Project: Data Analysis of Movie Genres and Directors by Country

1. Briefly describe the data and show a few lines of actual data in your report.
   There are 2 sources of our data:

   A. IMDB.com

   Most of our data comes from https://www.imdb.com/interfaces/, which contains various information related to movies and TV shows. It is stored in TSV format and consists of multiple files. Each file has a different number of records, from 1 million to 35 million, and in a total of approximately 109 million records. Each file has a different number of attributes, with the lowest number of 3, the highest 9. Properties include title, region, type, year, rating, director, etc. The dataset generally spans from the early 1900s to the present day. Some titles may have information dating back to the late 1800s.

   B. OMDb API

   There is no country information for each movie in the IMDB data however, so we have to pull the country tag of each movie from https://www.omdbapi.com/. We wrote a Java program to

   I. get the data from the OMDb API and
   II. performs data cleaning on the data to normalize country names so that different variations of the same country are represented consistently.

   (Source code: GetCountry.java)

   There are 5 tables in our dataset:

   - Basics: the basic information of movies, such as: name, type, genres, etc.
   - Crew: the crew of each movie, including directors and writers

   NOTE: the directors and writers here are stored as name id instead of human name.

   - Ratings: the rating of each movie, including average rating and number of votes
   - Countries: the country tag of each movie
   - Names: the information of each person, such as name, birthday, etc.

   Below is a diagram for the tables in our dataset, the column with a green arrow means the same object can be tracked between 2 tables using this column; for example, if we want to know the directors of a movie in table *basics*, we can use its title id to retrieve table *crew* to find that out.

To help get a better view of our dataset, below are data snippets for each table:

```
select * from basics limit 10;
```

	titleid	titletype	primarytitle	originaltitle	isadult	startyear	endyear	runtimeminutes	genres
1	tt0000001	short	Carmencita	Carmencita	0	1894	<null>	1	["Documentary","Short"]
2	tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	0	1892	<null>	5	["Animation","Short"]
3	tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892	<null>	4	["Animation","Comedy","Romance"]
4	tt0000004	short	Un bon bock	Un bon bock	0	1892	<null>	12	["Animation","Short"]
5	tt0000005	short	Blacksmith Scene	Blacksmith Scene	0	1893	<null>	1	["Comedy","Short"]
6	tt0000006	short	Chinese Opium Den	Chinese Opium Den	0	1894	<null>	1	["Short"]
7	tt0000007	short	Corbett and Courtney Before the Kinetograph	Corbett and Courtney Before the Kinetograph	0	1894	<null>	1	["Short","Sport"]
8	tt0000008	short	Edison Kinetoscopic Record of a Sneeze	Edison Kinetoscopic Record of a Sneeze	0	1894	<null>	1	["Documentary","Short"]
9	tt0000009	movie	Miss Jerry	Miss Jerry	0	1894	<null>	45	["Romance"]
10	tt0000010	short	Leaving the Factory	La sortie de l'usine Lumière à Lyon	0	1895	<null>	1	["Documentary","Short"]

```
select * from crew limit 10;
```

	titleid	directors	writers
1	tt0000001	["nm0005690"]	<null>
2	tt0000002	["nm0721526"]	<null>
3	tt0000003	["nm0721526"]	<null>
4	tt0000004	["nm0721526"]	<null>
5	tt0000005	["nm0005690"]	<null>
6	tt0000006	["nm0005690"]	<null>
7	tt0000007	["nm0005690","nm0374658"]	<null>
8	tt0000008	["nm0005690"]	<null>
9	tt0000009	["nm0085156"]	["nm0085156"]
10	tt0000010	["nm0525910"]	<null>

```
select * from ratings limit 10;
```

	titleid	rating	numvotes
1	tt0000001	5.7	1966
2	tt0000002	5.8	263
3	tt0000003	6.5	1803
4	tt0000004	5.6	179
5	tt0000005	6.2	2603
6	tt0000006	5.1	178
7	tt0000007	5.4	817
8	tt0000008	5.4	2100
9	tt0000009	5.3	204
10	tt0000010	6.9	7094

```
select * from countries limit 10;
```

	titleid	country
1	tt0001628	["United States"]
2	tt0002431	["France"]
3	tt0001051	["Spain"]
4	tt0002031	["United States"]
5	tt0002001	["United States"]
6	tt0002089	["United Kingdom"]
7	tt0002514	["Sweden"]
8	tt0002211	["Denmark"]
9	tt0001184	["Spain"]
10	tt0002333	["Germany"]

```
select * from names limit 10;
```

	nameid	primaryname	birthyear	deathyear	primaryprofession	knownfortitles
1	nm0000001	Fred Astaire	1899	1987	["soundtrack","actor","miscellaneous"]	["tt0045537","tt0050419","tt0072308","tt0053137"]
2	nm0000002	Lauren Bacall	1924	2014	["actress","soundtrack"]	["tt0117057","tt0075213","tt0037382","tt0038355"]
3	nm0000003	Brigitte Bardot	1934	<null>	["actress","soundtrack","music_department"]	["tt0057345","tt0056404","tt0054452","tt0049189"]
4	nm0000004	John Belushi	1949	1982	["actor","soundtrack","writer"]	["tt0080455","tt0078723","tt0072562","tt0077975"]
5	nm0000005	Ingmar Bergman	1918	2007	["writer","director","actor"]	["tt0050976","tt0083922","tt0050986","tt0060827"]
6	nm0000006	Ingrid Bergman	1915	1982	["actress","soundtrack","producer"]	["tt0036855","tt0038787","tt0034583","tt0038109"]
7	nm0000007	Humphrey Bogart	1899	1957	["actor","soundtrack","producer"]	["tt0042593","tt0043265","tt0037382","tt0034583"]
8	nm0000008	Marlon Brando	1924	2004	["actor","soundtrack","director"]	["tt0078788","tt0047296","tt0070849","tt0068646"]
9	nm0000009	Richard Burton	1925	1984	["actor","soundtrack","producer"]	["tt0061184","tt0057877","tt0087803","tt0059749"]
10	nm0000010	James Cagney	1899	1986	["actor","soundtrack","director"]	["tt0031867","tt0029870","tt0035575","tt0042041"]

2. Briefly describe each task you have completed so far or have spent a significant amount time on.
We have finished the data processing and completed 2 major tasks using Hive.

A. Load data

We wrote a HQL script to create and populate the movie database with five tables: basics, ratings, crew, names, and countries. The HQL script loads the data into each table from the respective TSV files. We store the tsv files on HDFS to leverage the distributed Hadoop file system to speed up the data loading.

Source code: load_data.hql

B. Major task 1: Determine the top 3 rated genres for each country using Hive

We wrote a HQL script to find the top 3 rated genres for each country based on their weighted average ratings. It involves joining three tables containing movie data, filtering out the relevant rows, and processing the data to calculate the weighted average rating for each genre-country

combination. Finally, the results are ranked, filtered, and inserted into a table called "top_genre" to store the top 3 genres for each country.

Source code: top-genre.hql

Pseudo code:

```
create top_genre table
create top_genre_temp view
    join basics, countries, and ratings tables
    filter relevant rows
    explode genres and countries columns
    calculate weighted average rating for each genre-country combination
    rank genres for each country based on their weighted average ratings
    filter top 3 ranked genres for each country
insert result into top_genre table
drop top_genre_temp view
```

Sample output:

country	genre	weightedaverating	ranking
China	Romance	6.5	1
China	Animation	6.5	1
China	Documentary	5.90814757878555	3
India	Animation	5.6	1
Japan	Romance	6.2	1
Japan	Documentary	5.8	2
Japan	Animation	5.8	2
United States	Documentary	5.7	1
United States	Animation	5.6	2

C. Major task 2: Determine the most successful director for each genre for each country using Hive

We wrote a Hive SQL script to get the best director(s) for each genre in each country.

I. First, we join the tables *basics*, *ratings*, *countries* and *crew* by movie title id and then join with table *names* to get the name of director;

II. Then we group the data by country, genre and director, and calculate the weighted rating by sum(rating * num of votes) / sum(num of votes);

III. Finally, we rank the directors by their weighted rating and return the director(s) with the highest rating.

Source code: best_director.hql

Pseudo code:

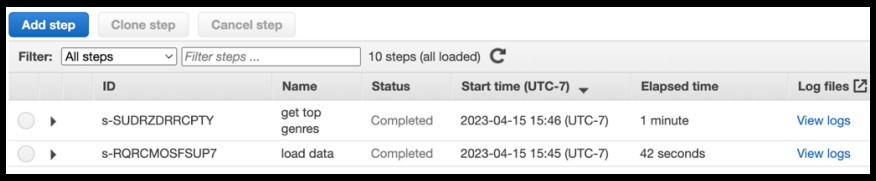
```
create best_director table
create best_director_tmp view
    join basics, countries, ratings and crew
    filter out noisy records
    explode genres, countries and directors which are array
    calculate weighted average rating group by country, genre and director
    rank directors for each genre in each country by the weighted rating
    select directors whose rank = 1 as the results
insert result into best_director table
drop best_director_tmp view
```

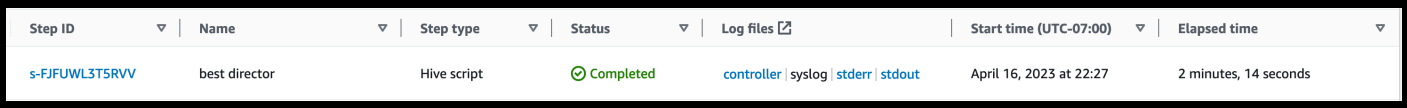
Sample output:

country	genre	director_name	weighted_rating
China	Action	Shuanbao Wang	9.4
China	Adventure	Chun-Hsien Wu	9.1
China	Animation	Pin Pin Tan	9.4
China	Biography	Shiwei Kang	9.8
China	Comedy	Peter Farrelly	8.2
China	Comedy	Gil Kofman	8.2
China	Comedy	Tanner King Barklow	8.2
China	Crime	Xu Jiang Hua	8.6
China	Documentary	Shiwei Kang	9.8
China	Drama	Shiwei Kang	9.8
China	Family	Alex Davidson	9.7
China	Fantasy	Xu Jiang Hua	8.6
China	History	Tiemu Jin	9.6
China	Horror	Eric Heise	8.2
China	Music	Han Niu	8.6
China	Musical	Hao Wu	7.3
China	Musical	Michael McFadden	7.3

D. Run on AWS EMR

We've configured an AWS EMR cluster with 1 master node (m5.xlarge) and 10 core nodes (m4.large) to execute Hive scripts. We've ensured that the data is successfully loaded into the specified Hive tables with the uploaded data files in S3 bucket. The expected results are also achieved and stored in a new table. Additionally, the precise running time of the program is also retrieved by analyzing the log files generated during the execution of the scripts on the EMR cluster.





Run time:

- Load data:

```
2023-04-15T21:44:30.538Z INFO Step created jobs:
2023-04-15T21:44:30.538Z INFO Step succeeded with exitCode 0 and took 40 seconds
```

- Top genre:

```
INFO total process run time: 62 seconds
2023-04-15T21:51:12.650Z INFO Step created jobs:
2023-04-15T21:51:12.650Z INFO Step succeeded with exitCode 0 and took 62 seconds
```

- Best director:

```
2023-04-17T05:29:37.123Z INFO Step created jobs:
2023-04-17T05:29:37.123Z INFO Step succeeded with exitCode 0 and took 134 seconds
```

3. Briefly discuss any major problems you ran into, which required major adjustments to your original proposal.

   We didn't encounter any problem that may result in major modifications on our proposal, the dataset quality is good and we have completed our major data analysis tasks using Hive.

4. Briefly discuss the remaining tasks that you will work on until the end of the semester.

   A. Combining the outputs from the two major tasks

      We will write code to merge the data from the two tasks and perform any necessary calculations or analysis to answer the final question. For instance, we will identify the most successful directors for the most popular movie genres in each country.

   B. Implementing the same logic using Pig Latin

      We will use Pig Latin to perform the same analysis as in Task A. This will enable us to compare the performance between Pig Latin and Hive.

   C. Analyzing the performance of each program

      Finally, we will evaluate and compare the performance of the two programs by measuring the execution time and resource utilization. This will help us determine the best tool for processing large datasets and performing complex queries.