Homework Assignment 03

DS 4300 - Spring 2025

Jake Ashkenase

In [6]:
```python
# Set up your connection to Mongo DB here.
from pymongo import MongoClient
from bson.json_util import dumps

uri = "mongodb://jakeash329:Ashcosh329@localhost:27017/"

client = MongoClient(uri)
mflixdb = client.mflix
```

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.

Question 1:

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

```
In [70]:  query1 = mflixdb.theaters.find( {"location.address.state": "MA"}, {"_id": 0,
                                                                               'location.
                                                                               'location.
                                                                               'location.



          print(dumps(query1, indent=2))
```

```
[
  {
    "location": {
      "address": {
        "street1": "162 Santilli Hwy",
        "city": "Everett",
        "zipcode": "02149"
      }
    }
  },
  {
    "location": {
      "address": {
        "street1": "14 Allstate Rd",
        "city": "Dorchester",
        "zipcode": "02125"
      }
    }
  },
```

Question 2:

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

In [39]:

```
query2 = mflixdb.theaters.aggregate([
    {"$group": {"_id": "$location.address.state", "count": {"$sum": 1}}},

    {"$sort": {"_id": 1}}

])


print(dumps(query2, indent=2))
```

```
[
  {
    "_id": "AK",
    "count": 4
  },
  {
    "_id": "AL",
    "count": 19
  },
  {
    "_id": "AR",
    "count": 16
  },
  {
    "_id": "AZ",
    "count": 26
  },
  {
    "_id": "CA",
```

Question 3:

How many movies are in the Comedy genre?

In [41]:

```
query3 = mflixdb.movies.count_documents({"genres": "Comedy"})

print(dumps(query3, indent=2))
```

```
6532
```

Question 4:

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

In [69]:
```
query4 = mflixdb.movies.find( { }, {"_id": 0, "title": 1, "genres": 1} ).sort

print(dumps(query4, indent=2))
```

```
[
  {
    "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 [66]:  query5 = mflixdb.movies.aggregate(
              [
              {"$match": {"year": {"$gt": 2010},  "tomatoes.viewer.rating": {"$gte": 3}

              {"$project": {"_id": 0, "title": 1, "viewer_rating": "$tomatoes.viewer.ra

              {"$sort": {"viewer_rating": -1}}

              ]
          )

          print(dumps(query5, indent=2))
```

```
[
  {
    "title": "Good Ol' Boy",
    "viewer_rating": 5
  },
  {
    "title": "Winds",
    "viewer_rating": 5
  },
  {
    "title": "Beethoven's Christmas Adventure",
    "viewer_rating": 5
  },
  {
    "title": "Scattered Cloud",
    "viewer_rating": 5
  },
  {
    "title": "All Watched Over by Machines of Loving Grace",
```

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 [76]:
```
query6 = mflixdb.movies.aggregate(
    [
    {"$match": {"fullplot": {"$regex": "police", "$options": "i"}}},

    {"$group": {"_id": "$year", "count": {"$sum": 1}}},

    {"$sort": {"_id": 1}}

    ]
)

print(dumps(query6, indent=2))
```

```
[
  {
    "_id": 1931,
    "count": 2
  },
  {
    "_id": 1932,
    "count": 1
  },
  {
    "_id": 1933,
    "count": 1
  },
  {
    "_id": 1934,
    "count": 1
  },
  {
    "_id": 1941,
```

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 [82]:  query7 = mflixdb.movies.aggregate(
              [
              {"$match": {"year": {"$gte": 1970, "$lte": 2000}}},

              {"$group": {"_id": "$year", "average_votes": {"$avg": "$imdb.votes"}}},

              {"$sort": {"_id": 1}}

              ]
          )

          print(dumps(query7, indent=2))
```

```
[
  {
    "_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,
```

Question 8:

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

In [86]:
```python
query8 = mflixdb.movies.distinct("languages")

print(dumps(query8, indent=2))
```

```
[
  " Ancient (to 1453)",
  " Old",
  "Abkhazian",
  "Aboriginal",
  "Acholi",
  "Afrikaans",
  "Aidoukrou",
  "Albanian",
  "Algonquin",
  "American Sign Language",
  "Amharic",
  "Apache languages",
  "Arabic",
  "Aramaic",
  "Arapaho",
  "Armenian",
  "Assamese",
  "Assyrian Neo-Aramaic",
```