

IMDB Movie Database and Query Generator



Design and develop a movie database and a query generator for IMDB movie data. You are given a .csv file which stores the following information for each movie. There are around 5000 movies listed in the file.

- id
- Color
- movie_title
- genres
- duration
- director_name
- actor_1_name • actor_2_name
- actor_3_name
- plot_keywords
- movie_imdb_link
- language
- country
- content_rating
- title_year
- imdb_score

Functional and Design Requirements Your

program

- creates a movie database by reading the data from .csv file into an array
- allows to add as many fields possible as search index ○ each time a new field index is added to the database, a new red black tree is created by the given field as the key. For example, `db.addFieldIndex("title")` will create a new red black tree by title field. Then, key is the title, and the value is the set of id's of movies having the same title.
- stores red black trees in a hash table
- allows to create a query by combining one or more of the following queries.
 - and
 - or ○ not
 - greater than or equal to ○ less than or equal

to ○ equal to ○ not

equal to

- Executes the query using the indexing trees
- Prints the information of all the movies that are in the result set

A sample test case is provided below. The program prints the movie information for all records with year= 2013 and imdb_scores=6.1.

```
package database;
```

```
public class MoviesDB<T extends Comparable<T>> {
```

```
    private String fileName;
    private Map<String, RedBlackTree<T, HashSet<Integer>>> indexTreeMap
= new HashMap<String, RedBlackTree<T, HashSet<Integer>>>();    private Movie[] db;
    private int n;
```

```
    //load the array with the data given in the csv file    public MoviesDB(String fileName)
throws FileNotFoundException{
```

```
}
```

```
    //create a new red black tree by field
    public void addFieldIndex(String field) {
```

```
}
```

```
    //returns the hash map for index trees (red black trees)    public Map<String,
RedBlackTree<T, HashSet<Integer>>> getIndexTreeMap(){
        return indexTreeMap;
    }
```

```
    //sample text case
```

```
    public static void main(String[] args) throws FileNotFoundException {
        MoviesDB movieDB = new MoviesDB("simple.csv");
        movieDB.addFieldIndex("year");
        movieDB.addFieldIndex("imdb_score");
        Query<Integer> query=new And(new Equal("year",2012),new Equal("imdb_score",6.1));
        result = (HashSet<Integer>) query.execute(movieDB.getIndexTreeMap());
        if(result!=null)
            System.out.println(result);
        Iterator<Integer> iditerator = result.iterator();
        while(iditerator.hasNext()) {
            int id = iditerator.next();
            movieDB.print(id);
        }
    }
}
```

//simple.csv

```
id,color,movie_title,duration,director_name,actor_1_name,actor_2_name,actor_3_name,movie_imdb_link,language,country,content_rating,title_year,imdb_score
1,Color,Avatar,178,James Cameron,CCH Pounder,Joel David Moore,Wes Studi,http://www.imdb.com/title/tt0499549/?ref_=fn_tt_1,English,USA,PG-13,2009,7.9
2,Color,Pirates of the Caribbean: At World's End,169,Gore Verbinski,Johnny Depp,Orlando Bloom,Jack Davenport,http://www.imdb.com/title/tt0449088/?ref_=fn_tt_1,English,USA,PG-13,2007,7.1
3,Color,Spectre,148,Sam Mendes,Christoph Waltz,Bory Kinnear,Stephanie Sieman,http://www.imdb.com/title/tt2379713/?ref_=fn_tt_1,English,UK,PG-13,2012,6.8
4,Color,John Carter,132,Andrew Stanton,Daryl Sabara,Samantha Morton,Polly Walker,http://www.imdb.com/title/tt0401729/?ref_=fn_tt_1,English,USA,PG-13,2012,6.6
5,Color,Spider-Man 3,156,Sam Raimi,J.K. Simmons,James Franco,Kirsten Dunst,http://www.imdb.com/title/tt0413300/?ref_=fn_tt_1,English,USA,PG-13,2012,6.1
6,Color,Tangled,100,Nathan Greno,Brad Garrett,Donna Murphy,M.C. Gainey,http://www.imdb.com/title/tt0398286/?ref_=fn_tt_1,English,USA,PG,2010,7.8
7,Color,Avengers: Age of Ultron,141,Joss Whedon,Chris Hemsworth,Robert Downey Jr.,Scarlett Johansson,http://www.imdb.com/title/tt2395427/?ref_=fn_tt_1,English,USA,PG-13,2015,7.5
8,Color,Harry Potter and the Half-Blood Prince,153,David Yates,Alan Rickman,Daniel Radcliffe,Rupert Grint,http://www.imdb.com/title/tt0417741/?ref_=fn_tt_1,English,UK,PG,2009,7.5
9,Color,Batman v Superman: Dawn of Justice,183,Zack Snyder,Henry Cavill,Lauren Cohan,Alan D. Purwin,http://www.imdb.com/title/tt2975590/?ref_=fn_tt_1,English,USA,PG-13,2016,6.9
10,Color,Superman Returns,169,Bryan Singer,Kevin Spacey,Marlon Brando,Frank Langella,http://www.imdb.com/title/tt0348150/?ref_=fn_tt_1,English,USA,PG-13,2012,6.1
```

Sample Output:

[5, 10]

id:5 color:Color color:Color
title:Spider-Man 3 duration:156
director_name:Sam Raimi
act1:J.K. Simmons act2:James
Franco act3:Kirsten Dunst
movie_imdb_link:http://www.imdb.com/title/tt0413300/?ref_=fn_tt_tt_1 language:English
country:USA content_rating:PG-13 title_year:2012 imdb_score:6.1

----- id:10 color:Color
color:Color title:Superman Returns
duration:169 director_name:Bryan Singer
act1:Kevin Spacey act2:Marlon Brando
act3:Frank Langella
movie_imdb_link:http://www.imdb.com/title/tt0348150/?ref_=fn_tt_tt_1 language:English
country:USA content_rating:PG-13 title_year:2012 imdb_score:6.1

Examples for more queries:

```
Query<Integer> query=new Not(new Equal("color", "Color"));
```

```
Query<Integer> query=new And(new LT("imdb_score", 7.0), new GT("imdb_score", 6.0));
```

```
Query<Integer> query=new And(new Or(new Equal("year", 2013), new GTE("imdb_score", 6.0)), new NotEqual("language", "English"));
```