the "\$count" operator. The configuration is as follows:

```
1 "total"
```

The output after the "\$count" stage (Sample of 1 document) is shown as follows:

```
{  
  "total": 185  
}
```

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

▼ Stage 1

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

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

```
_id: ObjectId('59a47286cfa9a3a73e51e72d')
theaterId: 1003
location: Object
```

▼ Stage 2

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

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

```
total: 36
```

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

▼ Stage 1

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

Output after [\\$sort](#) stage (Sample of 10 documents)

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

▼ Stage 2

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

Output after [\\$limit](#) stage (Sample of 1 document)

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

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

▼ Stage1

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

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

```
_id: ObjectId('573a1390f29313caabcd4803')
plot: "Cartoon figures announce, via comic
strip balloons, that they will mov..."
genres: Array (3)
runtime: 7
cast: Array (1)
num_mflix_comments: 1
poster: "https://m.media-
amazon.com/images/M/MV5BYzg2NjNhNT..."
```

+

▼ Stage2

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

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

```
total: 7024
```

- Exercici 2

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

▼ Stage1

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   year: 1932,
6   $or: [
7     {
8       genres: "Drama"
9     },
10    {
11      languages: "French"
12    }
13  ]
14 }
```

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-
```

- 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.

The screenshot shows the MongoDB Compass interface. On the left, the query editor displays a query for movies with 5-9 awards, produced between 2012 and 2014, in the USA. The query is as follows:

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   countries: "USA",
6   "awards.wins": {
7     $gte: 5,
8     $lte: 9
9   },
10  year: {
11    $gte: 2012,
12    $lte: 2014
13  }
14 }
```

On the right, the output after the \$match stage is shown (Sample of 10 documents). The output is a JSON document representing a movie:

```
{
  "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)
}
```

Nivell 2

- Exercici 1

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

The screenshot shows the MongoDB Compass interface with a two-stage query. Stage 1 is a \$lookup stage that joins the 'comments' collection with the 'users' collection based on the email field. Stage 2 is a \$match stage that filters for users whose email domain is 'GAMEOFTHRON.ES'. The query is as follows:

```
1 /**
2  * from: The target collection.
3  * localField: The local join field.
4  * foreignField: The target join field.
5  * as: The name for the results.
6  * pipeline: Optional pipeline to run on the target collection.
7  * let: Optional variables to use in the pipeline.
8  */
9 {
10   from: "comments",
11   localField: "email",
12   foreignField: "email",
13   as: "result"
14 }
15
```

Stage 2 is a \$match stage with the following query:

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   email: {
6     $regex: "@GAMEOFTHRON\\.ES$",
7     $options: "i"
8   }
9 }
```

The output after the \$lookup stage is shown (Sample of 10 documents). The output is a JSON document representing a user with a list of comments:

```
{
  "_id": ObjectId('59b99db4cfa9a34dcd7885b6'),
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM...",
  "result": Array (273)
}
```

The output after the \$match stage is shown (Sample of 10 documents). The output is a JSON document representing a user whose email domain is 'GAMEOFTHRON.ES':

```
{
  "_id": ObjectId('59b99db4cfa9a34dcd7885b6'),
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM...",
  "result": Array (273)
}
```

Stage 3

\$project

1

2

3

4

5

6

7

8

9

/**

* specifications: The fields to

* include or exclude.

*/

{

numItems: {

\$size: "\$result"

}

}

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

_id: ObjectId('59b99db4cfa9a34dcd7885b6')

numItems: 273

_id: ObjectId('59b99db4cfa9a34dcd7885b7')

numItems: 256

- Exercici 2

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

Stage 1

\$match

1

2

3

4

5

6

/**

* query: The query in MQL.

*/

{

"location.address.state": "DC"

}

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

_id: ObjectId('59a47286cfa9a3a73e51e785')

theaterId: 1092

location: Object

_id: ObjectId('59a47287cfa9a3a73e51ec33')

theaterId: 801

location: Object

+

Stage 2

\$group

1

2

3

4

5

6

7

8

9

10

/**

* _id: The id of the group.

* fieldN: The first field name.

*/

{

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

count: {

\$sum: 1

}

}

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

_id: "20010"

count: 1

_id: "20002"

count: 1

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.

The screenshot shows a data visualization interface. On the left, a query is written in MQL:

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   directors: "John Landis",
6   "imdb.rating": {
7     $gte: 7.5,
8     $lte: 8
9   }
10 }
```

 On the right, the output after the match stage is shown as a sample of 4 documents. Two documents are visible:

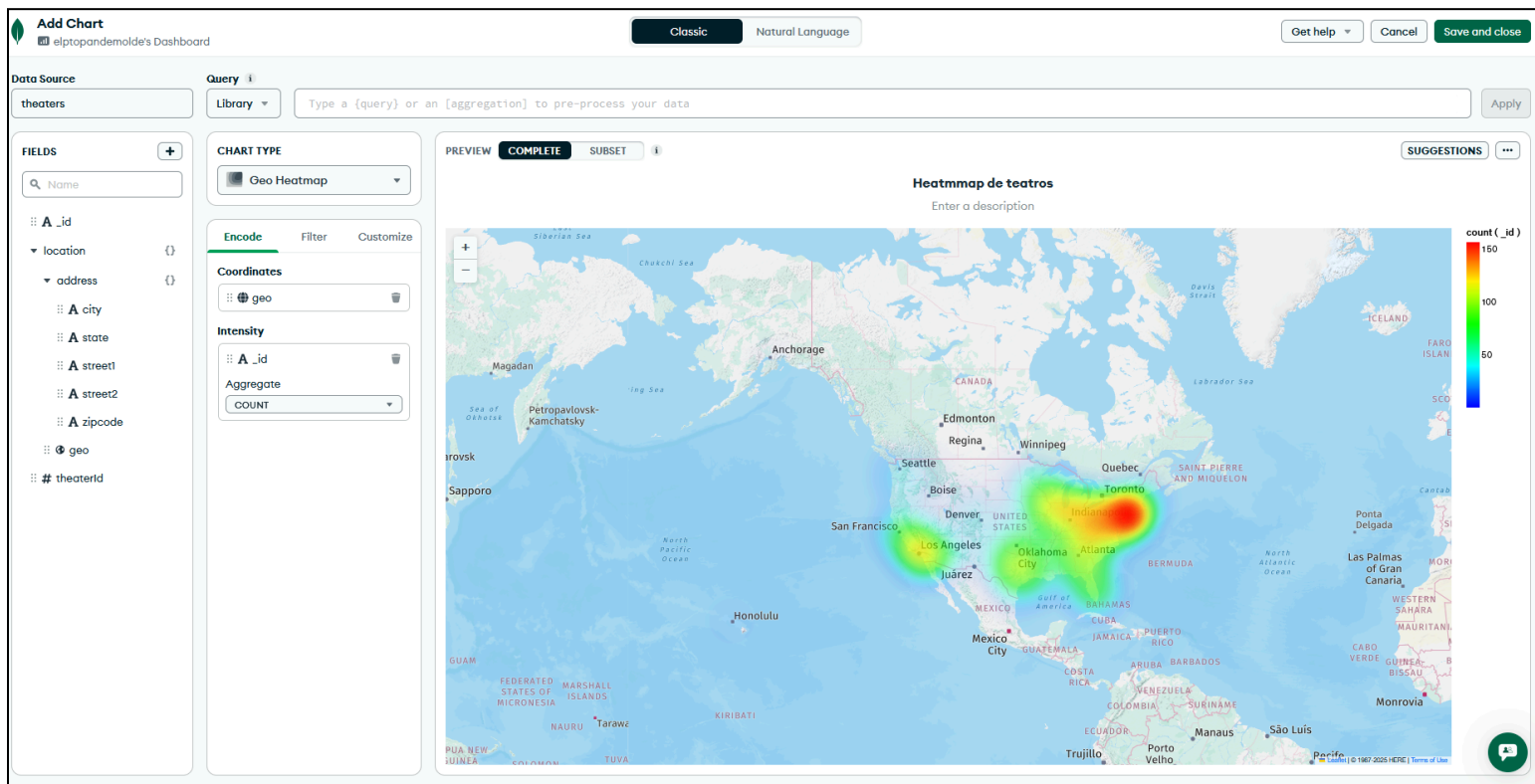
```
{
  "_id": ObjectId('573a1397f29313caabce6d94'),
  "fullplot": "Faber College has one frat house so disreputable it will take anyone. ...",
  "imdb": Object,
  "year": 1978,
  "plot": "At a 1962 college, Dean Vernon Wormer is determined to expel the entire..."
}
```

 and

```
{
  "_id": ObjectId('573a1397f29313caabce76f7'),
  "plot": "Jake Blues, just out from prison, puts together his old band to save t...",
  "genres": Array (3),
  "runtime": 133,
  "rated": "R",
  "cast": Array (4),
  "num_mflix_comments": 1
}
```

- Exercici 2

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



Usando atlas charts.