▾ Assignment03.ipynb Download

Homework Assignment 03

DS 4300 - Spring 2025

- **EC Due Date:** Feb 16, 2025 @ 11:59pm
- **Regular Due Date:** Feb 18, 2025 @ 11:59pm
- Upload to GradeScope (no question/solutions to Match)

Directions:

- Use the mflix sample database to prepare a pymongo query each of the following prompts.
- Be sure to print the results of your query using the `dumps` function.

In [55]:
```
import pymongo
from bson.json_util import dumps
import pprint


uri = "mongodb://ruch:ruch1r%40b%40n123!@localhost:27017/"
client = pymongo.MongoClient(uri)
mflixdb = client.mflix
demodb = client.demodb
```

Question 1:

Give the street, city, and zipcode of all theaters in Massachusetts.

In [57]:
```
ma_theaters =
mflixdb.theaters.find({"location.address.state": "MA"},
                                      {"_id": 0,
"location.address.street1": 1, "location.address.city": 1,
"location.address.zipcode": 1}).limit(5)

print(dumps(ma_theaters, indent=4))
```

```
[
    {
        "location": {
            "address": {
                    "street1": "162 Santilli Hwy",
                    "city": "Everett",
                    "zipcode": "02149"
                }
            }
        },
        {
            "location": {
                "address": {
                    "street1": "14 Allstate Rd",
                    "city": "Dorchester",
                    "zipcode": "02125"
                }
            }
        },
        {
            "location": {
                "address": {
                    "street1": "280 School St",
                    "city": "Mansfield",
                    "zipcode": "02048"
                }
            }
        },
        {
            "location": {
                "address": {
                    "street1": "208 Fortune Blvd",
                    "city": "Milford",
                    "zipcode": "01757"
                }
            }
        },
        {
            "location": {
                "address": {
                    "street1": "33 Orchard Hill Park Dr",
                    "city": "Leominster",
                    "zipcode": "01453"
                }
            }
        }
    ]
```

Question 2:

How many theaters are there in each state? Order the output in alphabetical order by 2-character state code.

In [68]:

```
theater_counts = mflixdb.theaters.aggregate([{"$group":
{"_id": "$location.address.state", "count": {"$sum": 1}}},
    {"$sort": {"_id": 1}},{"$limit": 10}])

print(dumps(theater_counts, indent=4))
```

```
[
    {
        "_id": "AK",
        "count": 4
    },
    {
        "_id": "AL",
        "count": 19
    },
    {
        "_id": "AR",
        "count": 16
    },
    {
        "_id": "AZ",
        "count": 26
    },
    {
        "_id": "CA",
        "count": 169
    },
    {
        "_id": "CO",
        "count": 26
    },
    {
        "_id": "CT",
        "count": 21
    },
    {
        "_id": "DC",
        "count": 3
    },
    {
        "_id": "DE",
        "count": 5
    },
    {
        "_id": "FL",
        "count": 111
    }
]
```

Question 3:

How many movies are in the Comedy genre?

In [9]:
```
comedy_count = mflixdb.movies.count_documents({"genres":
"Comedy"})

print(f"Number of Comedy movies: {comedy_count}")
```

```
Number of Comedy movies: 6532
```

Question 4:

What movie has the longest run time? Give the movie's title and genre(s).

In [18]:
```
longest_movie = mflixdb.movies.find({}, {"_id": 0,
"title": 1, "genres": 1}).sort("runtime", -1).limit(1)

print(dumps(longest_movie, indent=4))
```

```
[
    {
        "genres": [
            "Action",
            "Adventure",
            "Drama"
        ],
        "title": "Centennial"
    }
]
```

Question 5:

Which movies released after 2010 have a Rotten Tomatoes viewer rating of 3 or higher? Give the title of the movies along with their Rotten Tomatoes viewer rating score. The viewer rating score should become a top-level attribute of the returned documents. Return the matching movies in descending order by viewer rating.

In [75]:
```
criteria_top = [
    {"$match": {"year": {"$gt": 2010},
"tomatoes.viewer.rating": {"$gte": 3}}},
    {"$project": {"_id": 0, "title": 1, "viewer_rating":
"$tomatoes.viewer.rating"}},
    {"$sort": {"viewer_rating": -1}},
     {"$limit": 5}
]
criteria_bottom = [
    {"$match": {"year": {"$gt": 2010},
"tomatoes.viewer.rating": {"$gte": 3}}},
    {"$project": {"_id": 0, "title": 1, "viewer_rating":
"$tomatoes.viewer.rating"}},
    {"$sort": {"viewer_rating": 1}},
     {"$limit": 5}
]
matching_movies_top =
mflixdb.movies.aggregate(criteria_top)
matching_movies_bottom =
mflixdb.movies.aggregate(criteria_bottom)

print(dumps(matching_movies_top, indent=4))
print(dumps(matching_movies_bottom, indent=4))
```

```
[
    {
        "title": "Winds",
        "viewer_rating": 5
    },
    {
        "title": "Good Ol' Boy",
        "viewer_rating": 5
    },
    {
        "title": "All Watched Over by Machines of Loving Grace'
        "viewer_rating": 5
    },
    {
        "title": "Scattered Cloud",
        "viewer_rating": 5
    },
    {
        "title": "Beethoven's Christmas Adventure",
        "viewer_rating": 5
    }
]
[
    {
        "title": "From Prada to Nada",
        "viewer_rating": 3
    },
    {
        "title": "Hemingway & Gellhorn",
        "viewer_rating": 3
    },
    {
        "title": "Adult World",
        "viewer_rating": 3
    },
    {
        "title": "Sucker Punch",
        "viewer_rating": 3
    },
    {
        "title": "In Secret",
        "viewer_rating": 3
    }
]
```

Question 6:

How many movies released each year have a plot that contains some type of police activity (i.e., plot contains the word "police")? The returned data should be in ascending order by year.

In [77]:
```
criteria = [{"$match": {"plot": {"$regex": "police",
"$options": "i"}}},
    {"$group": {"_id": "$year", "count": {"$sum": 1}}},
    {"$sort": {"_id": 1}},
    {"$limit": 10}]

police_movies = mflixdb.movies.aggregate(criteria)

print(dumps(police_movies, indent=4))
```

```
[
    {
        "_id": 1913,
        "count": 1
    },
    {
        "_id": 1934,
        "count": 1
    },
    {
        "_id": 1935,
        "count": 1
    },
    {
        "_id": 1944,
        "count": 1
    },
    {
        "_id": 1947,
        "count": 1
    },
    {
        "_id": 1948,
        "count": 2
    },
    {
        "_id": 1949,
        "count": 1
    },
    {
        "_id": 1950,
        "count": 2
    },
    {
        "_id": 1951,
        "count": 2
    },
    {
        "_id": 1957,
        "count": 1
    }
]
```

Question 7:

What is the average number of imdb votes per year for movies released between 1970 and 2000 (inclusive)? Make sure the results are order by year.

In [78]:

```
criteria = [{"$match": {"year": {"$gte": 1970, "$lte": 
2000},"imdb.votes": {"$exists": True}}},
    {"$group": {"_id": "$year", "average_votes": {"$avg": 
"$imdb.votes"}}},
    {"$sort": {"_id": 1}}, {"$limit": 5}
]
```

```
avg_votes = mflixdb.movies.aggregate(criteria)

print(dumps(avg_votes, indent=4))
```

```
[
    {
        "_id": 1970,
        "average_votes": 4786.925
    },
    {
        "_id": 1971,
        "average_votes": 8528.462264150943
    },
    {
        "_id": 1972,
        "average_votes": 13582.685950413223
    },
    {
        "_id": 1973,
        "average_votes": 14478.785714285714
    },
    {
        "_id": 1974,
        "average_votes": 17602.0
    }
]
```

Question 8:

What distinct movie languages are represented in the database? You only need to provide the list of languages.

In [83]:
```
languages = mflixdb.movies.distinct("languages")

print(dumps(languages, indent=4))
```

```
[
    " Ancient (to 1453)",
    " Old",
    "Abkhazian",
    "Aboriginal",
    "Acholi",
    "Afrikaans",
    "Aidoukrou",
    "Albanian",
```