

Queries with MongoDB

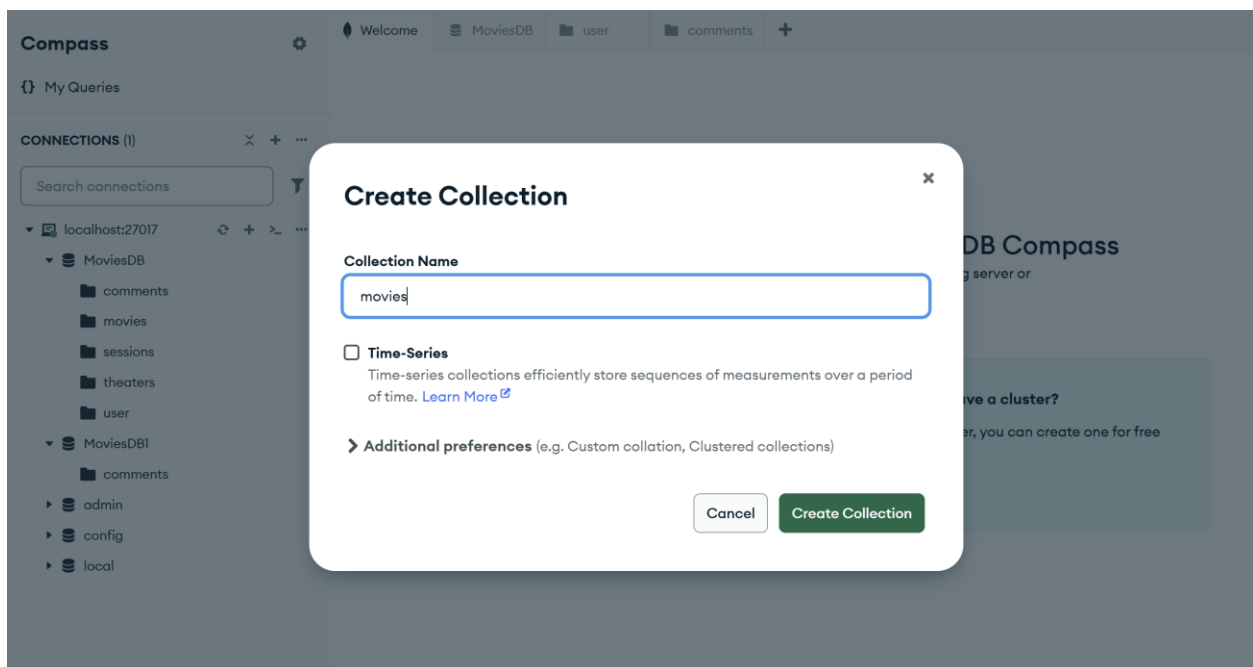
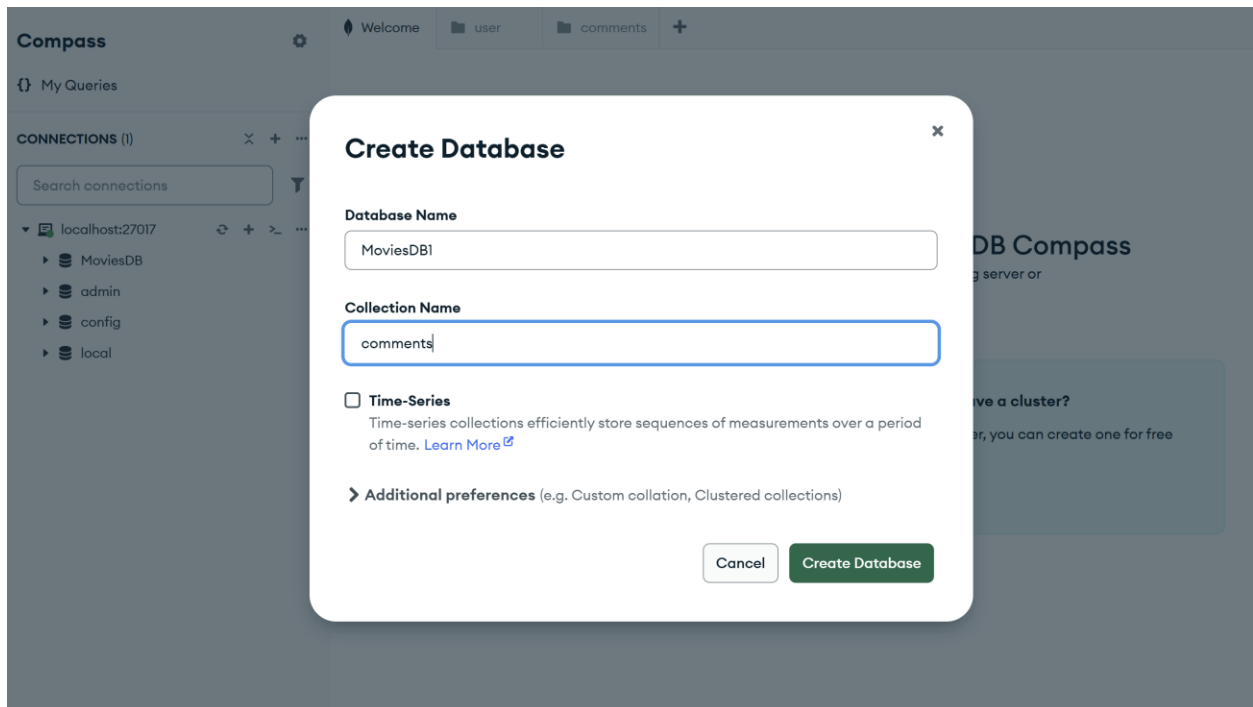
We will work with a database that contains collections related to a film entertainment application:

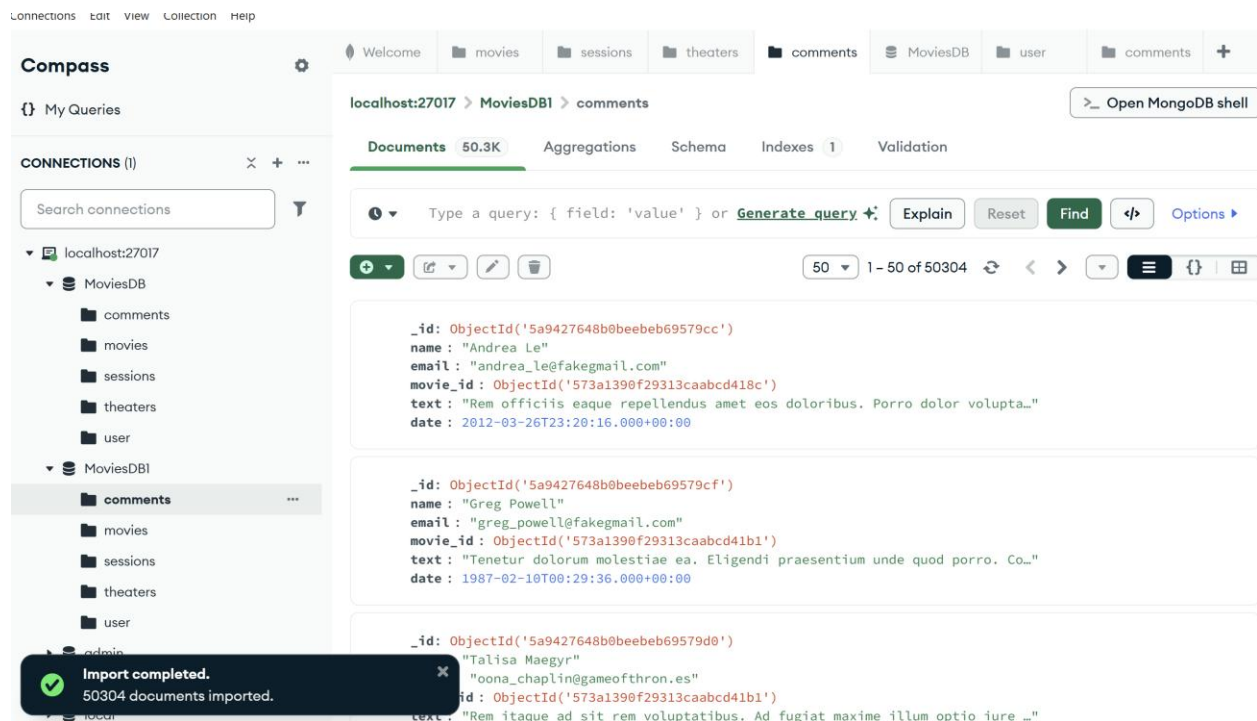
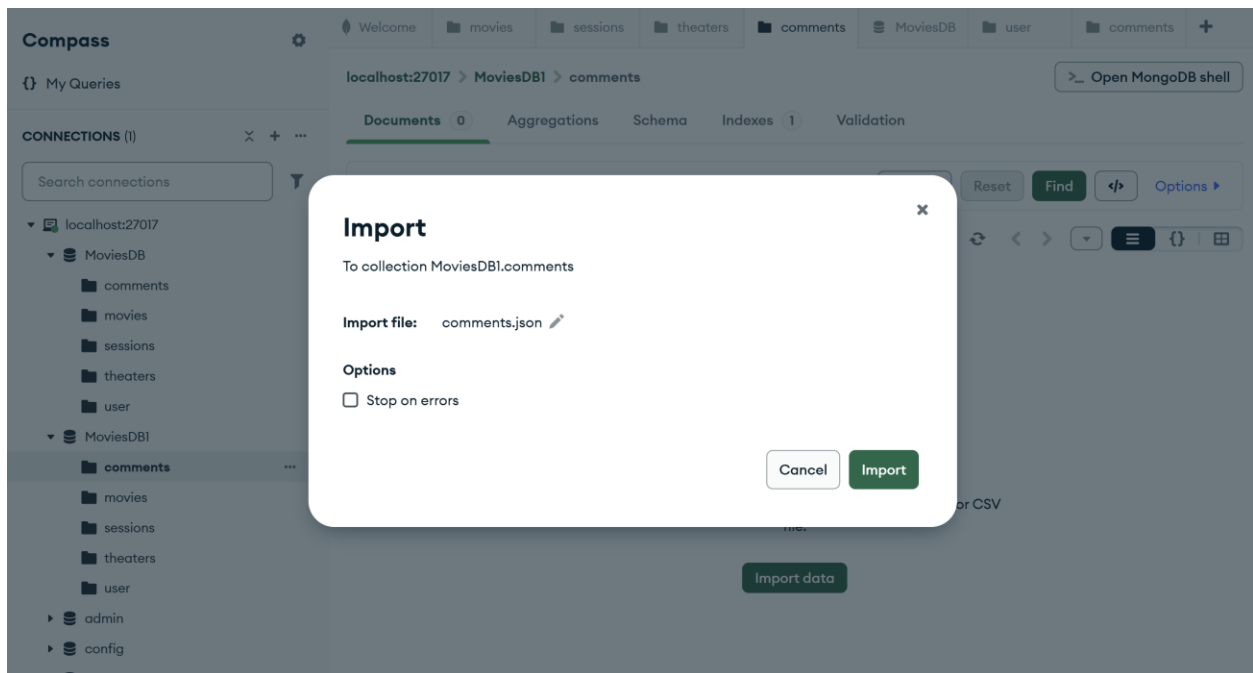
- users: Stores user information, including names, emails, and encrypted passwords.
- theatres: Contains cinema data, such as ID, location (address and geographical coordinates).
- sessions: Stores user sessions, including user IDs and JWT tokens for authentication.
- movies: Includes movie details, such as plot, genres, duration, cast, comments, release year, directors, ratings and awards.
- comments: Stores user comments about movies, with information about the author of the comment, movie ID, comment text, and date.

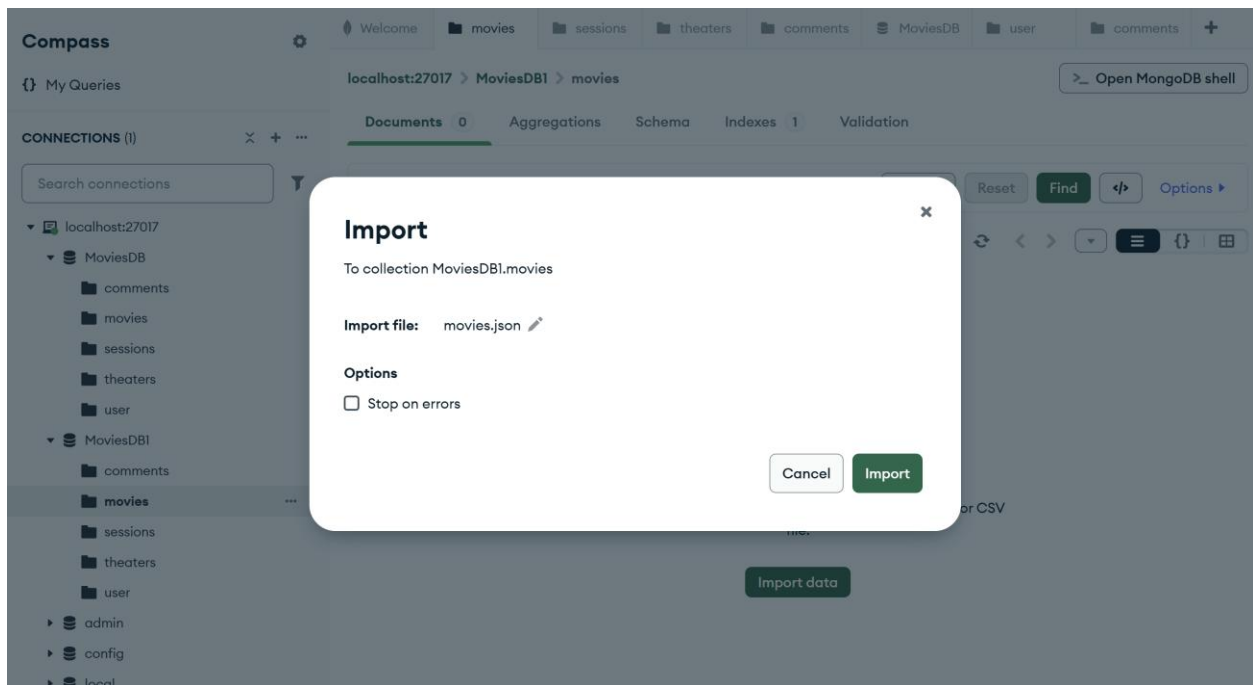
You will carry out some queries requested by the client, who is measuring whether or not you will be able to take charge of the analytical part of the project linked to their database.

Level 1

Create a database with MongoDB using the attached files as collections.







Exercise 1

- Shows the first 2 comments in the database.

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

Compass

My Queries

CONNECTIONS (1)

Search connections

localhost:27017

MoviesDB

comments

movies

sessions

theaters

user

MoviesDB1

comments

movies

sessions

theaters

user

admin

config

local

localhost:27017 > MoviesDB1 > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

Your pipeline is currently empty. Need help getting started?

Explain Export Run Options

Untitled

PREVIEW

50304 Documents in the collection

Preview of documents

```
_id: ObjectId('5a9427648b0beebe69579cc')
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd41...')
text: "Rem officiis eaque repellendus amet
eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
```

```
_id: ObjectId('5a9427648b0beebe69579cf')
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd41...')
text: "Tenetur dolorum molestiae ea.
Eligendi praesentium unde quod porro.
Co..."
date: 1987-02-10T00:29:36.000+00:00
```

+ Add stage

Learn more about aggregation pipeline stages

Compass

My Queries

CONNECTIONS (1)

Search connections

localhost:27017

MoviesDB

comments

movies

sessions

theaters

user

MoviesDB1

comments

movies

sessions

theaters

user

admin

config

local

localhost:27017 > MoviesDB1 > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

\$limit

Explain Export Run Options

Untitled - modified

PREVIEW

Stage 1 \$limit

1 2

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

```
_id: ObjectId('5a9427648b0beebe69579cc')
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd418...')
text: "Rem officiis eaque repellendus amet
eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
```

Show details

localhost:27017 > MoviesDB1 > comments

>_ Open MongoDB shell

Documents 50.3K

Aggregations

Schema

Indexes 1

Validation



\$limit



Edit



Explain

Export

Run

Options >

ALL RESULTS

Showing 1 - 2 count results



```
_id: ObjectId('5a9427648b0beeb69579cc')
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd418c')
text: "Rem officiiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
```



```
_id: ObjectId('5a9427648b0beeb69579cf')
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd41b1')
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00
```

- How many users do we have registered?

```
[
  {
    $count: "count"
  }
]
```

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

localhost:27017

- MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
- MoviesDB1
 - comments
 - movies
 - sessions
 - theaters
- user
 - admin
 - config
 - local

localhost:27017 > MoviesDB1 > user

Documents 185 Aggregations Schema Indexes 1 Validation

\$count Explain Export Run Options

Untitled - modified CREATE NEW

PREVIEW

Stage 1 \$count

1
2 'count'

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

count : 185

+ Add stage

[Learn more about aggregation pipeline stages](#)

Connections Edit View Collection Help

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

localhost:27017

- MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
- MoviesDB1
 - comments
 - movies
 - sessions
 - theaters
- user
 - admin
 - config
 - local

localhost:27017 > MoviesDB1 > user

Documents 185 Aggregations Schema Indexes 1 Validation

\$count Edit Explain Export Run Options

ALL RESULTS Showing 1 - 1 count results

count : 185

- How many movie theaters are there in the state of California?

```
[
  {
    $match: {
      "location.address.state": "CA"
    }
  },
  {
    $count: "TheaterID"
  }
]
```

The screenshot shows the MongoDB Compass web interface. On the left, the 'CONNECTIONS' panel shows a connection to 'localhost:27017' with a database named 'MoviesDB'. The 'theaters' collection is selected. The main panel shows the 'Aggregations' tab with a single stage named '\$match'. The query is:

```
1
2 {
3   "location.address.state": "CA"
4 }
```

 The 'Output after \$match stage (Sample of 10 documents)' shows a document with fields:

```
_id: ObjectId('59a47286cfa9a3a73e51e72e')
theaterId: 1808
location: Object
```


Compass

My Queries

CONNECTIONS (1)

Search connections

localhost:27017

- MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
- MoviesDB1
 - comments
 - movies
 - sessions
 - theaters
 - user
- admin
- config
- local

localhost:27017 > MoviesDB1 > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

\$match \$count Explain Export Run Options

Untitled - modified + CREATE NEW

PREVIEW

Stage 2 \$count

1
2 'TheaterID'

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

TheaterID : 169

localhost:27017 > MoviesDB1 > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

\$match \$count Edit Explain Export Run Options

ALL RESULTS Showing 1 - 1 count results

TheaterID : 169

- Who was the first user to register?

```
[
{
  $sort: {
    registered: 1
  }
},
{
  $limit: 1
}
]
```

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS (1)' panel shows a connection to 'localhost:27017' with a database named 'MoviesDB'. The 'user' collection is selected. The main panel shows the 'Aggregations' tab with a pipeline consisting of two stages: '\$sort' and '\$limit: 1'. The output of the '\$sort' stage is displayed, showing a sample of 10 documents. The first document is:

```
{
  "_id": "ObjectId('59b99db8cfa9a34dcd7885be')",
  "name": "Viserys Targaryen",
  "email": "harry_lloyd@gameofthron.es",
  "password": "$2b$12$cwVmU4DyuQxgwpdrVJhaudzb..."
}
```

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

localhost:27017

- MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
- MoviesDB1
 - comments
 - movies
 - sessions
 - theaters
 - user

localhost:27017 > MoviesDB1 > user

Documents 185 Aggregations Schema Indexes 1 Validation

\$sort \$limit Explain Export Run Options

Untitled - modified CREATE NEW

PREVIEW

Stage 2 \$limit

1

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

```
{
  "_id": ObjectId('59b99db4cf9a34dcd7885b6'),
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNk0Lz00HM..."
}
```

Add stage

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

localhost:27017

- MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
- MoviesDB1
 - comments
 - movies
 - sessions
 - theaters
 - user

localhost:27017 > MoviesDB1 > user

Documents 185 Aggregations Schema Indexes 1 Validation

\$sort \$limit Edit Explain Export Run Options

ALL RESULTS

Showing 1 - 1 count results

```
{
  "_id": ObjectId('59b99db4cf9a34dcd7885b6'),
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNk0Lz00HM/jLhgUCNNI39RJAqMUQ74cr1J1Vu"
}
```

- How many comedy films are in our database?

```
[
{
  $match: {
    genres: "Comedy"
  },
{
  $count:
    "Total"
}
]
```

The screenshot shows the MongoDB Compass web interface. On the left, the 'CONNECTIONS (1)' sidebar shows a connection to 'localhost:27017' with a database named 'MoviesDB'. The 'movies' collection is selected. The main panel displays the 'Aggregations' tab for the 'movies' collection. A single stage named '\$match' is added to the pipeline. The 'tomatoes' field is visible in the document preview. The 'Stage 1 \$match' configuration shows a filter for 'genres' equal to 'Comedy'. The 'Output after \$match stage (Sample of 10 documents)' shows a document with fields: '_id', 'plot', 'genres', 'runtime', 'cast', 'num_mflix_comments', and 'poster'.

```
1 {
2   "genres" : "Comedy"
3 }
```

```
{
  "_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..."
}
```

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

localhost:27017

MoviesDB

comments

movies

sessions

theaters

user

MoviesDB1

comments

movies

sessions

theaters

user

admin

config

local

Welcome

movies

sessions

theaters

user

comm...

Movies...

user

comm...

localhost:27017 > MoviesDB1 > movies

> Open MongoDB shell

Documents 23.5K Aggregations Schema Indexes 1 Validation

\$match \$count

Explain Export Run Options

Untitled - modified

+ CREATE NEW

PREVIEW

Stage 2 \$count

1

2 "Total"

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

Total : 7624

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

localhost:27017

MoviesDB

comments

movies

sessions

theaters

user

MoviesDB1

comments

movies

sessions

theaters

user

admin

config

local

Welcome

movies

sessions

theaters

user

comm...

Movies...

user

comm...

localhost:27017 > MoviesDB1 > movies

> Open MongoDB shell

Documents 23.5K Aggregations Schema Indexes 1 Validation

\$match \$count Edit

Explain Export Run Options

ALL RESULTS

Showing 1 - 1 count results

Total : 7624

Exercise 2

Show me all the documents for films produced in 1932, but the genre is drama or they are in French.

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

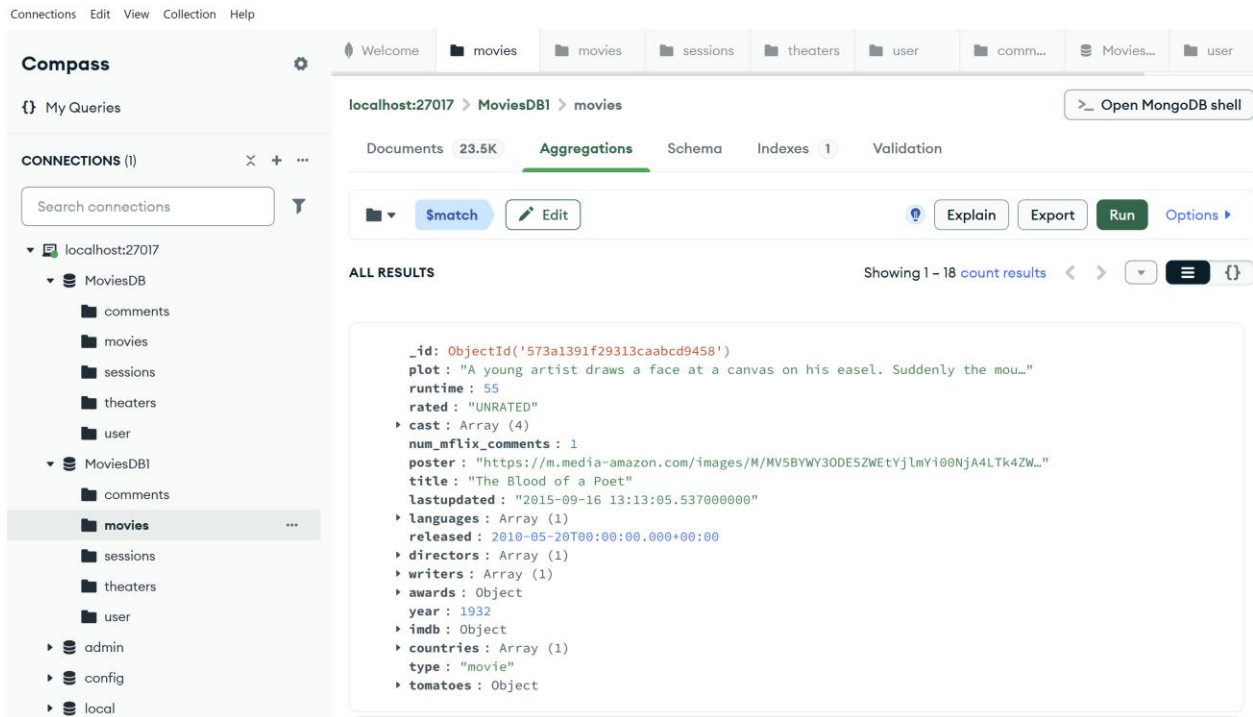
The screenshot shows the MongoDB Compass application interface. On the left, the 'Connections' panel shows a connection to 'localhost:27017' with a database named 'MoviesDB'. The 'collections' list on the left includes 'comments', 'movies', 'sessions', 'theaters', and 'user'. The 'movies' collection is selected. The main panel displays the 'Aggregations' tab for the 'movies' collection. The aggregation pipeline is defined as follows:

```
1 {
2   year: 1932,
3   $or: [
4     {
5       genres: "Drama"
6     },
7     {
8       languages: "French"
9     }
10  ]
11 }
```

The 'Output after \$match stage (Sample of 10 documents)' panel shows a sample of the 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-"
}
```

At the bottom, there is a '+ Add stage' button and a link to 'Learn more about aggregation pipeline stages'.



Exercise 3

Show me all documents of American films with between 5 and 9 awards that were produced between 2012 and 2014.

```

[
  {
    $match: {
      countries: "USA",
      "awards.wins": {
        $gte: 5,
        $lte: 9
      }
    }
  }
]

```

```

},
year: {
  $gte: 2012,
  $lte: 2014
}
}
}
]

```

The screenshot shows the MongoDB Compass application interface. The left sidebar displays the 'Connections' list with 'localhost:27017' selected, and the 'Collections' list for 'MoviesDB' with 'movies' highlighted. The main panel shows the 'Aggregations' tab for the 'movies' collection. A single aggregation stage named '\$match' is configured with the following query:

```

1 {
2   countries: "USA",
3   awards.wins: {
4     $gte: 5,
5     $lte: 9
6   },
7   year: {
8     $gte: 2012,
9     $lte: 2014
10  }
11 }

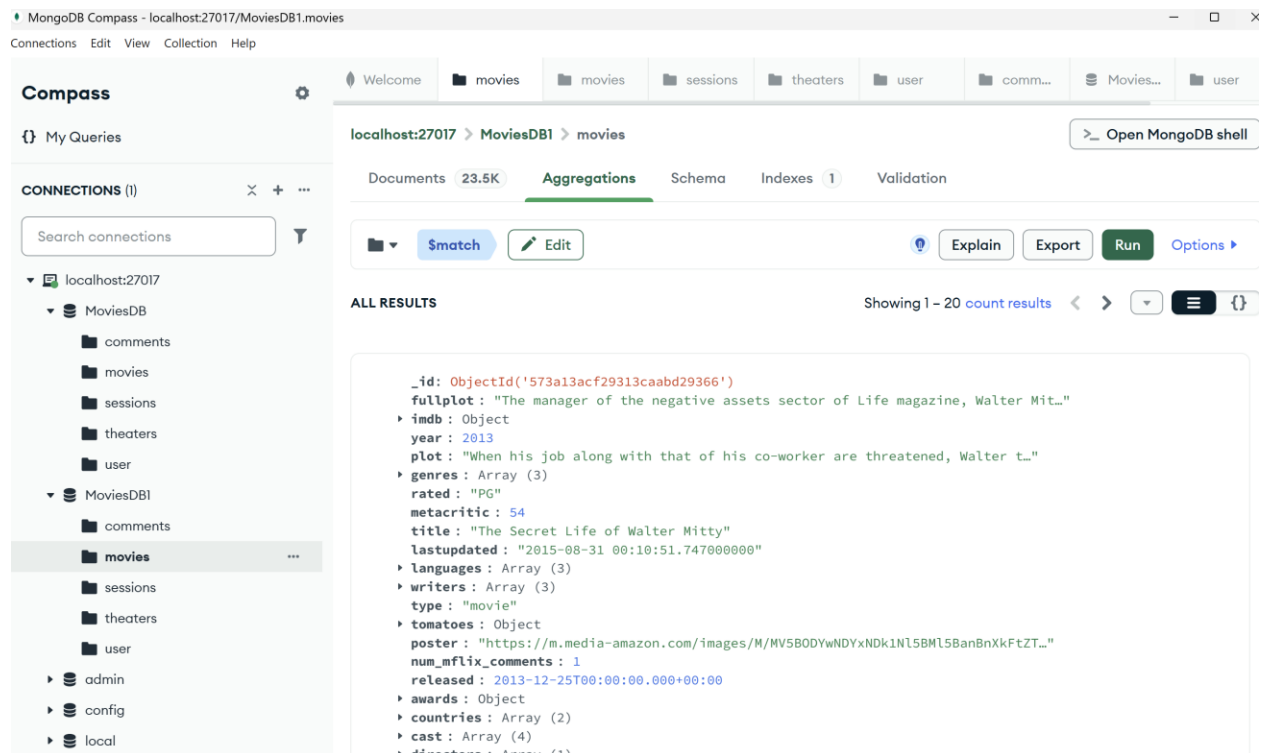
```

The 'Output after \$match stage (Sample of 10 documents)' panel shows a sample document:

```

{
  "_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)
}

```

Level 2

Exercise 1

Count how many comments a user writes who uses "GAMEOFTHRON.ES" as their email domain.

```
[
{
  $match: {
    email: {
      $regex: "@gameofthron.es",
      $options: "i"
    }
  }
}
```

```

    }
  },
  {
    $count: "Email game of thorns"
  }
]

```

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' panel shows a connection to 'localhost:27017' with a database named 'MoviesDB'. The 'collections' list on the left includes 'comments', 'movies', 'sessions', 'theaters', and 'user'. The 'comments' collection is selected.

The main panel shows the 'Aggregations' tab. The pipeline consists of one stage: `$match`. The criteria for the `$match` stage are:


```

    {
      email: {
        $regex: "@gameofthron.es",
        $options: "i"
      }
    }
  
```

The output of the aggregation is shown on the right. It displays a sample of 10 documents. One document is visible:


```

    {
      _id: ObjectId('5a9427648b0beebe69579d0'),
      name: 'Talisa Maegyr',
      email: 'oona_chaplin@gameofthron.es',
      movie_id: ObjectId('573a1390f29313caabcd41b...'),
      text: 'Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure _',
      date: 1998-08-22T11:45:03.000+00:00
    }
  
```

MongoDB Compass - localhost:27017/MoviesDB1.comments

Connections Edit View Collection Help

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

- localhost:27017
 - MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
 - MoviesDB1
 - comments
 - movies
 - sessions
 - theaters
 - user
 - admin
 - config
 - local

localhost:27017 > MoviesDB1 > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

\$match \$count Explain Export Run Options

Untitled - modified + CREATE NEW

PREVIEW

Stage 2 \$count

```
1
2 "Email game of thorns"
```

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

Email game of thorns : 22841

MongoDB Compass - localhost:27017/MoviesDB1.comments

Connections Edit View Collection Help

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

- localhost:27017
 - MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
 - MoviesDB1
 - comments
 - movies
 - sessions
 - theaters
 - user
 - admin
 - config
 - local

localhost:27017 > MoviesDB1 > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

\$match \$count Edit Explain Export Run Options

ALL RESULTS

Showing 1 - 1 count results

Email game of thorns : 22841

Exercise 2

How many movie theaters are there in each zip code located within the state of Washington DC (DC)?

```
[
  {
    $match: {
      "location.address.state": "DC"
    }
  },
  {
    $group: {
      _id: "$location.address.zipcode"
    }
  },
  {
    $count: "Total"
  }
]
```

Compass

localhost:27017 > MoviesDB1 > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

\$match

Untitled - modified + CREATE NEW </> PREVIEW {} </> *

Stage 1 \$match

```
1
2 {
3   "location.address.state" : "DC"
4 }
```

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

```
_id: ObjectId('59a47286cfa9a3a73e51e785')
theaterId : 1892
location : Object
```

Compass

localhost:27017 > MoviesDB1 > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

\$match \$group

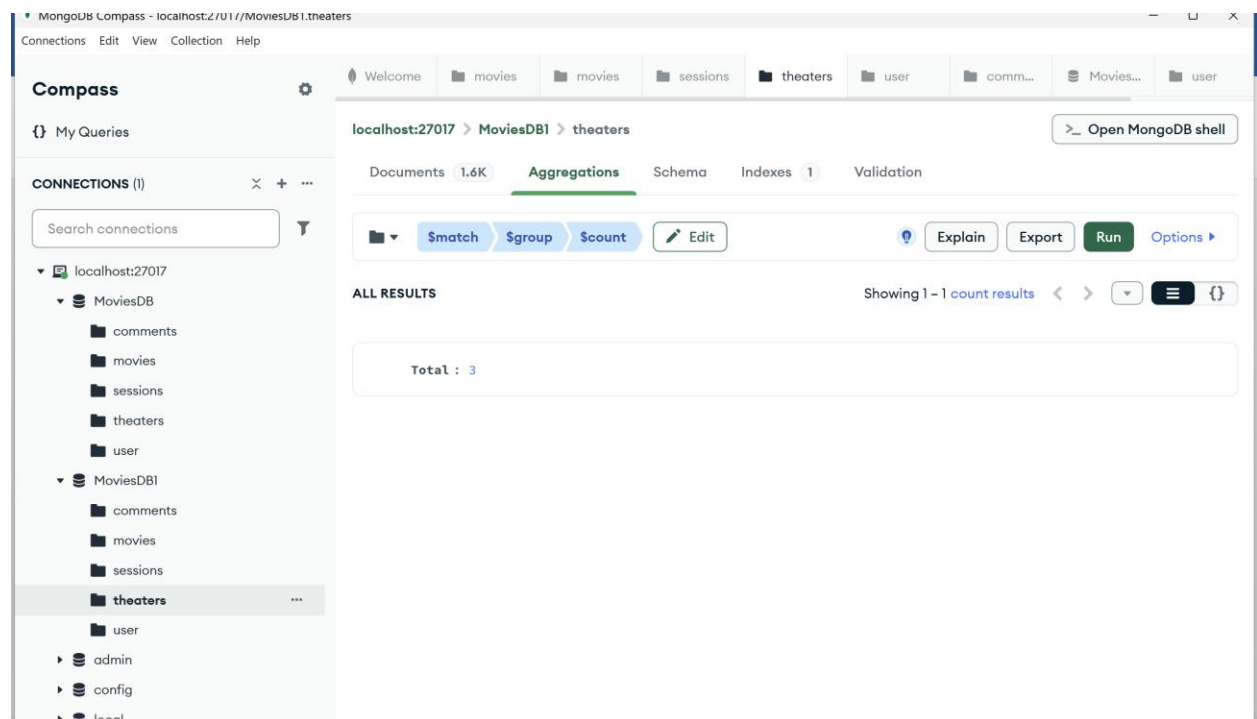
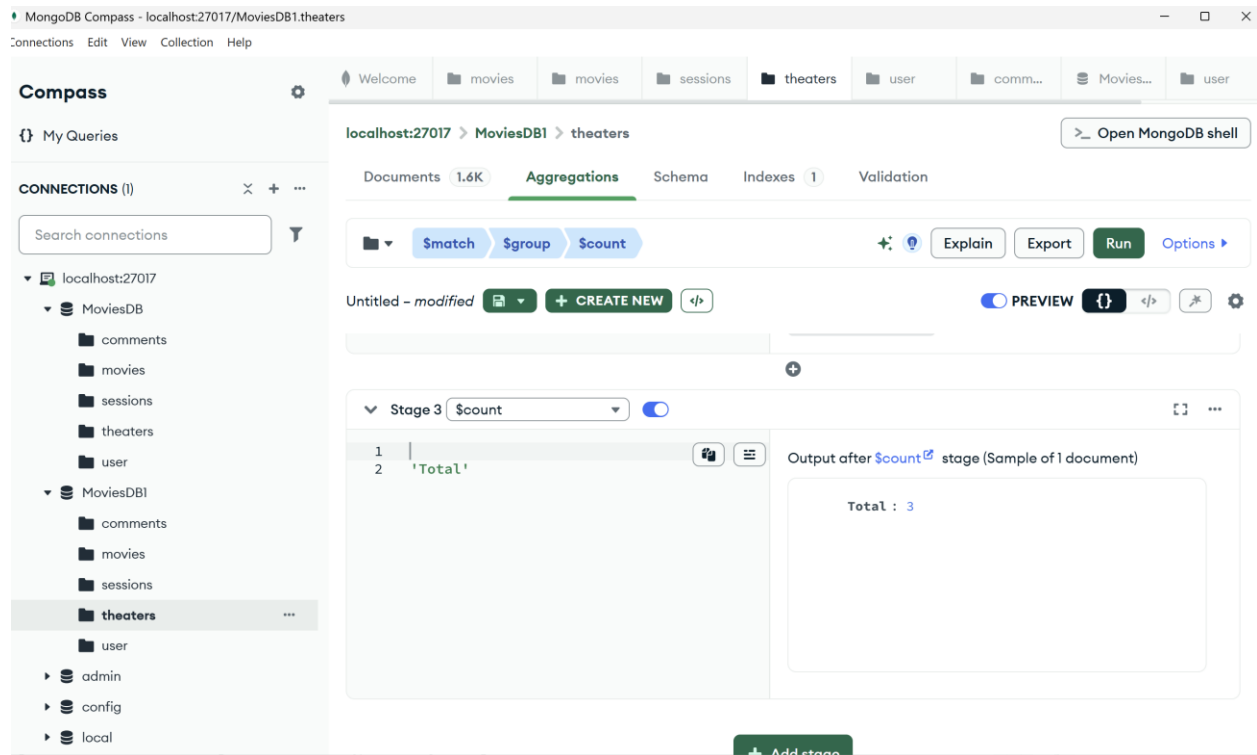
Untitled - modified + CREATE NEW </> PREVIEW {} </> *

Stage 2 \$group

```
1
2 {
3   _id: "$location.address.zipcode"
4 }
5
```

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

```
_id: "20016"
```



Level 3

Exercise 1

Find all films directed by John Landis with an IMDb (Internet Movie Database) rating between 7.5 and 8.

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

MongoDB Compass - localhost:27017/MoviesDB1.movies

Connections Edit View Collection Help

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

- localhost:27017
 - MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
 - MoviesDB1
 - comments
 - movies**
 - sessions
 - theaters
 - user
 - admin
 - config

localhost:27017 > MoviesDB1 > movies

Documents 23.5K Aggregations Schema Indexes 1 Validation

\$match Explain Export Run Options

Untitled - modified + CREATE NEW

PREVIEW

Stage 1 \$match

```
1 |
2 {
3   directors: "John Landis",
4   "imdb.rating": {
5     $gte: 7.5,
6     $lte: 8 }
7 }
```

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": {
    "year": 1978,
    "plot": "At a 1962 college, Dean Vernon Wormer is determined to expel the entire..."
  }
}
```

MongoDB Compass - localhost:27017/MoviesDB1.movies

Connections Edit View Collection Help

Compass

{ } My Queries

CONNECTIONS (1)

Search connections

- localhost:27017
 - MoviesDB
 - comments
 - movies
 - sessions
 - theaters
 - user
 - MoviesDB1
 - comments
 - movies**
 - sessions
 - theaters
 - user
 - admin
 - config
 - local

localhost:27017 > MoviesDB1 > movies

Documents 23.5K Aggregations Schema Indexes 1 Validation

\$match 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": {
    "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": {
    "poster": "https://m.media-amazon.com/images/M/MV5BM2M2ZDA4MTYtOGRjMi00OTg5LWI1ZT..."
  },
  "num_mflix_comments": 1,
  "released": "1978-07-28T00:00:00.000+00:00",
  "awards": {
    "countries": Array (1),
    "cast": Array (4),
    "directors": Array (1)
  }
}
```


Exercise 2

Shows the location of all the theaters in the database on a map.

