**1. Project Name:** Movie review

**2. Project Requirements**

- Hadoop components: Java MapReduce

**3. Problem Statement:**

1. Top ten most viewed movies with their movies Name (Ascending or Descending order)

2. Top twenty rated movies (Condition: The movie should be rated/viewed by at least 40 users)

**4. Description:**

The problem mentioned below revolves around movies dataset. The dataset contains 4 files which are follows,

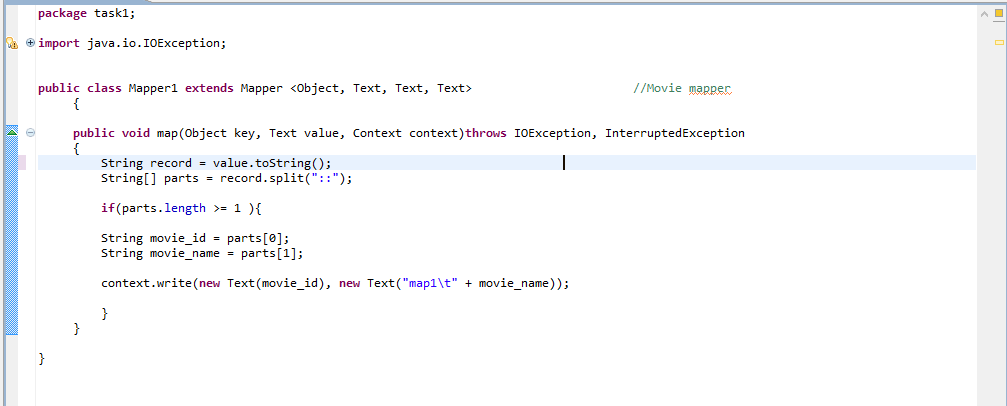
|  |  |
| --- | --- |
| **File Name** | **Description / Schema** |
| movies.dat | MovieID – Title – Genres |
| ratings.dat | UserID – MovieID – Rating – Timestamp |
| users.dat | UserID – Gender – Age – Occupation – ZipCode |

**5. Project Objective and Description of the project in detail points by points**

* ***Top ten most viewed movies with their movies Name (Ascending or Descending order)***
  + 1. The objective is to obtain top ten most viewed movie name in descending order using MapReduce program
    2. Using “ Reduce join “ joined achieved between two data sets.
    3. Using Hashmap achieved descending order and Limiting the top ten viewed movies.
* ***Top twenty rated movies (Condition: The movie should be rated/viewed by at least 40 users)***
  + 1. The objective is to obtain Top twenty rated movieswhich condition.
    2. Using “ Reduce join “ joined achieved between two data sets.
    3. Using Hashmap achieved descending order and Limiting the top 40 rated movie.

**Task1: *Top ten most viewed movies with their movies Name (Ascending or Descending order)***

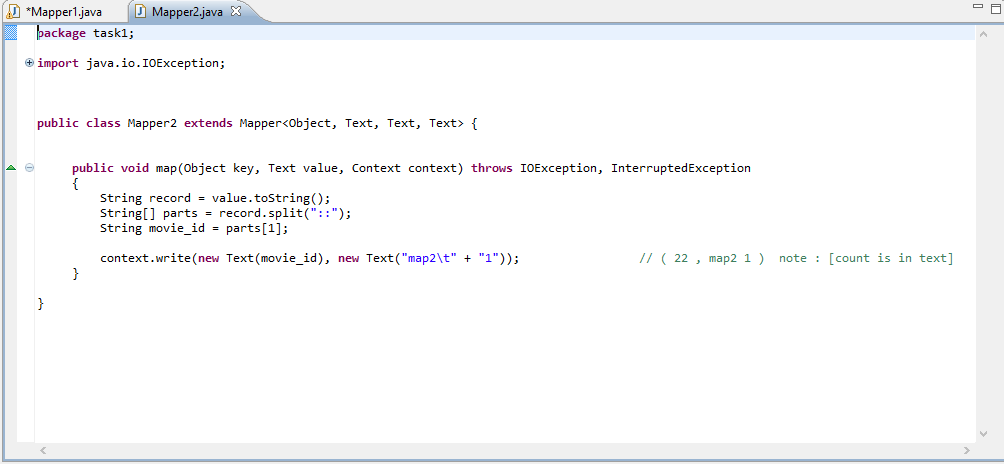
Mapper1:



We are splitting the line by using “::” delimiter and storing the values in a String Array so that all the columns in a row are stored in the string array. Then we are taking a condition if we have the string array length greater than 1 which means if the line or row has at least 1columns then it will enter into the if condition and execute the code to eliminate the **ArrayIndexOutOfBoundsException.**

  We are storing the movie id and movie name into the String. We are writing the key and value into the *context* which will be the output of the map method.

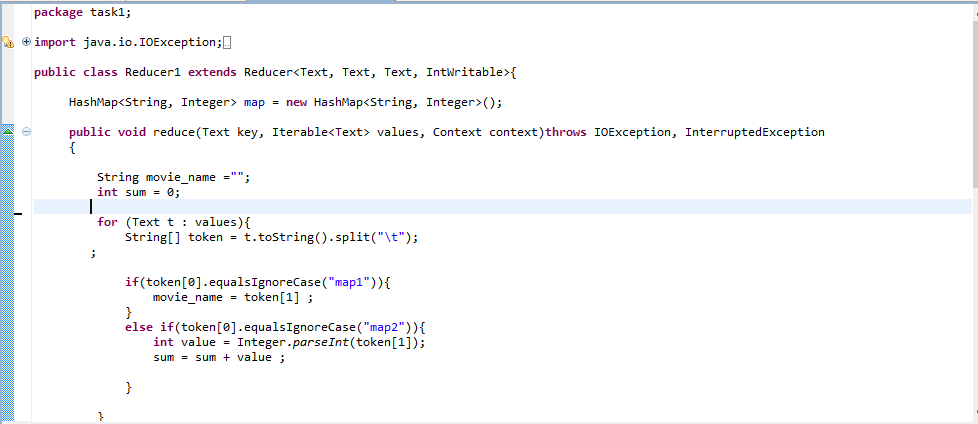
Mapper2:

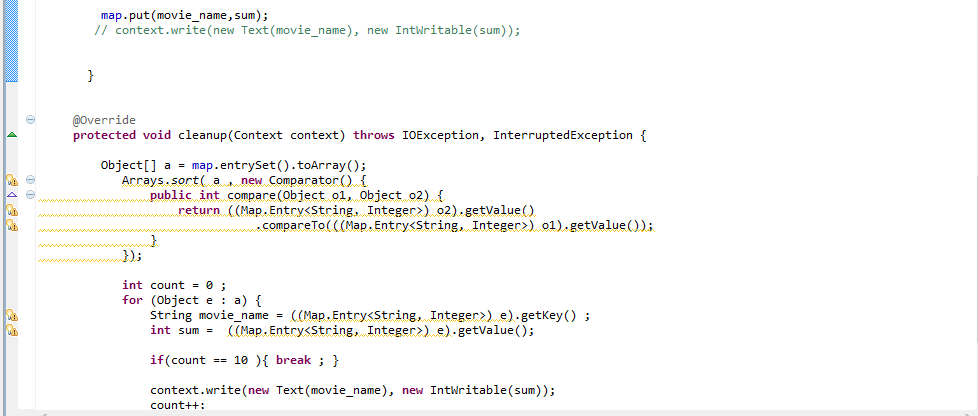


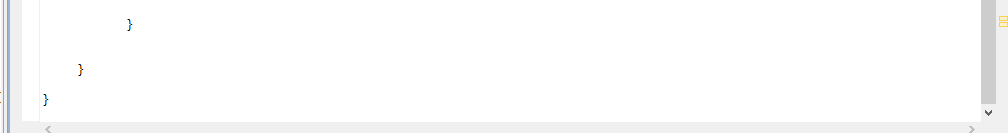
We are splitting the line by using “::” delimiter and storing the values in a String Array so that all the columns in a row are stored in the string array. Then we are taking a condition if we have the string array length greater than 1 which means if the line or row has at least 1columns then it will enter into the if condition and execute the code to eliminate the **ArrayIndexOutOfBoundsException.**

  We are storing the movie id into the String. We are writing the key and value into the *context* which will be the output of the map method.

Reducer:







Hashmap is used to sort the result and to limit the result, in this project reduce join is performed.

First mapper will produce 🡪 (movie id and movie name)

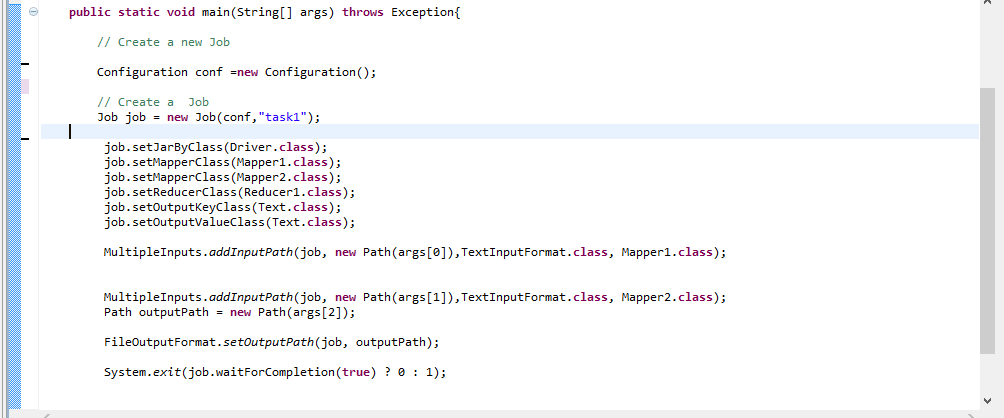
Second mapper will produce 🡪 (movie id and count)

When we pass the values, according to the match (map1 or map2) it will branch and do its described job. Then we store the result in map rather than direct writing using context.

***Clean method*** is written to perform the sorting the result according to the key. “Count” key is used to limit the 10 value, after the count reaches to 10 it will come out from the loop.

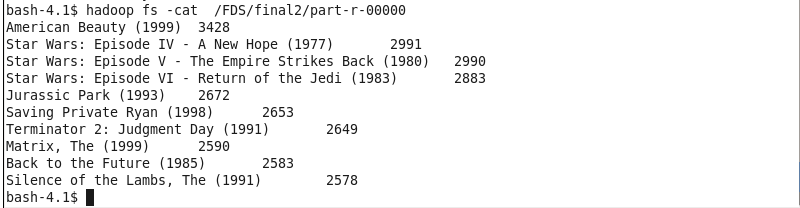
Finally we are storing Key = movie name and value = movie \_ count using content writer;

Driver:



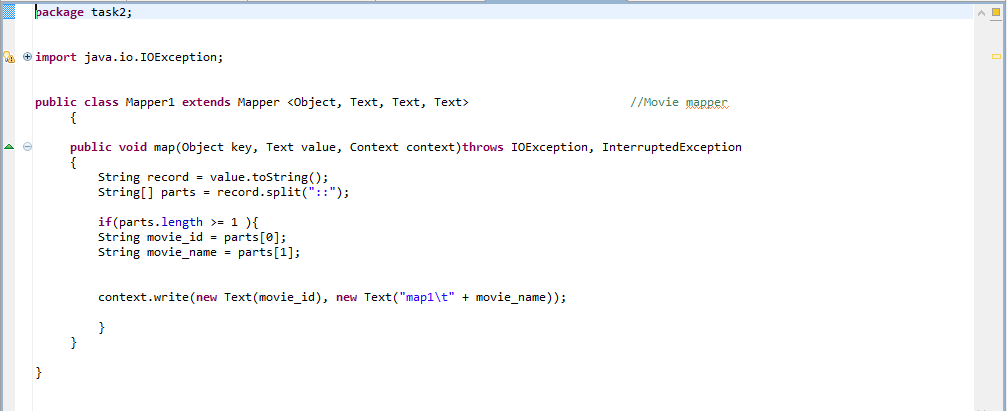
Multiple input is used, where arrg[0] is for mapper1 input and arrg[1] is for mapper2 finally arrg[3] is to store the result.

Result:



**Task2: *Top twenty rated movies (Condition: The movie should be rated/viewed by at least 40 users)***

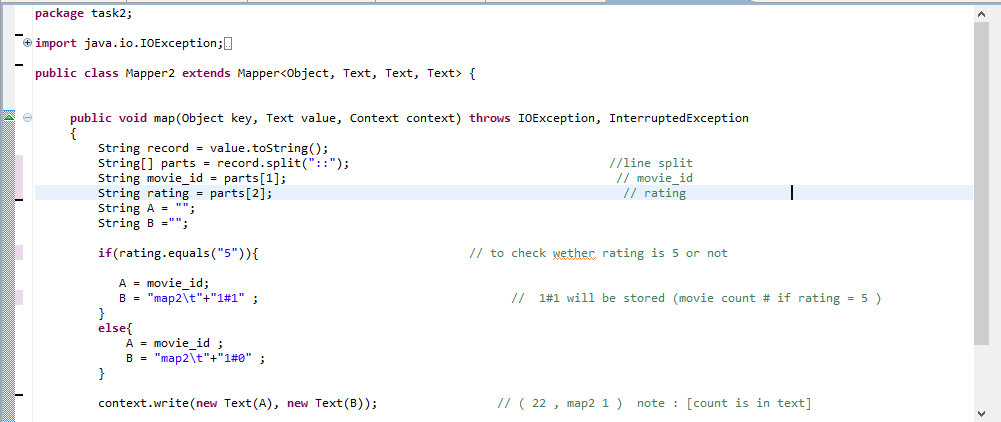
**Mapper1:**



We are splitting the line by using “::” delimiter and storing the values in a String Array so that all the columns in a row are stored in the string array. Then we are taking a condition if we have the string array length greater than 1 which means if the line or row has at least 1columns then it will enter into the if condition and execute the code to eliminate the **ArrayIndexOutOfBoundsException.**

  We are storing the movie id into the String. We are writing the key and value into the *context* which will be the output of the map method.

**Mapper2:**

****

****

We are splitting the line by using “::” delimiter and storing the values in a String Array so that all the columns in a row are stored in the string array. Then we are taking a condition if we have the string array length greater than 1 which means if the line or row has at least 1columns then it will enter into the if condition and execute the code to eliminate the **ArrayIndexOutOfBoundsException.**

We are storing the movie id and count into the String. We are writing the key and value into the *context* which will be the output of the map method.

**if**(rating.equals("5")){

A = movie\_id;

B = "map2\t"+"1#1" ; }

**Else**

{

A = movie\_id ;

B = "map2\t"+"1#0" ;

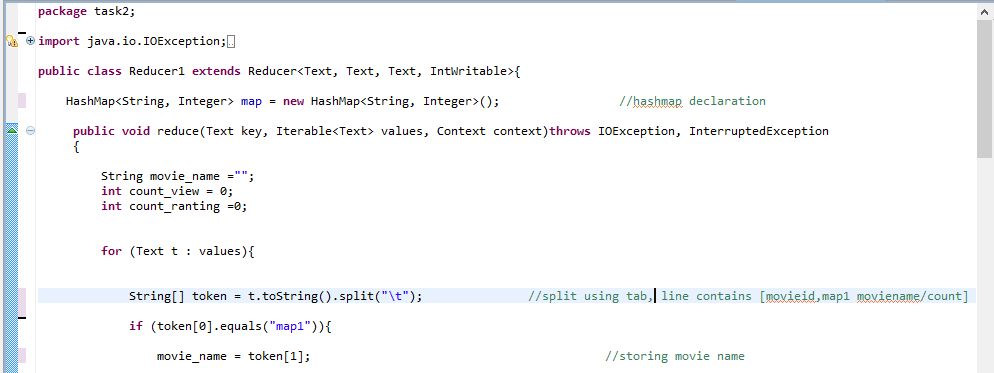
}

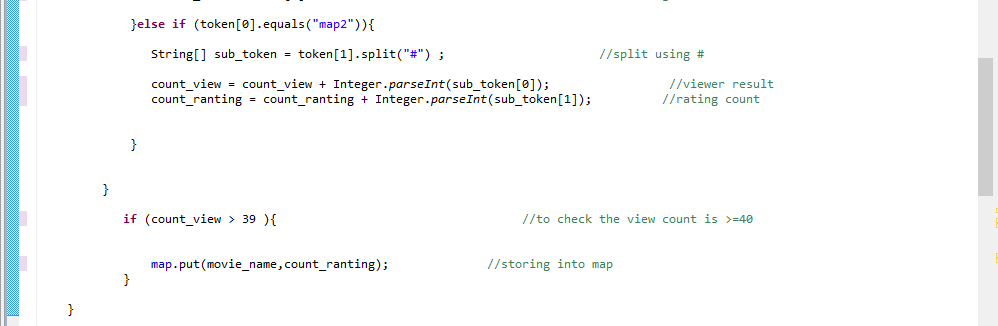
A stores the movie id;

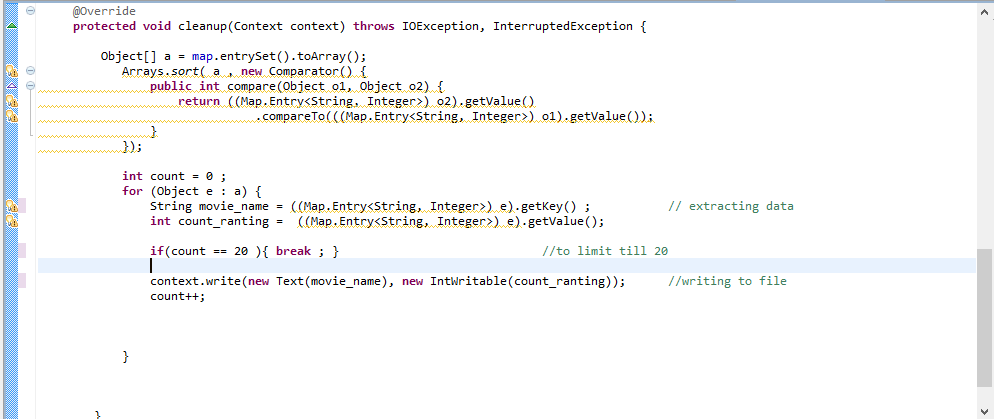
B Stores the count i.e. condition is movie should be rated/viewed by at least 40 and to count the high rated movies (hence I’m just passing the high rated movies in the loop i.e. considering only 5 rated movie)

(1#0) 🡪 (Viewed count # (if rating is 5 than 1 else 0));

Reducer:



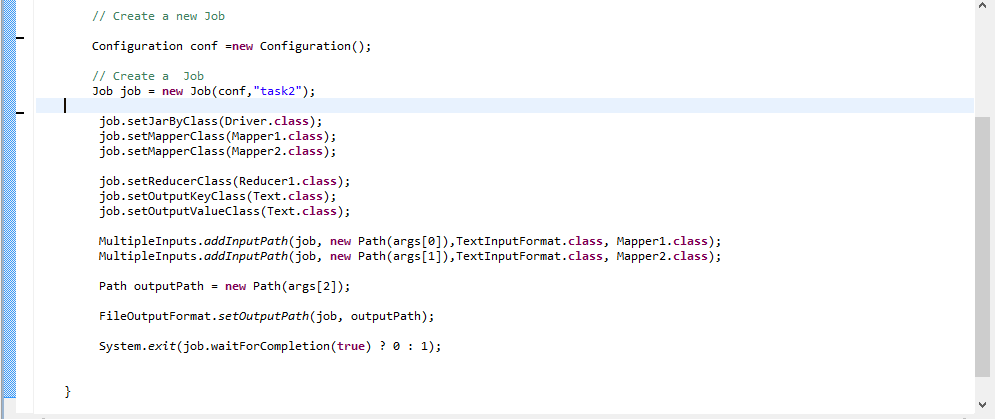




***Clean method*** is written to perform the sorting the result according to the key. “Count” key is used to limit the 10 value, after the count reaches to 10 it will come out from the loop.

Finally we are storing Key = movie name and value = movie \_ count using content writer;

Driver:



Result:

