**Document Structure**

In this assignment, I create three documents which store the data of the three .dat file. The document structure is designed as below:

Document Structure 1 for movies.dat

{

"movieid" : 65130 ,

"title" : "Revolutionary Road (2008)" ,

"gen" : [ "Drama" , "Romance"]

}

Document Structure 2 for tags.dat

{

"movieid" : 6299 ,

"tags" : [

{ "userid" : 146 , "tag" : "animals" , "timestamp" : 1204114254} ,

{ "userid" : 146 , "tag" : "nature" , "timestamp" : 1196066387}

]

}

Document Structure 3 for ratings.dat

{

"movieid" : 1 ,

"userid" : 1 ,

"ratings" : 5.0 ,

"timestamp" : 838985046

}

**Program structure brief**

I program with java using eclipse on win7 64 bit. And there are three projects in the .zip.

Project ‘movies’ solves query 1 and query 6 using data of movies.dat.

Project ‘ratings’ solves query 2 and query3 using data of ratings.dat.

Project ‘tags’ solves query 4 and query5 using data of ratings.dat.

Data storage process is programed in the main class and all the queries in the functions.

* movies.java

**public** **class** movies {

**public** **static** **void** main(String[] args) {

//connect with mongodb

//read from the ‘movies.dat’ file

//get data from file

//transform data into Mongodb document

//insert data into mongodb

*showdata*(movies);

*countmoviebygen*(movies);

}

**public** **static** **void** countmoviebygen(DBCollection movies) // Query 1 function

**public** **static** **void** showdata(DBCollection users) // Query 6 function

get all the data from mongodb

}

* Rating.java

public class rating {

public static void main(String[] args) {

… //connect with mongodb

… //read from the ‘ratings.dat’ file

… //get data from file

… //transform data into Mongodb document

… //insert data into mongodb

showsameuserid(ratings,1); //Query 2 function. The second parameter will be the id of target user.

showsimilaruserid(ratings,1);

getavgratebymovie(ratings,1); // find target movie’s average rating

getallmovieavg(ratings); //Query 3 function.

showdata(ratings);

}

public static void getallmovieavg(DBCollection ratings)

public static void getavgratebymovie(DBCollection ratings,Object object)

public static void showsimilaruserid(DBCollection ratings,int userid)

public static void showsameuserid(DBCollection ratings,int userid)

… // Another supporting functions

}

* Tags.java

**public** **class** tags {

**public** **static** **void** main(String[] args) {

… //connect with mongodb

… //read from the ‘ratings.dat’ file

… //get data from file

… //transform data into Mongodb document

// if movie has already exist

{ …// update }

**//else** { … //insert data into mongodb }

*showdat*a(movies);

*showmosttagmovie*(movies);

*showmoviesize*(movies);

}

**public** **static** **void** showmosttagmovie(DBCollection movies)

**public** **static** **void** showmoviesize(DBCollection movies)

**public** **static** **void** showdata(DBCollection movies)

}

**Result**

1. Write a query that finds average rating of each movie.
2. show specified movie’s average rating by ‘getavgratebymovie(DBCollection,movieid)’

movie No.1 average rating is 3.928768573481039

2. show all the movies’ average rating by ‘getallmovieavg(DBCollection)’

movie No.1average rating is 3.928768573481039

movie No.2average rating is 3.208070146276596

movie No.3average rating is 3.150385109114249

movie No.4average rating is 2.860544217687075

movie No.5average rating is 3.0774351786965664

movie No.6average rating is 3.813011098130841

movie No.7average rating is 3.365017361111111

movie No.8average rating is 3.131256952169077

movie No.9average rating is 2.9968228752978554

movie No.10average rating is 3.4283012176380185

1. Write a query that finds users who are similar to a given user (target user), the id of the target user is an input parameter. Users are similar to the target user if they rate the same movies.

1.show the user’s id who only rated the movies that target user rated.

Using function ‘showsameuserid(DBCollection,userid)’

same user's id set: [1]

2.show the user’s id who rated the movies that target user rated.

Using function ‘showsimilaruserid(DBCollection,userid)’

similar user's id: [1, 556, 719, 1819, 2498, 2787, 4383, 9128, 8249, 8622, 10484, 12030, 12303, 15012, 14463, 15722, 15560, 19635, 20452, 21732, 20597, 25082, 28562, 27764, 28126, 27468, 30592, 30575, 29877, 32584, 31758, 31642, 35052, 36139, 34786, 42791, 47842, 47345, 47213, 48901, 48275, 45997, 45157, 52291, 52405, 51571, 50008, 49682, 56915, 55328, 54644, 54630, 53354, 61172, 59943, 59659, 59269, 58357, 58146, 58087, 57481, 62579, 65567, 67385, 67058]

1. Write a query that finds to number of movies in each genre.

Comedy has 3703

Action has 1473

Adventure has 1025

Children's has 528

Crime has 1118

Documentary has 482

Drama has 5339

Fantasy has 543

Film-Noir has 148

Musical has 436

Romance has 1685

Sci-Fi has 754

Thriller has 1706

War has 511

Western has 275

4)find the movie which was tagged by the most times.

The most tagged movie is No. 296 by 308 times

5)find the number of movies which was tagged as ‘base on the book’

The number of Movies which was tagged as 'based on a book' was 369

6)show all the movie.

{ "\_id" : { "$oid" : "5111721e36a826410558c577"} , "movieid" : 64999 , "title" : "War of the Worlds 2: The Next Wave (2008)" , "gen" : [ "Action"]}

{ "\_id" : { "$oid" : "5111721e36a826410558c578"} , "movieid" : 65001 , "title" : "Constantine's Sword (2007)" , "gen" : [ "Documentary"]}

{ "\_id" : { "$oid" : "5111721e36a826410558c579"} , "movieid" : 65006 , "title" : "Impulse (2008)" , "gen" : [ "Mystery" , "Thriller"]}

{ "\_id" : { "$oid" : "5111721e36a826410558c57a"} , "movieid" : 65011 , "title" : "Zona Zamfirova (2002)" , "gen" : [ "Comedy" , "Drama"]}

{ "\_id" : { "$oid" : "5111721e36a826410558c57b"} , "movieid" : 65025 , "title" : "Double Dynamite (1951)" , "gen" : [ "Comedy" , "Musical"]}

{ "\_id" : { "$oid" : "5111721e36a826410558c57c"} , "movieid" : 65027 , "title" : "Death Kiss, The (1933)" , "gen" : [ "Comedy" , "Mystery"]}

{ "\_id" : { "$oid" : "5111721e36a826410558c57d"} , "movieid" : 65037 , "title" : "Ben X (2007)" , "gen" : [ "Drama"]}