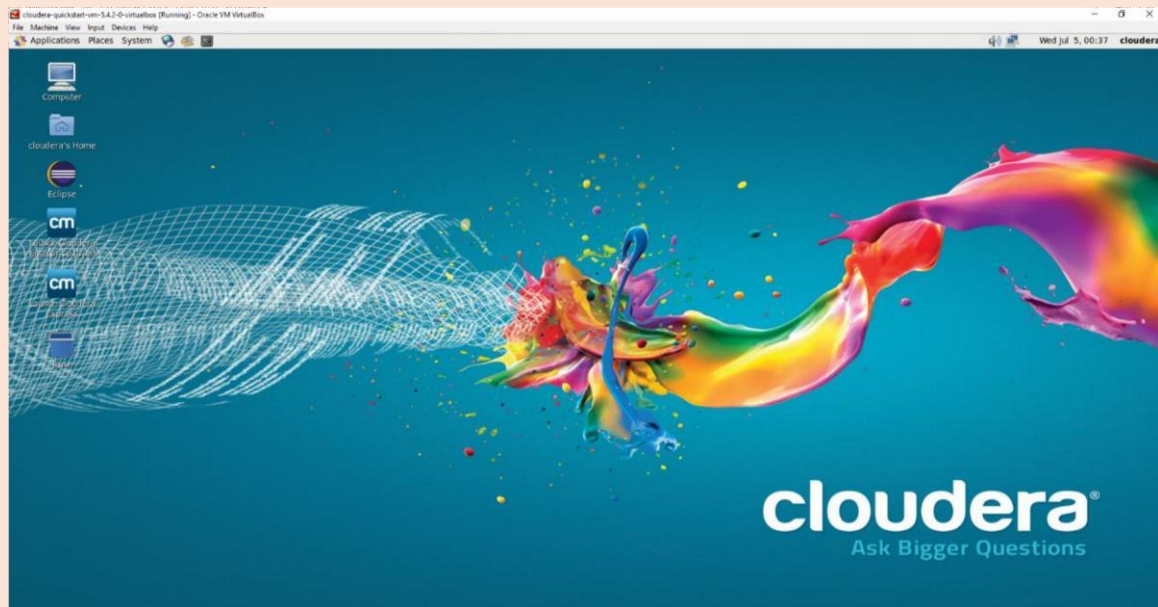


Name	Sufyan Ahmad
Roll No	201980013
Task	05
Course	Big Data Analytics
Section	A

TASK

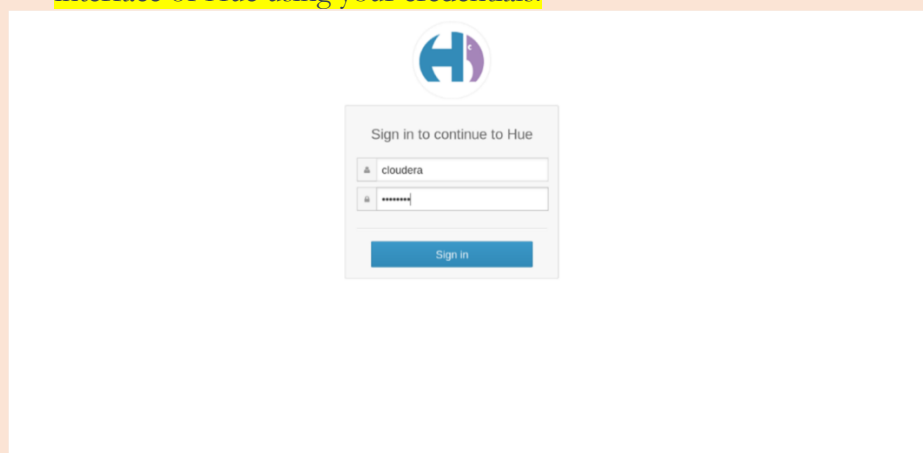
Execute Cloudera or initiate the running of Cloudera:

In this step, you will execute or initiate Cloudera to start running and functioning



Access the Hue interface or log in to Hue by providing your credentials:

In this step, you will log in to Hue, which involves accessing the user interface or interface of Hue using your credentials.



Transfer or import the input text file to the designated location or system:

In this step, you will upload text file containing Titanic works, which involves transferring the file to the system or platform you are using.

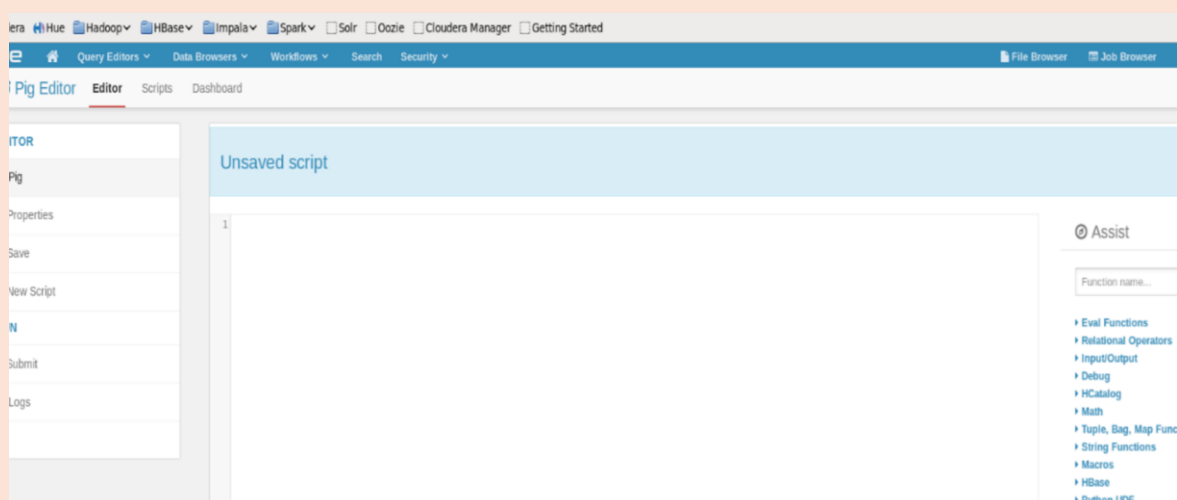
Name	Size	User	Group	Permissions	Date
↑		hdfs	supergroup	drwxr-xr-x	July 07, 2023 06:08 AM
↓		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:51 AM
TitanicData.txt	59.7 KB	cloudera	cloudera	-rw-r--r--	July 07, 2023 06:01 AM
...		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:02 AM

Navigate to the query editor or access the query editing interface:

In this step, you will access the Pig query prompt, which involves launching or opening the interface or environment where you can execute Pig queries

Compose the code and execute or run it:

In this step, you will compose the query and execute the code, which involves writing the desired Pig query and initiating the execution of the code to process and analyze the data.



Step 1:

Load the Titanic dataset The code begins by loading the Titanic dataset from the file '/user/cloudera/TitanicData.txt' using PigStorage(',') which assumes the data is commaseparated. It defines the schema of the dataset with columns such as PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, and Embarked.

```
titanic = LOAD '/user/cloudera/TitanicData.txt' USING PigStorage(',') AS (  
    PassengerId: int,  
    Survived: int,  
    Pclass: int,  
    Name: chararray,  
    Sex: chararray,  
    Age: float,  
    SibSp: int,  
    Parch: int,  
    Ticket: chararray,  
    Fare: float,  
    Cabin: chararray,  
    Embarked: chararray  
);
```

Step 2:

Calculate the average fare paid by passengers in each passenger class The code then uses the GROUP BY clause to group the 'titanic' relation by the 'Pclass' column. Inside the FOREACH statement, it calculates the sum of fares for each group using SUM(titanic.Fare) and counts the number of records in each group using COUNT(titanic). The result is stored in the 'fare_avg' relation, where each record contains the passenger class ('Pclass') and the average fare ('AvgFare').

```
fare_avg = FOREACH (GROUP titanic BY  
Pclass) {    fare_sum = SUM(titanic.Fare);  
fare_count = COUNT(titanic);    GENERATE
```

```
    group AS Pclass,  
    fare_sum / fare_count AS  
    AvgFare; }
```

Step 3:

Store and display the average fare results The 'fare_avg' relation is stored in the Hadoop Distributed File System (HDFS) with the name 'avg_fare_by_class1' using PigStorage(',') to specify comma-separated values. Then, the contents of the 'fare_avg' relation are displayed using the DUMP statement.

```
STORE fare_avg INTO 'avg_fare_by_class1' USING PigStorage(',');  
DUMP fare_avg;
```

Step 4:

Count the number of passengers embarked from each port Similarly to Step 2, the code groups the 'titanic' relation by the 'Embarked' column and calculates the count of passengers in each group using COUNT(titanic). The result is stored in the 'embarked_count' relation, which contains the port of embarkation ('Embarked') and the passenger count ('PassengerCount').

```
embarked_count = FOREACH (GROUP titanic BY  
Embarked) {    passenger_count = COUNT(titanic);  
GENERATE  
    group AS Embarked,  
    passenger_count AS  
    PassengerCount; }
```

Step 5:

Store and display the passenger count results by embarkation port The 'embarked_count' relation is stored in HDFS with the name 'passengers_by_embarked' using PigStorage(','), and then the contents of the 'embarked_count' relation are displayed using the DUMP statement.

```
STORE embarked_count INTO 'passengers_by_embarked' USING  
PigStorage(','); DUMP embarked_count;
```

Step 6:

Calculate the total number of passengers in each age group. The code uses a FOREACH statement on the 'titanic' relation without grouping to process each record individually. It assigns an age group to each passenger based on their age using a CASE statement. The age groups are defined from 0-9, 10-19, 20-29, and 'Unknown' for other age values. The result is stored in the 'age_grouped' relation, which contains the 'age_group' column.

```
age_grouped = FOREACH titanic {  
    age_group = CASE  
        WHEN Age >= 0 AND Age <= 9 THEN '0-9'  
        WHEN Age >= 10 AND Age <= 19 THEN '10-19'  
        WHEN Age >= 20 AND Age <= 29 THEN '20-29'  
        -- Add more age groups as needed  
        ELSE 'Unknown'  
    END;  
    GENERATE age_group;  
}
```

Step 7:

Count the number of passengers in each age group. Using the GROUP BY clause, the code groups the 'age_grouped' relation by the 'age_group' column. Inside the FOREACH statement, it calculates the count of passengers in each age group using

COUNT(age_grouped). The result is stored in the 'age_group_count' relation, which contains the age group ('AgeGroup') and the passenger count ('PassengerCount').

```
age_group_count = FOREACH (GROUP age_grouped BY
age_group) { passenger_count = COUNT(age_grouped);
GENERATE
    group AS AgeGroup,
    passenger_count AS
    PassengerCount; }
```

Step 8:

Store and display the passenger count results by age group The 'age_group_count' relation is stored in HDFS with the name 'passengers_by_age_group' using PigStorage(','), and then the contents of the 'age_group_count' relation are displayed using the DUMP statement.

```
STORE age_group_count INTO 'passengers_by_age_group' USING PigStorage(',');
DUMP age_group_count;
```

task5a

```
1  -- Load the Titanic dataset
2  titanic = LOAD '/user/cloudera/TitanicData.txt' USING PigStorage(',') AS (
3      PassengerId: int,
4      Survived: int,
5      Pclass: int,
6      Name: chararray,
7      Sex: chararray,
8      Age: float,
9      SibSp: int,
10     Parch: int,
11     Ticket: chararray,
12     Fare: float,
13     Cabin: chararray,
14     Embarked: chararray
15 );
16 -- Calculate the average fare paid by passengers in each passenger class
17 fare_avg = FOREACH (GROUP titanic BY Pclass) {
18     fare_sum = SUM(titanic.Fare);
19     fare_count = COUNT(titanic);
20     GENERATE
21         group AS Pclass,
22         fare_sum / fare_count AS AvgFare;
23 }
24
25 -- Store the result in a relation and display its contents
26 STORE fare_avg INTO 'avg_fare_by_class1' USING PigStorage(',');
27 DUMP fare_avg;
28
29
30 -- Count the number of passengers embarked from each port
31 embarked_count = FOREACH (GROUP titanic BY Embarked) {
32     passenger_count = COUNT(titanic);
33     GENERATE
34         group AS Embarked,
35         passenger_count AS PassengerCount;
36 }
37 -- Store the result in a relation and display its contents
38 STORE embarked_count INTO 'passengers_by_embarked' USING PigStorage(',');
```


In this step, you will check the status of the running job, which involves monitoring and observing the progress and status of the job execution.

PyCharm | Editor | Scripts | Dashboard

EDITOR

OF Pyg

- Properties
- Save
- Share
- New Script
- RUN
- Submit
- Logs
- Copy
- Delete
-

Task4

Progress: 100%

```

2023-07-05 02:07:46,562 [InfoControl] INFO org.apache.hadoop.yarn.client.api.impl.AbstractContainerManager - CONTROLLING SERVICE MANAGER IS AVAILABLE AT 10.0.0.0:8030
2023-07-05 02:07:46,562 [InfoControl] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - File input path is deprecated. Instead, use FileUtils#addFileToClassPath()
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths to process : 1
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths to process : 1
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths (combined) to process : 1
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.hadoop.mapreduce.jobhistory - number of splits: 1
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.hadoop.mapreduce.jobhistory - Submitting tokens for job: job_1580548097219_0005
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.hadoop.mapreduce.jobhistory - Kind: YARN_MR_TOKEN, Service: token [(org.apache.hadoop.yarn.security.AMRTokenIdentifier@f1e0c4d0)]
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.hadoop.mapreduce.jobhistory - Kind: MR_DELEGATION_TOKEN, Service: token [2.0.0-1.0002, Issue: (owner=clusteradmin, remanence=0.001, realizer=0.001, issueDate=1580548097219, expiryDate=1580548097219+0002), sequenceNumber=14, state=Succeeded]
2023-07-05 02:07:56,587 [InfoControl] WARN org.apache.hadoop.mapreduce.v2.util.MRAppUtils - cache file (mapreduce.job.cache.files) https://github.com:8020/user/realizer/dmr/118.210.55000000300/pig/join-simile-1.jar conflicts with cache file (mapreduce.job.cache.files) https://github.com:8020/user/realizer/dmr/118.210.55000000300/pig/join-simile-1.jar
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl - Submitted application application_1580548097219_0005
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.hadoop.mapreduce.job - The url to track the job: http://github.com:8080/proxy/application_1580548097219_0005/
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - mapreduce.job_1580548097219_0005
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - Processing aliases D-C-DATA
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - detailed locations: M: [0x1,17,0x2,4],[0,4,1],[2,4,1],[2,4,1]; C: [0,4,1]
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - More information at: http://localhost:50030/jobdetails.py?jobid=job_1580548097219_0005
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - job
2023-07-05 02:07:56,587 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - job complete
2023-07-05 02:08:08,457 [InfoControl] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - job complete
Heart beat
2023-07-05 02:08:15,315 [communication Thread] INFO org.apache.hadoop.mapred.TaskAtteptListenerImpl - Progress of TaskAttept attempt_1580548097219_0005_u_000000_0 is : 1.0
    
```

Username

cloudera

Text

Search for text

Succeeded

Running

Failed

Killed

Logs	ID	Name	Status	User	Maps	Reduces	Queue	Priority	Duration	Submitted
	1688736915089_0008	PigLatin:script.pig	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	2m:6s	07/07/23 06:52:48
	1688736915089_0007	PigLatin:script.pig	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	1m:37s	07/07/23 06:50:55
	1688736915089_0006	PigLatin:script.pig	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	1m:36s	07/07/23 06:49:04
	1688736915089_0005	PigLatin:script.pig	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	1m:24s	07/07/23 06:47:20
	1688736915089_0004	PigLatin:script.pig	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	1m:25s	07/07/23 06:45:43
	1688736915089_0003	PigLatin:script.pig	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	1m:51s	07/07/23 06:43:34
	1688736915089_0002	oozie:launcher:T=pig;W=pig-app-hue-script:A=pig;ID=0000001-230707063725341-oozie-oozi-W	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	12m:2s	07/07/23 06:42:57
	1688736915089_0001	oozie:launcher:T=pig;W=pig-app-hue-script:A=pig;ID=0000000-230707063725341-oozie-oozi-W	SUCCEEDED	cloudera	100%	100%	root.cloudera	N/A	49s	07/07/23 06:41:21

Showing 1 to 8 of 8 entries

Previous


1


Next

Display Results


In this step, you will view the result and generate an output file, which involves displaying the desired outcome or output and creating a file to store the generated results


<input type="checkbox"/>	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	 f		hdfs	supergroup		July 07, 2023 06:08 AM
<input type="checkbox"/>	 .		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:51 AM
<input type="checkbox"/>	 TitanicData.txt	59.7 KB	cloudera	cloudera	-rw-r--r--	July 07, 2023 06:01 AM
<input type="checkbox"/>	 avg_fare_by_class		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:22 AM
<input type="checkbox"/>	 avg_fare_by_class1		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:45 AM
<input type="checkbox"/>	 oozie-oozi		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:55 AM
<input type="checkbox"/>	 passengers_by_age_group		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:52 AM
<input type="checkbox"/>	 passengers_by_embarked		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:48 AM

 Home / [user](#) / [cloudera](#) / [avg_fare_by_class1](#) / **part-r-00000**

Page of 1 

```
1,84.15468752825701
2,20.66218318109927
3,13.675550210257411
```

 Home / [user](#) / [cloudera](#) / [passengers_by_age_group](#) / **part-r-00000**

Page of 1 

```
0-9, 62
10-19, 102
20-29, 220
Unknown, 330
, 0
```

Home

/

user

/

cloudera

/

passengers_by_embarked

/

part-r-00000

Page 1 of 1

⏮

⏪

⏩

⏭

C,168
Q,77
S,644
,2