INFSCI 2725 Assignment 1

Dan Sun(das225@pitt.edu) Yue Su(yus55@pitt.edu)

Abstraction

In this assignment, we take advantage of mongoDB and Java to accomplish the assignment.

Data Pre-Process

Before we create the data models and functions to find out the answers of questions, we decide to pre-process the 3 data files we have.In this step we decide to build a function to convert the .dat format data into .json format data.After that,we are able to type the following commands in terminal in order to import collections into database.Now, we have a database named “test” and we have 3 data collections:”movies”,”ratings”as well as “tags”.

1.mongoimport --db test --collection movies --drop file movies.json
2.mongoimport --db test --collection ratings --drop file ratings.json
3.mongoimport --db test --collection tags --drop file tags.json

Question 1:What genre is the movie CopyCat in?

—-Answer: we can find the movie”Copycat” is in genres:”**Crime”,”Drama”,”Horror”,”Mystery” and ”Thriller”.**

```
INFO: opened connection [connectionId{localValue:2, serverValue:55}] to 127.0.0.1:27017
Document{{_id=56b869138362d7dde2675136, movieID=22, title=Copycat (1995), generes=[Crime, Drama, Horror, Mystery, Thriller]}}
BUILD SUCCESSFUL (total time: 2 seconds)
```

——Queries:

1.db.getCollection("movies").find(new Document("title", "Copycat (1995)”));

Question 2: What genre has the most movies?

—-Answer:We deal with the question in class”Question2.class”,we are able to find out that the genre **“Drama”** owns the most movies and the total amount is **5339**.

```
INFO: Opened connection [connectionId{localValue:4,
{ "_id" : "Drama", "count" : 5339 }
{ "_id" : "Comedy", "count" : 3703 }
{ "_id" : "Thriller", "count" : 1706 }
{ "_id" : "Romance", "count" : 1685 }
{ "_id" : "Action", "count" : 1473 }
{ "_id" : "Crime", "count" : 1118 }
{ "_id" : "Adventure", "count" : 1025 }
{ "_id" : "Horror", "count" : 1013 }
{ "_id" : "Sci-Fi", "count" : 754 }
{ "_id" : "Fantasy", "count" : 543 }
{ "_id" : "Children", "count" : 528 }
{ "_id" : "War", "count" : 511 }
{ "_id" : "Mystery", "count" : 509 }
{ "_id" : "Documentary", "count" : 482 }
{ "_id" : "Musical", "count" : 436 }
{ "_id" : "Animation", "count" : 286 }
{ "_id" : "Western", "count" : 275 }
{ "_id" : "Film-Noir", "count" : 148 }
{ "_id" : "IMAX", "count" : 29 }
{ "_id" : "(no genres listed)", "count" : 1 }
BUILD SUCCESSFUL (total time: 3 seconds)
```

—-Queries:

1.*db.getCollection("movies").aggregate(asList(new Document("$unwind","$generes"),*

new Document("$group", new Document("_id", "$generes").append("count", new Document("$sum", 1))),

new Document("$sort",new Document("count",-1))));

Question 3: What tags did user 146 use to describe the movie "2001: A Space Odyssey"

—-Answer: These tags are :**"Arthur C. Clarke","artificial intelligence","based on a book".**

```
INFO: Opened connection [connectionId{localValue:2, serverValue:82}] to 127.0.0.1:27017
Arthur C. Clarke
artificial intelligence
based on a book
BUILD SUCCESSFUL (total time: 2 seconds)
```

——Queries:

1.db.getCollection("movies").find(eq("title", "2001: A Space Odyssey (1968)"));

2.db.getCollection("tags").find(and(eq("userID", "146"),eq("movieID",movieID)));

Question 4: What are the top 5 movies with the highest avg rating?

—Answer:

```
ovie_id: 51209  Rating: 5.0  Title: Fighting Elegy (Kenka erejii) (1966)
ovie_id: 64275  Rating: 5.0  Title: Blue Light, The (Das Blaue Licht) (1932)
ovie_id: 53355  Rating: 5.0  Title: Sun Alley (Sonnenallee) (1999)
ovie_id: 33264  Rating: 5.0  Title: Satan's Tango (Sátántangó) (1994)
ovie_id: 42783  Rating: 5.0  Title: Shadows of Forgotten Ancestors (1964)
```

—-Queries:

1.*db.getCollection("ratings").aggregate(asList(new Document("$group", new Document("_id", "$movieID").append("AVR_Rating", new Document("$avg", "$rating"))),new Document("$sort",new Document("AVR_Rating",-1))));*

2.db.getCollection("movies").find(eq("movieID",MymovieID));

Question5: What is the highest avg rating possible?

—Answer: we can find the highest avg rating is **:5.0.**

```
ovie_id: 51209  Rating: 5.0  Title: Fighting Elegy (Kenka erejii) (1966)
ovie_id: 64275  Rating: 5.0  Title: Blue Light, The (Das Blaue Licht) (1932)
ovie_id: 53355  Rating: 5.0  Title: Sun Alley (Sonnenallee) (1999)
ovie_id: 33264  Rating: 5.0  Title: Satan's Tango (Sátántangó) (1994)
ovie_id: 42783  Rating: 5.0  Title: Shadows of Forgotten Ancestors (1964)
```

—Queries:

1.*db.getCollection("ratings").aggregate(asList(new Document("$group", new Document("_id", "$movieID").append("AVR_Rating", new Document("$avg", "$rating"))),new Document("$sort",new Document("AVR_Rating",-1))));*

Demostration:

In order to demonstrate our data storage is working, we use 3 different queries to demonstrate.

The first query is to count the total number of movies in this database. I use count function in collection movies.

```
the total movies are: 10681
```

The second query is to select the rating record according to a given user's ID. The result is shown as below:

```
Please enter userID: you can check the rating records
123
Document{{_id=56b95da78362d7dde201e4e0, userID=123, movieID=15, rating=3.0, timestamp=981573054}}
Document{{_id=56b95da78362d7dde201e4e1, userID=123, movieID=34, rating=4.0, timestamp=983457462}}
Document{{_id=56b95da78362d7dde201e4e2, userID=123, movieID=110, rating=5.0, timestamp=985968637}}
Document{{_id=56b95da78362d7dde201e4e3, userID=123, movieID=260, rating=5.0, timestamp=983384806}}
Document{{_id=56b95da78362d7dde201e4e4, userID=123, movieID=356, rating=5.0, timestamp=985885011}}
Document{{_id=56b95da78362d7dde201e4e5, userID=123, movieID=527, rating=5.0, timestamp=985883840}}
Document{{_id=56b95da78362d7dde201e4e6, userID=123, movieID=539, rating=5.0, timestamp=985885011}}
Document{{_id=56b95da78362d7dde201e4e7, userID=123, movieID=653, rating=5.0, timestamp=983384901}}
```

The third query is to check a certain movie's tag. The result is shown as below.

```
Please enter movieID:
134
Document{{_id=56b908018362d7dde268df38, userID=61327, movieID=134, tag=illegal art, timestamp=1184949668}}
```