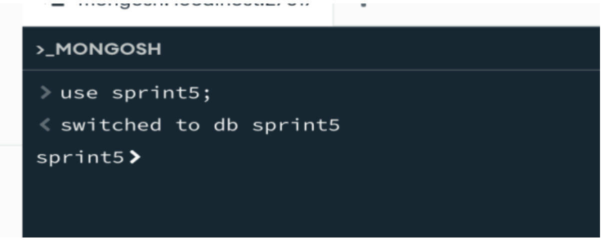
Name: Andre Mazurkiewitz

SPRINT 5

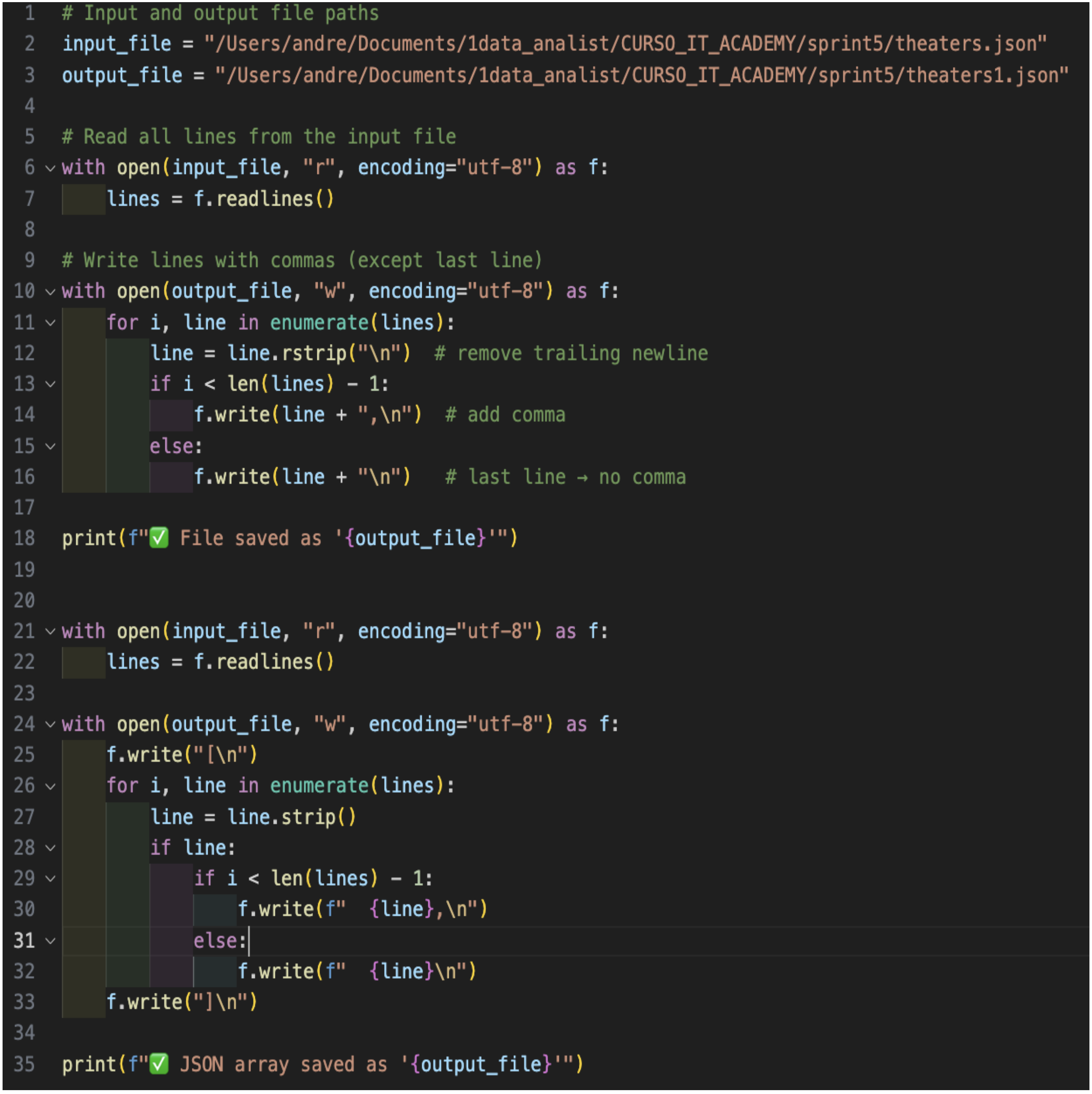
- Level 1

  Database creation;



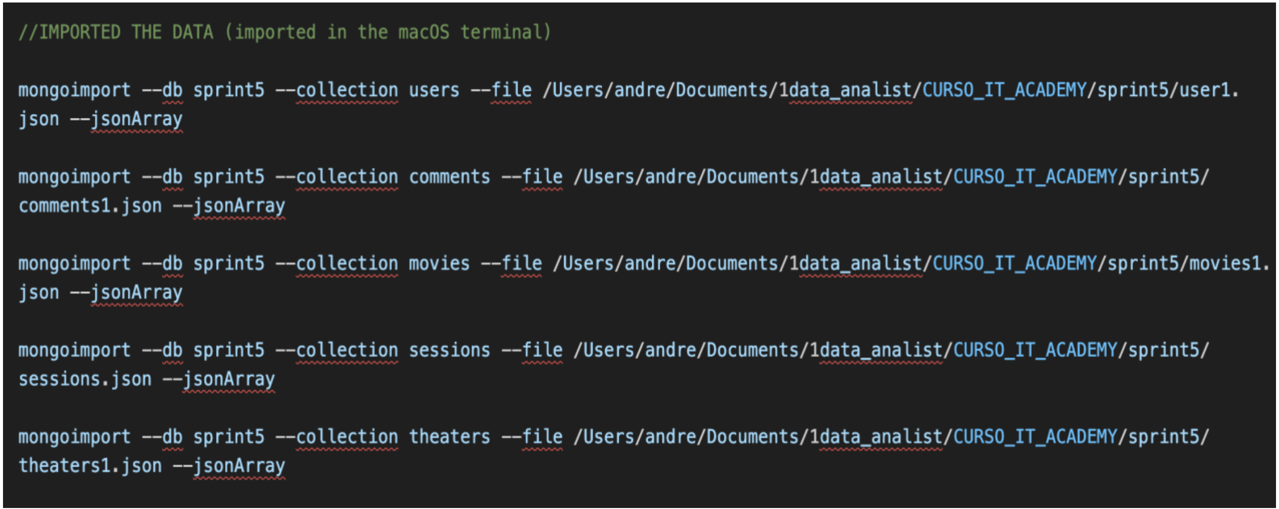
```
>_MONGOSH
> use sprint5;
< switched to db sprint5
sprint5>
```

Then I used the following Python code to insert commas between each line in the given JSON files. chatGTP generated the code.

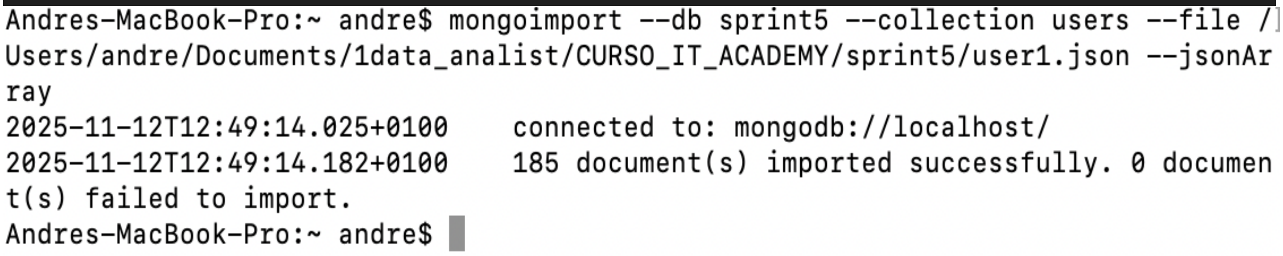


```python
# Input and output file paths
input_file = "/Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/theaters.json"
output_file = "/Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/theaters1.json"

# Read all lines from the input file
with open(input_file, "r", encoding="utf-8") as f:
    lines = f.readlines()

# Write lines with commas (except last line)
with open(output_file, "w", encoding="utf-8") as f:
    for i, line in enumerate(lines):
        line = line.rstrip("\n")  # remove trailing newline
        if i < len(lines) - 1:
            f.write(line + ",\n")  # add comma
        else:
            f.write(line + "\n")   # last line → no comma

print(f"✅ File saved as '{output_file}'")


with open(input_file, "r", encoding="utf-8") as f:
    lines = f.readlines()

with open(output_file, "w", encoding="utf-8") as f:
    f.write("[\n")
    for i, line in enumerate(lines):
        line = line.strip()
        if line:
            if i < len(lines) - 1:
                f.write(f"  {line},\n")
            else:
                f.write(f"  {line}\n")
    f.write("]\n")

print(f"✅ JSON array saved as '{output_file}'")
```

And then imported it to MongoDB using the following command lines.



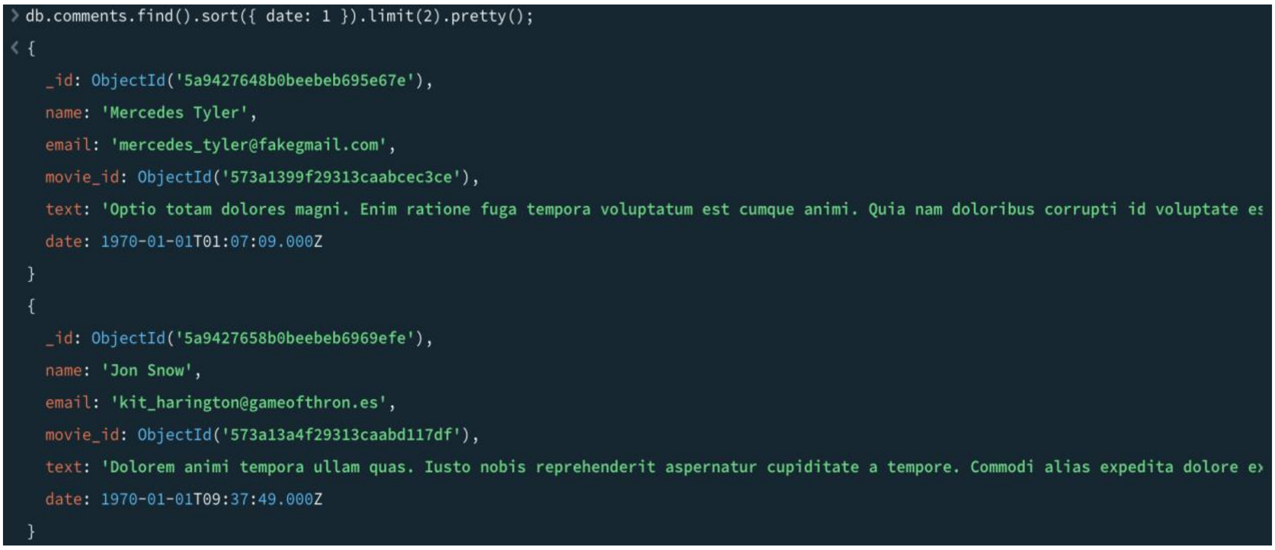
```
//IMPORTED THE DATA (imported in the macOS terminal)

mongoimport --db sprint5 --collection users --file /Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/user1.
json --jsonArray

mongoimport --db sprint5 --collection comments --file /Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/
comments1.json --jsonArray

mongoimport --db sprint5 --collection movies --file /Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/movies1.
json --jsonArray

mongoimport --db sprint5 --collection sessions --file /Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/
sessions.json --jsonArray

mongoimport --db sprint5 --collection theaters --file /Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/
theaters1.json --jsonArray
```



```
Andres-MacBook-Pro:~ andre$ mongoimport --db sprint5 --collection users --file /
Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint5/user1.json --jsonAr
ray
2025-11-12T12:49:14.025+0100    connected to: mongodb://localhost/
2025-11-12T12:49:14.182+0100    185 document(s) imported successfully. 0 documen
t(s) failed to import.
Andres-MacBook-Pro:~ andre$ 
```

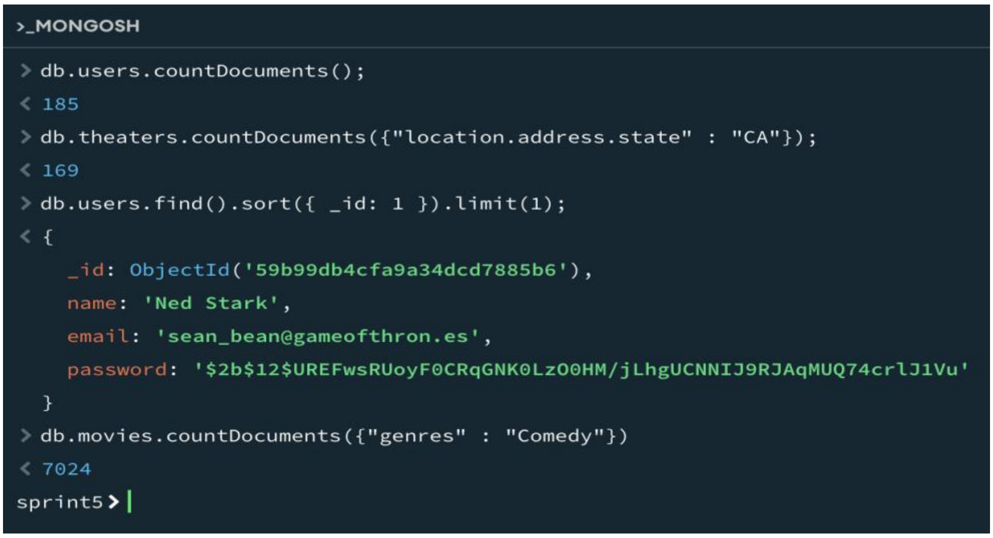
And the same was done for the other 4 files.

- Level 1 Exercise 1
  Shows the first 2 comments in the database



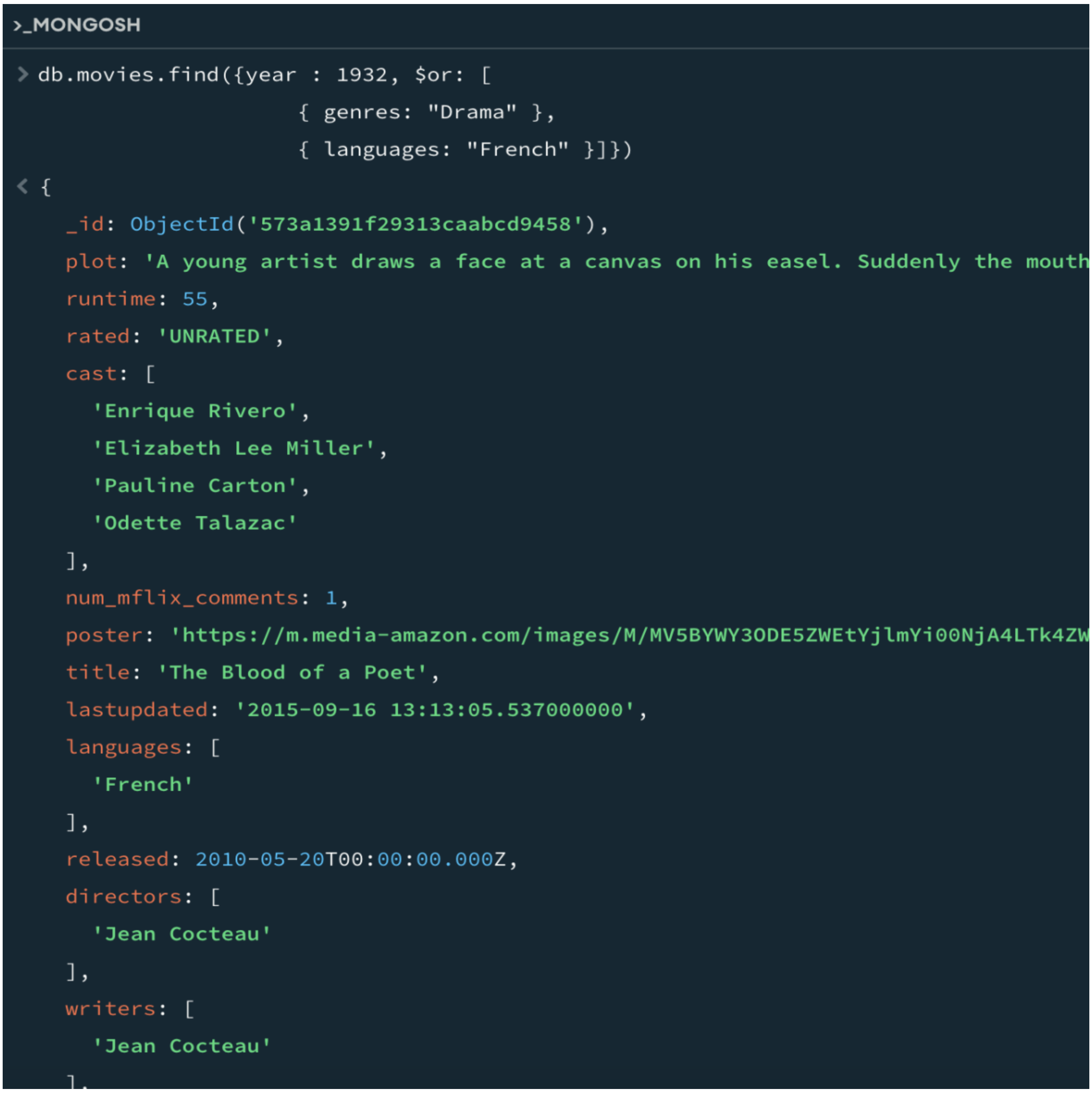
```
> db.comments.find().sort({ date: 1 }).limit(2).pretty();
< {
    _id: ObjectId('5a9427648b0beebeb695e67e'),
    name: 'Mercedes Tyler',
    email: 'mercedes_tyler@fakegmail.com',
    movie_id: ObjectId('573a1399f29313caabcec3ce'),
    text: 'Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cumque animi. Quia nam doloribus corrupti id voluptate es
    date: 1970-01-01T01:07:09.000Z
  }
  {
    _id: ObjectId('5a9427658b0beebeb6969efe'),
    name: 'Jon Snow',
    email: 'kit_harington@gameofthron.es',
    movie_id: ObjectId('573a13a4f29313caabd117df'),
    text: 'Dolorem animi tempora ullam quas. Iusto nobis reprehenderit aspernatur cupiditate a tempore. Commodi alias expedita dolore ex
    date: 1970-01-01T09:37:49.000Z
  }
```

How many users do we have registered? 185
How many movie theaters are in the state of California? 169
Who was the first user to register? Ned Stark
How many comedy films are in our database? 7024



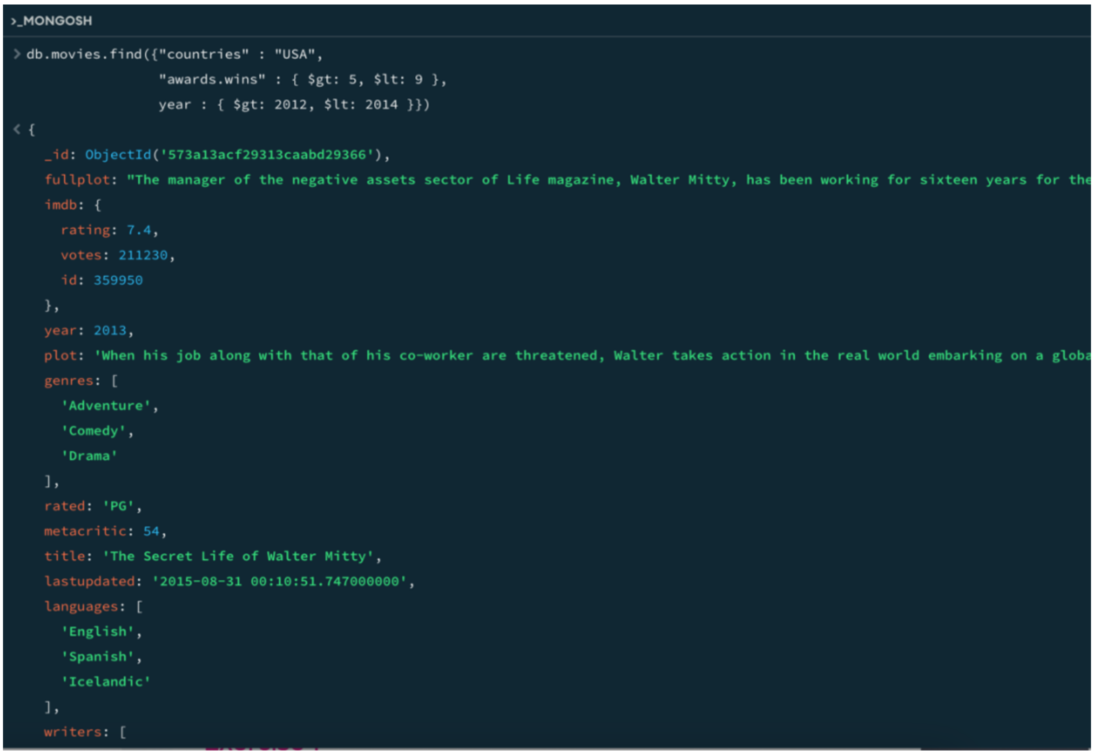
```
> db.users.countDocuments();
< 185
> db.theaters.countDocuments({"location.address.state" : "CA"});
< 169
> db.users.find().sort({ _id: 1 }).limit(1);
< {
    _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
    name: 'Ned Stark',
    email: 'sean_bean@gameofthron.es',
    password: '$2b$12$UREFwsRUoyF0CRqGNK0LzO0HM/jLhgUCNNIJ9RJAqMUQ74crlJ1Vu'
  }
> db.movies.countDocuments({"genres" : "Comedy"})
< 7024
sprint5>
```

- Level 1 Exercise 2



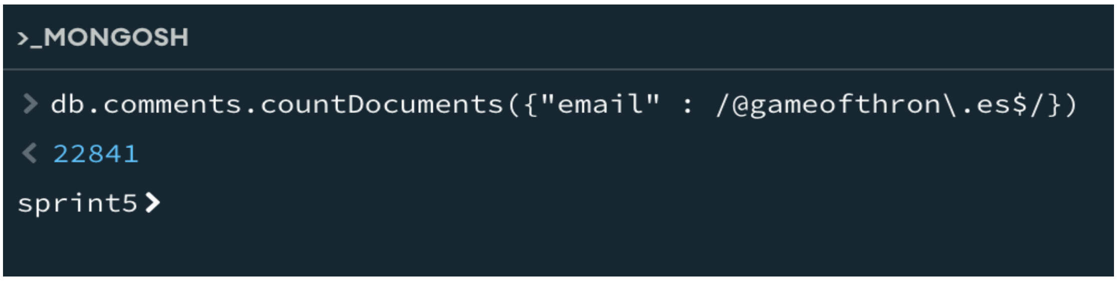
```
>_MONGOSH
> db.movies.find({year : 1932, $or: [
                      { genres: "Drama" },
                      { languages: "French" }]})
< {
    _id: ObjectId('573a1391f29313caabcd9458'),
    plot: 'A young artist draws a face at a canvas on his easel. Suddenly the mouth
    runtime: 55,
    rated: 'UNRATED',
    cast: [
      'Enrique Rivero',
      'Elizabeth Lee Miller',
      'Pauline Carton',
      'Odette Talazac'
    ],
    num_mflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4ZW
    title: 'The Blood of a Poet',
    lastupdated: '2015-09-16 13:13:05.537000000',
    languages: [
      'French'
    ],
    released: 2010-05-20T00:00:00.000Z,
    directors: [
      'Jean Cocteau'
    ],
    writers: [
      'Jean Cocteau'
    ],
```

- Level 1 Exercise 3



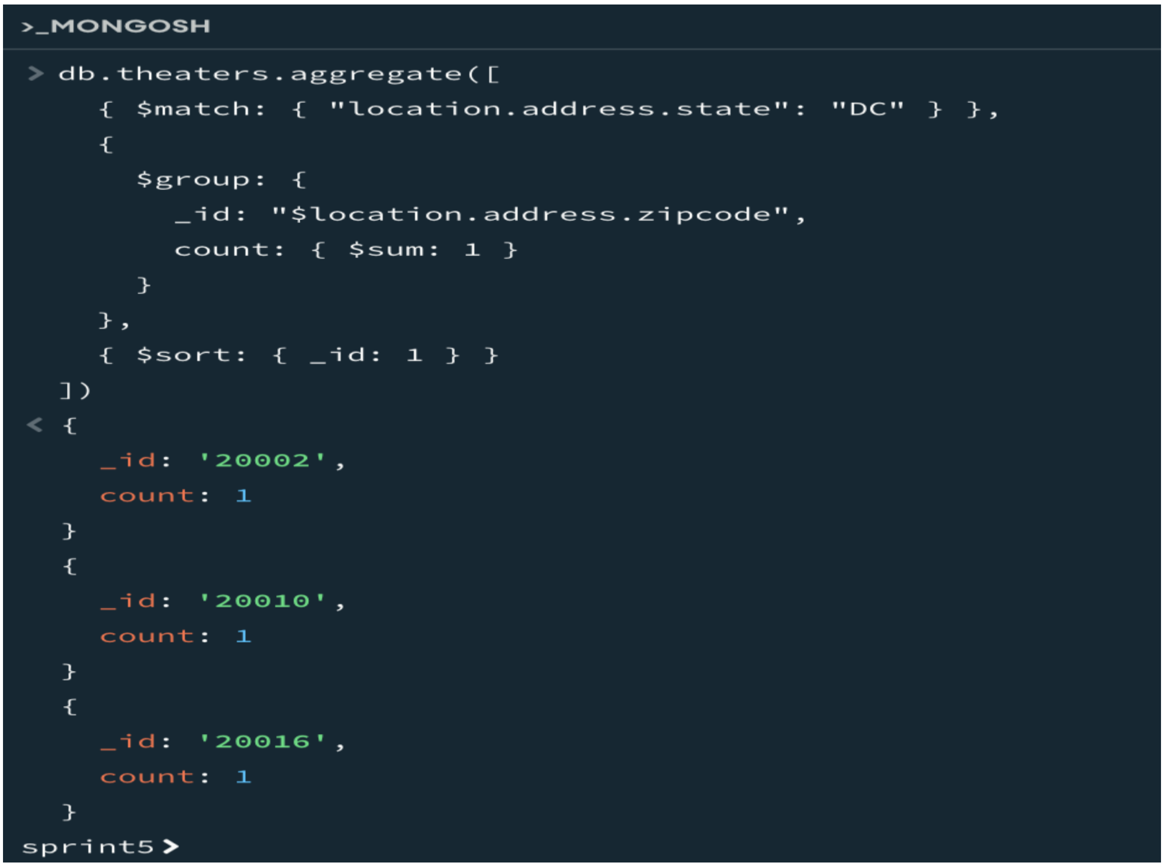
```
>_MONGOSH
> db.movies.find({"countries" : "USA",
                  "awards.wins" : { $gt: 5, $lt: 9 },
                  year : { $gt: 2012, $lt: 2014 }})
< {
    _id: ObjectId('573a13acf29313caabd29366'),
    fullplot: "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen years for the
    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 globa
    genres: [
      'Adventure',
      'Comedy',
      'Drama'
    ],
    rated: 'PG',
    metacritic: 54,
    title: 'The Secret Life of Walter Mitty',
    lastupdated: '2015-08-31 00:10:51.747000000',
    languages: [
      'English',
      'Spanish',
      'Icelandic'
    ],
    writers: [
```

- Level 2 Exercise 1



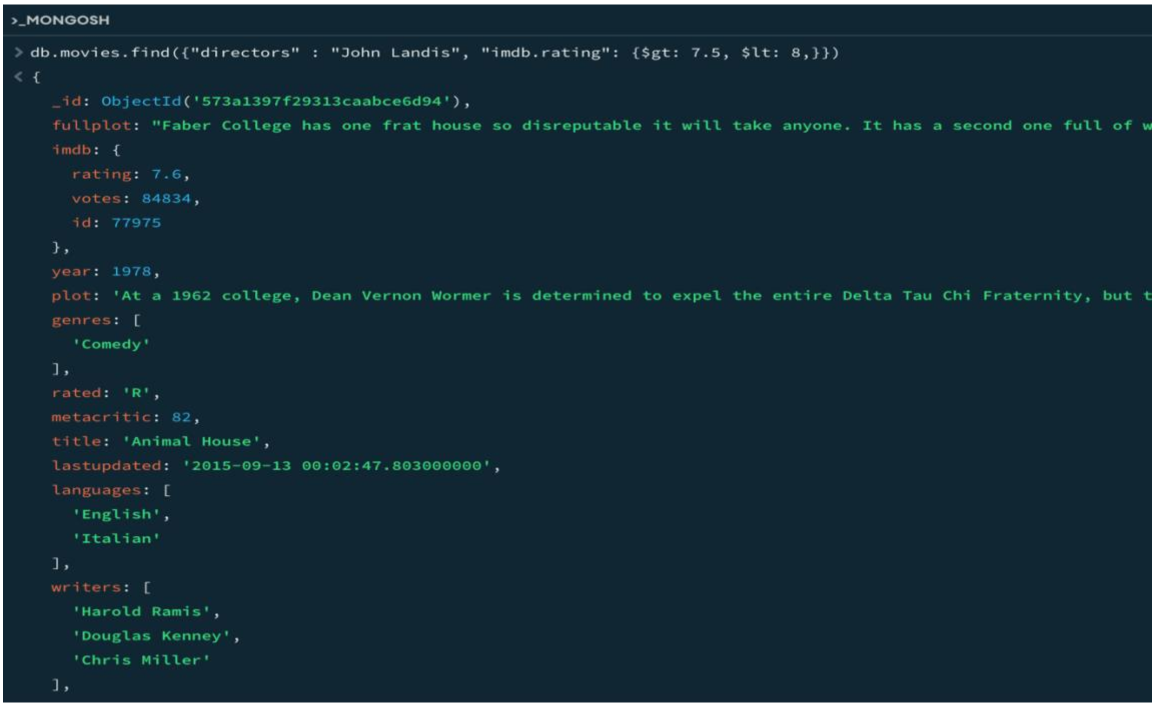
```
>_MONGOSH
> db.comments.countDocuments({"email" : /@gameofthron\.es$/})
< 22841
sprint5>
```

- Level 2 Exercise 2



```
>_MONGOSH
> db.theaters.aggregate([
    { $match: { "location.address.state": "DC" } },
    {
      $group: {
        _id: "$location.address.zipcode",
        count: { $sum: 1 }
      }
    },
    { $sort: { _id: 1 } }
  ])
< {
    _id: '20002',
    count: 1
  }
  {
    _id: '20010',
    count: 1
  }
  {
    _id: '20016',
    count: 1
  }
sprint5>
```

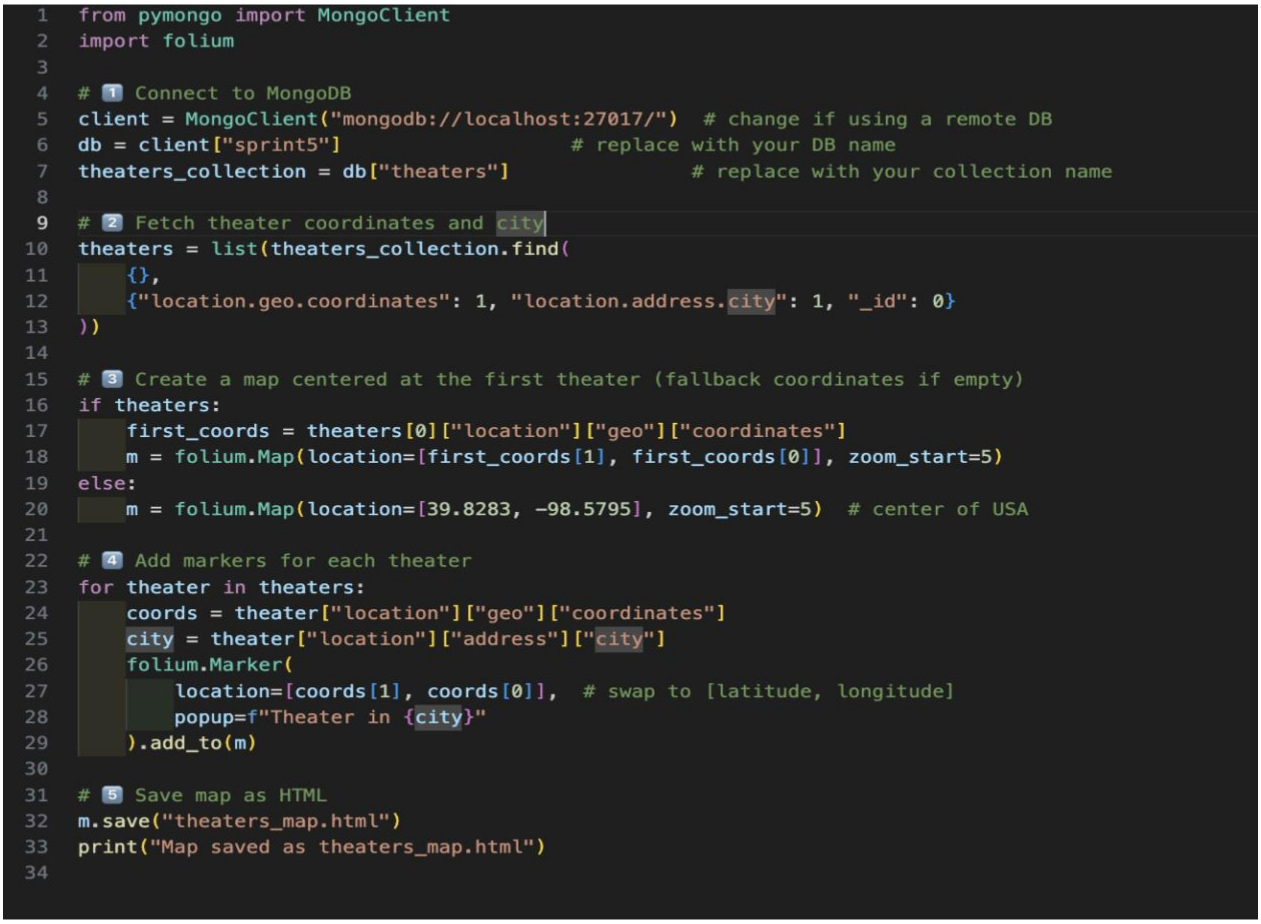
- Level 3 Exercise 1



```
>_MONGOSH
> db.movies.find({"directors" : "John Landis", "imdb.rating": {$gt: 7.5, $lt: 8,}})
< {
    _id: ObjectId('573a1397f29313caabce6d94'),
    fullplot: "Faber College has one frat house so disreputable it will take anyone. It has a second one full of w
    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 t
    genres: [
      'Comedy'
    ],
    rated: 'R',
    metacritic: 82,
    title: 'Animal House',
    lastupdated: '2015-09-13 00:02:47.803000000',
    languages: [
      'English',
      'Italian'
    ],
    writers: [
      'Harold Ramis',
      'Douglas Kenney',
      'Chris Miller'
    ],
```

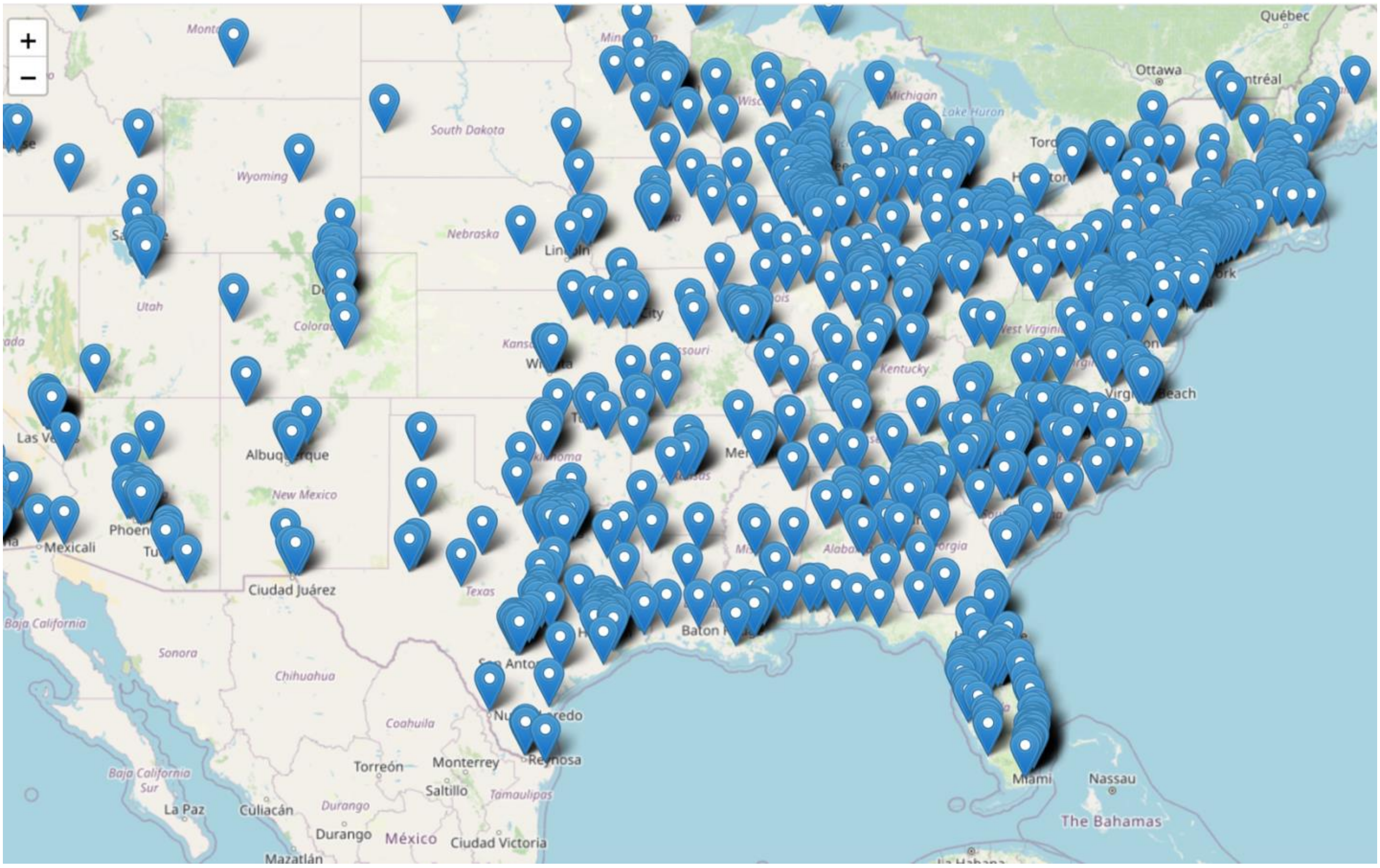
- Level 3 Exercise 2

For this exercise I used the following python code to generate the map. chatGTP generated the code.



```python
from pymongo import MongoClient
import folium

# 1️⃣ Connect to MongoDB
client = MongoClient("mongodb://localhost:27017/")  # change if using a remote DB
db = client["sprint5"]                    # replace with your DB name
theaters_collection = db["theaters"]               # replace with your collection name

# 2️⃣ Fetch theater coordinates and city
theaters = list(theaters_collection.find(
    {},
    {"location.geo.coordinates": 1, "location.address.city": 1, "_id": 0}
))

# 3️⃣ Create a map centered at the first theater (fallback coordinates if empty)
if theaters:
    first_coords = theaters[0]["location"]["geo"]["coordinates"]
    m = folium.Map(location=[first_coords[1], first_coords[0]], zoom_start=5)
else:
    m = folium.Map(location=[39.8283, -98.5795], zoom_start=5)  # center of USA

# 4️⃣ Add markers for each theater
for theater in theaters:
    coords = theater["location"]["geo"]["coordinates"]
    city = theater["location"]["address"]["city"]
    folium.Marker(
        location=[coords[1], coords[0]],  # swap to [latitude, longitude]
        popup=f"Theater in {city}"
    ).add_to(m)

# 5️⃣ Save map as HTML
m.save("theaters_map.html")
print("Map saved as theaters_map.html")
```

And then a .html was created with the map.


+
−
Québec
Ottawa
ntréal
Toro
Lake Huron
Michigan
South Dakota
Wyoming
Nebraska
Lincoln
Utah
Colora
Kansa
Wichita
Kansas City
ssouri
Kentucky
West Virginia
Virginia Beach
York
Las Ve
Albuquerque
New Mexico
Mexicali
Phoen
Tu
Ciudad Juárez
Texas
San Anto
Nue
redo
Reynosa
Baton R
Mem
Alabam
orgia
Miami
Nassau
The Bahamas
La Habana
Baja California
Sonora
Chihuahua
Coahuila
Torreón
Monterrey
Saltillo
Tamaulipas
Baja California Sur
La Paz
Culiacán
Durango
México
Ciudad Victoria
Mazatlán
Monta
Wisc
Min