

Name: Mahedi Hasan Mridul

Student Number: 632901

Assignment 1:

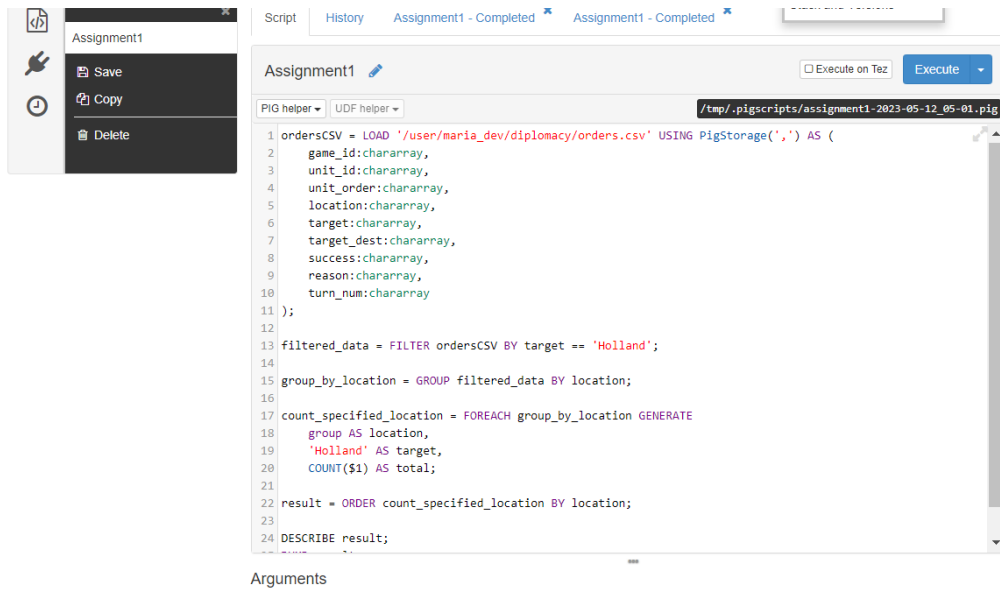
Github Url: https://github.com/mridul-max/Hadoop_Assignments

Project 7. Diplomacy done on Apache Pig [#2]

		Example result
Assessment	<ul style="list-style-type: none">Do research on Pig	
	<ul style="list-style-type: none">Make a alphabetic list from all locations from the orders.csv.Group by "location" with target "Holland"Count how many times Holland was the target from that locationCode is executed from Pig View	Adriatic Sea , Holland, 6 Albania, Holland, 5 etc...
	<ul style="list-style-type: none">Make a list how many times every country won.	(T,6000) (R,5000) etc..
Dataset needed	Diplomacy	
Hadoop system	Apache Pig	

Steps to execute Pig Script:

- I Have Hadoop configured and up and running according to slides of 2nd lesson.
- In Hadoop Files view, upload the required CSVs : Moodle -> diplomacy dataset: orders.csv.
- In the Pig view, create a new script by clicking on the "New Script" button.
- Either paste the script provided earlier into the script editor field or upload the script file.
- Review the script and make the necessary modifications or changes to the wording as per your requirement. For example, you can update the target location from 'Holland' to a different location that you want to filter and count.
- Once you have made the changes, click on the "Run" or "Execute" button to execute the script.
- Monitor the progress and wait for the script to complete. The results will be displayed in the Pig view.



Loading and Filtering the Data:

In this part, the `LOAD` statement is used to load the `orders.csv` file, which is located at `/user/maria_dev/diplomacy/orders.csv`. The file is loaded using `PigStorage(',')`, indicating that it is a comma-separated values file.

The `AS` clause is used to assign names to the fields in the loaded data. Each field is assigned a name and a data type, such as `game_id: chararray`, `unit_id: chararray`, etc.

The `FILTER` statement is then used to filter the loaded data. In this case, it filters the `ordersCSV` relation and keeps only the records where the `target` field is equal to the string "Holland"

```
ordersCSV = LOAD '/user/maria_dev/diplomacy/orders.csv' USING PigStorage(',') AS (  
    game_id: chararray,  
    unit_id: chararray,  
    unit_order: chararray,  
    location: chararray,  
    target: chararray,  
    target_dest: chararray,  
    success: chararray,  
    reason: chararray,  
    turn_num: chararray  
);
```

Grouping and Counting:

the GROUP statement is used to group the filtered_data relation based on the location field. This creates a new relation, group_by_location, where each group contains records with the same location.

The FOREACH statement is then used to process each group. In this case, it generates a new relation, count_specified_location, where each record consists of the location, the string literal "Holland" as the target, and the count of records in each group (COUNT(\$1)). \$1 represents the bag of records associated with each group.

```
filtered_data = FILTER ordersCSV BY (target == "Holland");
```

```
group_by_location = GROUP filtered_data BY location;
```

```
count_specified_location = FOREACH group_by_location GENERATE  
  group AS location,  
  "Holland" AS target,  
  COUNT($1) AS total;
```

Ordering and Displaying the Result:

In the last part, the ORDER statement is used to order the count_specified_location relation by the location field in ascending order.

The DESCRIBE statement is used to describe the schema of the result relation, showing the field names and types.

Finally, the DUMP statement is used to display the contents of the result relation, which includes the ordered locations, the target ("Holland"), and the total count for each location.

```
result = ORDER count_specified_location BY location;
```

```
DESCRIBE result;
```

```
DUMP result;
```

Results:

Assignment1

Save

Copy

Delete

Script

History

Assignment1 - Completed

Assignment1 - Completed

Assignment1 - COMPLETED

Job ID: job_1683908430535_0050

Started: 2023-05-12 22:00

Results

Download

```
result: {location: chararray,target: chararray,total: long}
("Adriatic Sea","Holland",1)
("Aegean Sea","Holland",5)
("Albania","Holland",1)
("Armenia","Holland",1)
("Baltic Sea","Holland",326)
("Barents Sea","Holland",38)
("Belgium","Holland",35134)
("Berlin","Holland",1282)
("Black Sea","Holland",3)
("Bohemia","Holland",5)
("Brest","Holland",32)
("Budapest","Holland",1)
("Bulgaria","Holland",2)
("Burgundy","Holland",1153)
("Clyde","Holland",19)
("Constantinople","Holland",4)
("Denmark","Holland",4051)
("Eastern Mediterranean","Holland",4)
("Edinburgh","Holland",3023)
("English Channel","Holland",1231)
("Finland","Holland",2)
("Galicia","Holland",1)
("Gascony","Holland",6)
("Greece","Holland",3)
("Gulf of Bothnia","Holland",2)
("Gulf of Lyons","Holland",3)
("Helgoland Bight","Holland",9107)
("Holland","Holland",325)
```