Aditya Gupta
TAS050
MongoDB Assignment

Doubts- unable to run query -votes>1000

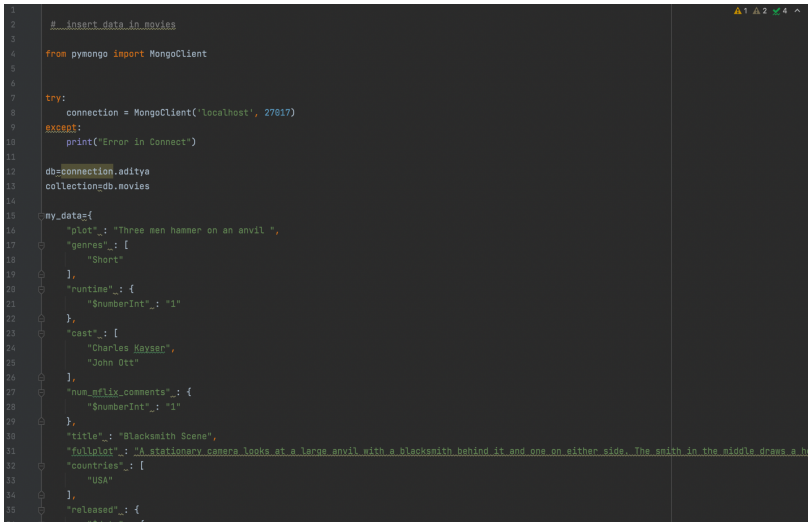
Q1) *Bulk load the JSON files in the individual MongoDB collections using Python. MongoDB collections -*

a. *comments*
b. *movies*
c. *theaters*
d. *Users*

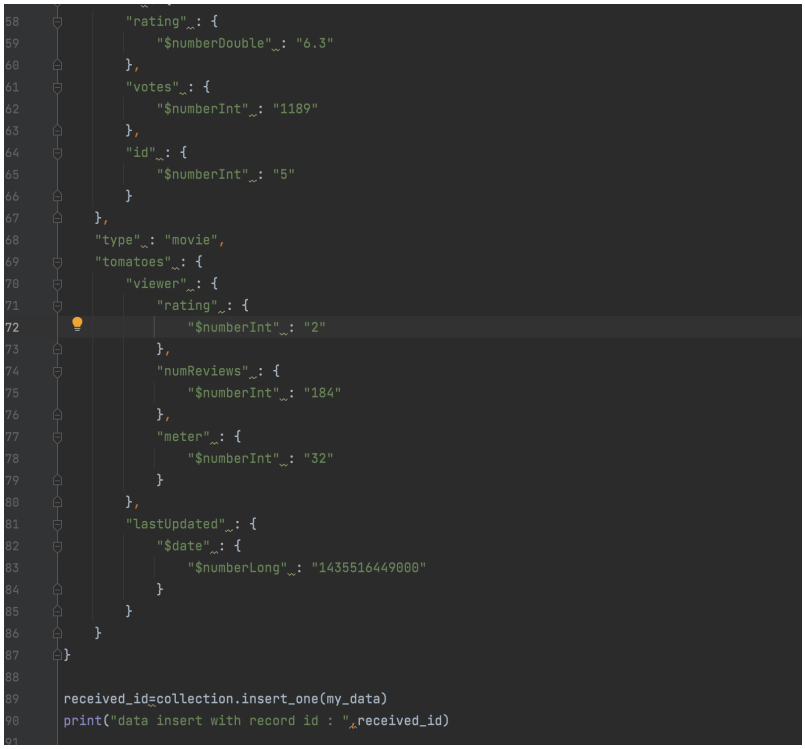
```python
from pymongo import MongoClient
import json
from bson import ObjectId

try:
    connection = MongoClient('localhost', 27017)
except:
    print("Error in Connect")

db = connection['aditya']

collection = db['movies']

item_list = []

with open('/Users/adityagupta/Downloads/mongo database/aditya/movies.json') as f:
    for json_obj in f:
        if json_obj:
            my_dict = json.loads(json_obj)
            my_dict["_id"] = ObjectId(my_dict["_id"]["$oid"])
            item_list.append(my_dict)

collection.insert_many(item_list)
```

In this way i upload all the collections in mongodb through python
The name of the database is **aditya** under which i have collections like movies , theaters, users,and comments.
In try and except part i tried to make connection with the mongodb, if the connection is successful then connecting to the database (aditya) after that connecting to the required collection , changing the id format to normal format and then inserting the values .

Q2) *Create Python methods and MongoDB queries to insert new comments, movies, theatres, and users into respective MongoDB collections.*



```
# insert data in movies

from pymongo import MongoClient


try:
    connection = MongoClient('localhost', 27017)
except:
    print("Error in Connect")

db=connection.aditya
collection=db.movies

my_data={
    "plot": "Three men hammer on an anvil ",
    "genres": [
        "Short"
    ],
    "runtime": {
        "$numberInt": "1"
    },
    "cast": [
        "Charles Kayser",
        "John Ott"
    ],
    "num_mflix_comments": {
        "$numberInt": "1"
    },
    "title": "Blacksmith Scene",
    "fullplot": "A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a h
    "countries": [
        "USA"
    ],
    "released": {
```



```
        "rating": {
            "$numberDouble": "6.3"
        },
        "votes": {
            "$numberInt": "1189"
        },
        "id": {
            "$numberInt": "5"
        }
    },
    "type": "movie",
    "tomatoes": {
        "viewer": {
            "rating": {
                "$numberInt": "2"
            },
            "numReviews": {
                "$numberInt": "184"
            },
            "meter": {
                "$numberInt": "32"
            }
        },
        "lastUpdated": {
            "$date": {
                "$numberLong": "1435516449000"
            }
        }
    }
}

received_id=collection.insert_one(my_data)
print("data insert with record id : "+received_id)
```

In the same manner insert a single value in every collection .

Q4) ***comments*** *collection*

i. *Find top 10 users who made the maximum number of comments*

ii. *Find top 10 movies with most comments*

iii. *Given a year find the total number of comments created each month in that year*

1)
2) Finding the total number of users by email as it is always unique and printing the names and count the no of comments by user using sum function and sorting the result by descending order.

```
# a.i) find top 10 users who made the maximum no of comments

top_10_user = collection.aggregate([

  {"$group": {"_id": {"email": "$email"}, "name": {"$first": "$name"}, "total_comment": {"$sum": 1}}},
  {"$sort": {"total_comment": -1}},
  {"$project": {"_id": 0, "name": 1, "total_comment": 1}},
  {"$limit": 10}
  ])
for user in top_10_user:
    print(user)
```

```
queries_4_b_2 × queries_4_a × 4_b_3 × queries_4_b_2 ×
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/adityagupta/PycharmProjects/mongoproject
{'name': 'Mace Tyrell', 'total_comment': 331}
{'name': 'Missandei', 'total_comment': 327}
{'name': 'The High Sparrow', 'total_comment': 315}
{'name': 'Sansa Stark', 'total_comment': 308}
{'name': 'Rodrik Cassel', 'total_comment': 305}
{'name': 'Robert Jordan', 'total_comment': 304}
{'name': 'Thoros of Myr', 'total_comment': 304}
{'name': 'Brienne of Tarth', 'total_comment': 302}
{'name': 'Megan Richards', 'total_comment': 296}
{'name': 'Catherine Romero', 'total_comment': 296}

Process finished with exit code 0
```

3) In this find the no of movies with the max comment and return the movie id by this and search the given ids in the movies collection to get the name of the movies .
Although not satisfied by the approach suggest some other solution

```python
#  a.ii) find top 10 movies with most comments

top_10= collection.aggregate([

    {"$group": {"_id": {"movies_id": "$movie_id"},"total_comment":{"$sum":1}}},
    {"$sort": {"total_comment": -1}},
    {"$project": {"movies_id": 1,"total_comment":1}},
    {"$limit": 10}
])

mov=list(top_10)
mov_id=[]
for i in range(0,len(mov)):
    oid=mov[i]['_id']['movies_id']['$oid']
    obj=ObjectId(oid)
    mov_id.append(obj)

collection1=db.movies

for id1 in mov_id:
    val=collection1.find({"_id":id1},{"_id":0,"title":1})
    for title in val:
        print(title)
```

Output-

```
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/adityagupta/PycharmProjects/mongoproject/d
{'title': 'The Taking of Pelham 1 2 3'}
{'title': 'Terminator Salvation'}
{'title': 'About a Boy'}
{'title': "Ocean's Eleven"}
{'title': '50 First Dates'}
{'title': 'Sherlock Holmes'}
{'title': 'The Mummy'}
{'title': 'Hellboy II: The Golden Army'}
{'title': 'Anchorman: The Legend of Ron Burgundy'}
{'title': 'The Mummy Returns'}
```

Given a year find the total number of comments created each month in that year

In this i searched for the year and count the no of comments created in month in that year and print the result

```python
# a.iii) given a year find the total no of comments created each month in that year

year=input("enter the year : ")

dic= {  "01":0,"02":0,  "03": 0, "04": 0,"05": 0,"06": 0,"07": 0,"08": 0, "09": 0,"10": 0,"11": 0,"12": 0}
for i in collection.find():
    dte = i['date']['$date']['$numberLong']
    datetime_obj = datetime.fromtimestamp(float(dte)/1e3)
    date = datetime_obj.date()
    x = str(date)
    yr = x[0:4]
    mo = x[5:7]
    if(yr==year):
        dic[mo] +=1

for k,v in dic.items():
    print(k,"->",v)
```

```
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/adityagupta/
enter the year : 1991
01 -> 101
02 -> 82
03 -> 82
04 -> 86
05 -> 80
06 -> 80
07 -> 89
08 -> 88
09 -> 73
10 -> 72
11 -> 81
12 -> 99
```

a. ***movies*** *collection*

i. **Find top `N` movies -**

1. **with the highest IMDB rating**
2. **with the highest IMDB rating in a given year**
3. **with highest IMDB rating with number of votes > 1000**
4. **with title matching a given pattern sorted by highest tomatoes ratings**

1) Just sorting the imdb rating in descending order and limit it by value of n .

```
n=int(input("enter the value of n : "))


# b.i.1) find top n movies with highest imdb rating

result=collection.find({"imdb.rating": {"$ne":""} },{"_id": 0, "title": 1, "imdb.rating": 1}).sort("imdb.rating",-1).limit(n)
for movie in result:
    print(movie)
```

output-

```
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/aditya
enter the value of n : 4
{'title': 'The Marathon Family', 'imdb': {'rating': {'$numberInt': '9'}}}
{'title': 'Prerokbe Ognja', 'imdb': {'rating': {'$numberInt': '9'}}}
{'title': 'Heart of a Dog', 'imdb': {'rating': {'$numberInt': '9'}}}
{'title': 'I, Claudius', 'imdb': {'rating': {'$numberInt': '9'}}}
```

2)

In this we are checking the year in the collection that is enter by the user and project the result like title and imdb rating with the ratings in the sorted order(descending) and limit the no of values to n .

```python
# b.i.2) with the highest imdb rating in a year

year=input("enter the year : ")
answer=collection.aggregate([
    {"$match":{"year.$numberInt":year}},
    {"$project":{"_id":0,"movies":"$title","rating":"$imdb.rating"}},
    {"$sort":{"rating":-1}},
    {"$limit":n}
])
for value in answer:
    print(value)
```

Output-

```
enter the value of n : 4
enter the year : 1994
{'movies': 'Once Were Warriors', 'rating': {'$numberInt': '8'}}
{'movies': "Sharpe's Company", 'rating': {'$numberInt': '8'}}
{'movies': "Sharpe's Enemy", 'rating': {'$numberInt': '8'}}
{'movies': 'Crumb', 'rating': {'$numberInt': '8'}}
```

3) ***with highest IMDB rating with number of votes > 1000***

Had a doubt in it as it is not working as expected

What i do here is convert the imdb votes to int as it is not in int and cannot use to compare with the value until then , after that check the condition if votes>1000 then print the movies with the highest imdb rating

```python
# b.i.3) with no of votes>1000


###  not showing the answer dont know why ...

result1=collection.aggregate([
    {"$project":{"_id":0,"vote":{"$convert":{"input":"$imdb.votes","to":"int","onError":0}},"rating":"$imdb.rating","title":"$title"}},
    {"$match":{"vote": {"$gt":1000}}},
    {"$project":{"_id":0,"title":1,"rating":1,"votes":1}},
    {"$sort":{"rating":-1}},
    {"$limit":n}
])
for movie in result1:
    print(movie)
```

4)with title matching a given pattern sorted by highest tomatoes ratings

In this i use regex function where i find the pattern enter by the user in the title section and print the title with the imdb rating

```
# b.i.4) with title matching a given pattern sorted by highest tomatoes ratings

pattern=input("enter the pattern to search ")
result2=db.movies.find({"title":{"$regex":pattern}},{"_id":0,"title":1,"tomatoes.viewer.rating":1}).sort("tomatoes.viewer.rating",-1).limit(n)
for i in result2:
    print(i)
```

Output-

```
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/adityagupta/PycharmProjects/mongoproject
enter the value of n : 4
enter the pattern to search little
{'title': 'Dr. Dolittle: Million Dollar Mutts', 'tomatoes': {'viewer': {'rating': {'$numberDouble': '3.2'}}}}
{'title': 'Doctor Dolittle', 'tomatoes': {'viewer': {'rating': {'$numberDouble': '3.1'}}}}
{'title': 'Dr. Dolittle 2', 'tomatoes': {'viewer': {'rating': {'$numberDouble': '2.8'}}}}
{'title': 'Doctor Dolittle', 'tomatoes': {'viewer': {'rating': {'$numberDouble': '2.8'}}}}
```

2) Find top `N` directors -

- **a) who created the maximum number of movies**
- **b) who created the maximum number of movies in a given year**
- **c) who created the maximum number of movies for a given genre**

- In the 1st query i unwind the directors as directors are given to us in array , after that i group the directors name and count the number of times they occur in different movies.and then print the names of directors who created the maximum no of movies.
- In the 2nd query i unwind the directors and search for the given year and count the no of directors that created a movie in that year and sort the result according to the no of movies and print the name of directors.
- In this first we need to enter the genre for the movie after that look for that genre in the collection and count the same as did previously.

```python
# 4.2.1) find n directors with max movies
result=db.movies.aggregate([
    {"$unwind":"$directors"},
    {"$group":{"_id":{"director":"$directors"},"num_of_movies":{"$sum":1}}},
    {"$sort":{"num_of_movies":-1}},
    {"$limit":n}
])
for i in result:
    print(i)

# 4.2.2) who created the max num of movies in given year

year=input("enter the year: ")
result1=db.movies.aggregate([
    {"$unwind":"$directors"},
    {"$match":{"year.$numberInt":year}},
    {"$group":{"_id":{"director":"$directors"},"num_of_movies":{"$sum":1}}},
    {"$sort":{"num_of_movies":-1}},
    {"$limit":n}
])
for i in result1:
    print(i)

# 4.2.3) for given genre

genre=input("enter the genre: ")
result2=db.movies.aggregate([
    {"$unwind":"$directors"},
    {"$match":{"genres":genre}},
    {"$group":{"_id":{"director":"$directors"},"num_of_movies":{"$sum":1}}},
    {"$sort":{"num_of_movies":-1}},
    {"$limit":n}
])
for i in result2:
    print(i)
```

```
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/adity
enter the value of n: 2
{'_id': {'director': 'Woody Allen'}, 'num_of_movies': 40}
{'_id': {'director': 'John Ford'}, 'num_of_movies': 35}
enter the year: 1994
{'_id': {'director': 'Tom Clegg'}, 'num_of_movies': 3}
{'_id': {'director': 'Corey Yuen'}, 'num_of_movies': 3}
enter the genre: Drama
{'_id': {'director': 'Jean-Luc Godard'}, 'num_of_movies': 27}
{'_id': {'director': 'Michael Winterbottom'}, 'num_of_movies': 26}
```

3) ***Find top `N` actors -***
   1. ***who starred in the maximum number of movies***
   2. ***who starred in the maximum number of movies in a given year***
   3. ***who starred in the maximum number of movies for a given genre***

- For getting top actors first we need to unwind the actors from the cast array , after that group the actors with the no of movies they starred and sort the result accordingly.
- Rest queries are identical to the previous question

```
# 1 actors in max movies
result1=collection.aggregate([
    {"$unwind":"$cast"},
    {"$group":{"_id":{"actor":"$cast"},"no_of_movies":{"$sum":1}}},
    {"$sort":{"no_of_movies":-1}},
    {"$limit":n}
])
for i in result1:
    print(i)

# 2 who starred in max movies in given year

year=input("enter the year : ")
result2=collection.aggregate([
    {"$match":{"year.$numberInt":year}},
    {"$unwind":"$cast"},
    {"$group":{"_id":{"actor":"$cast"},"no_of_movies":{"$sum":1}}},
    {"$sort":{"no_of_movies":-1}},
    {"$limit":n}
])
for i in result2:
    print(i)

# 3 for given genre

genre=input("enter the genre: ")
result3=collection.aggregate([
    {"$unwind":"$cast"},
    {"$match":{"genres":genre}},
    {"$group":{"_id":{"actor":"$cast"},"no_of_movies":{"$sum":1}}},
    {"$sort":{"no_of_movies":-1}},
    {"$limit":n}
])
for i in result3:
    print(i)
```

```
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/adityagupta/Pyc
enter the value of n : 3
{'_id': {'actor': 'Gèrard Depardieu'}, 'no_of_movies': 68}
{'_id': {'actor': 'Robert De Niro'}, 'no_of_movies': 60}
{'_id': {'actor': 'Michael Caine'}, 'no_of_movies': 53}
enter the year : 1994
{'_id': {'actor': 'J.T. Walsh'}, 'no_of_movies': 5}
{'_id': {'actor': 'Gèrard Depardieu'}, 'no_of_movies': 4}
{'_id': {'actor': 'Tommy Lee Jones'}, 'no_of_movies': 4}
enter the genre: Drama
{'_id': {'actor': 'Gèrard Depardieu'}, 'no_of_movies': 49}
{'_id': {'actor': 'Marcello Mastroianni'}, 'no_of_movies': 39}
{'_id': {'actor': 'Robert Duvall'}, 'no_of_movies': 39}
```

4) *Find top `N` movies for each genre with the highest IMDB rating*

For finding each genre first of all we need to unwind the array genre , then i store all the genre in the python set to get rid of all the duplicates,then iterating on the unique genres and search for it in the movies collection and return the movie names with the highest imdb rating upto n elements.

```python
from pymongo import MongoClient

try:
    connection=MongoClient('localhost',27017)
except:
    print("Error in connection")

db=connection.aditya
collection=db.movies

n=int(input("enter the value of n : "))

result=collection.aggregate([
    {"$unwind":"$genres"},
    {"$project":{"_id":0,"genre":"$genres"}}
])

# to get the distinct genre storing it in a set
distinct=set()
for i in result:
    distinct.add(i.get("genre"))


# iterating through the set to get highest rating movies in each genre
for genre in distinct:
    print("The genre of the movie is : ",genre)
    result=collection.find({"genres":genre},{"_id": 0, "title": 1, "imdb.rating": 1}).sort("imdb.rating",-1).limit(n)
    for movie in result:
        print(movie)
```

```
enter the value of n : 1
The genre of the movie is :  News
{'title': 'Burma VJ: Reporter i et lukket land', 'imdb': {'rating': {'$numberInt': '8'}}}

The genre of the movie is :  Biography
{'title': 'Spartacus', 'imdb': {'rating': {'$numberInt': '8'}}}

The genre of the movie is :  Short
{'title': 'Meshes of the Afternoon', 'imdb': {'rating': {'$numberInt': '8'}}}

The genre of the movie is :  Mystery
{'title': 'The Lady Vanishes', 'imdb': {'rating': {'$numberInt': '8'}}}

The genre of the movie is :  Family
{'title': 'Seven Chances', 'imdb': {'rating': {'$numberInt': '8'}}}

The genre of the movie is :  Western
{'title': 'The Searchers', 'imdb': {'rating': {'$numberInt': '8'}}}

The genre of the movie is :  Talk-Show
{'title': 'The Late Shift', 'imdb': {'rating': {'$numberInt': '7'}}}

The genre of the movie is :  Crime
{'imdb': {'rating': {'$numberInt': '9'}}, 'title': 'The Dark Knight'}

The genre of the movie is :  Animation
{'title': 'Death Note', 'imdb': {'rating': {'$numberInt': '9'}}}

The genre of the movie is :  Documentary
{'title': 'Prerokbe Ognja', 'imdb': {'rating': {'$numberInt': '9'}}}

The genre of the movie is :  Fantasy
{'title': 'King Kong', 'imdb': {'rating': {'$numberInt': '8'}}}
```

a. ***Theater collection***

   i. ***Top 10 cities with the maximum number of theatres***

   ii. ***top 10 theatres nearby given coordinates***

- In the first query we need to find the top 10 cities with max theaters
  So group the cities by their name and count the number of times they appear in the theater collection and print the names according to descending order of total theaters.

```python
# 1. Top 10 cities with  maximum number of theatres

res = collection.aggregate([
        {"$group": {"_id": "$location.address.city","total": {"$sum": 1}}},
        {"$sort": {"total": -1}},
        {"$limit": 10}
])
for i in res:
    print(i)
```

```
/Users/adityagupta/PycharmProjects/mongo
{'_id': 'Las Vegas', 'total': 29}
{'_id': 'Houston', 'total': 22}
{'_id': 'San Antonio', 'total': 14}
{'_id': 'Orlando', 'total': 13}
{'_id': 'Los Angeles', 'total': 12}
{'_id': 'Dallas', 'total': 12}
{'_id': 'Atlanta', 'total': 10}
{'_id': 'Jacksonville', 'total': 9}
{'_id': 'San Francisco', 'total': 9}
{'_id': 'Miami', 'total': 8}
```

2) calculating distance from all coordinates and store it in dictionary and then sort according to the distance and in the value side i have the ids of theater , then getting first 10 theater ids who are near the given coordinates.

```python
# 2. theaters near the given coordinates

dic = {}
coord=['-91.22265','45.91266']
for i in collection.find():
    cord_data = i['location']['geo']['coordinates']
    x = float(coord[0]) - float(cord_data[0]['$numberDouble'])
    y = float(coord[1]) - float(cord_data[1]['$numberDouble'])
    x = round(x * x + y * y, 5)
    if dic.get(x):
        dic[x].append(i['theaterId'])
    else:
        dic[x] = []
        dic[x].append(i['theaterId'])
a = dict(sorted(dic.items()))
ans = []

for k, v in a.items():
    if len(ans) + len(v) > 10:
        x = 10 - len(ans)
        ans += v[0:x]
    else:
        ans += v
    if len(ans) >= 10:
        break

for i in ans:
    print(i)
```

```
/Users/adityagupta/PycharmProjects/mongoproject/venv/bin/python /Users/adityagupta/PycharmProjects/mongoproject/database_load/4_c.py
{'$numberInt': '2757'}
{'$numberInt': '40'}
{'$numberInt': '2718'}
{'$numberInt': '43'}
{'$numberInt': '399'}
{'$numberInt': '15'}
{'$numberInt': '2701'}
{'$numberInt': '10'}
{'$numberInt': '18'}
{'$numberInt': '2916'}
```