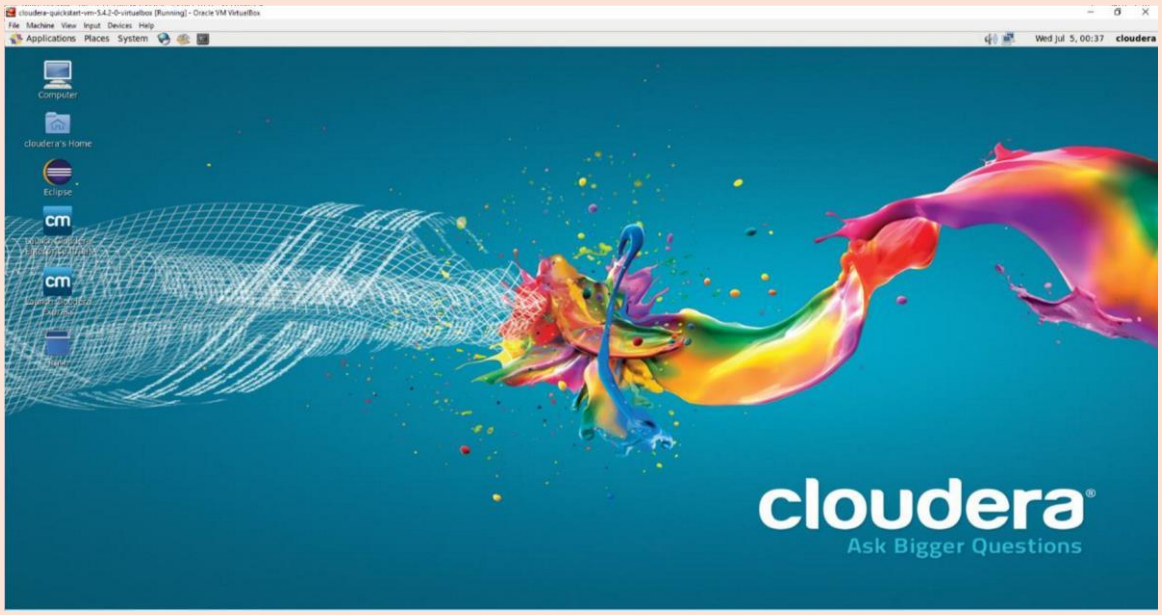
Name	**Sufyan Ahmad**
Roll No	**201980013**
Task	**05**
Course	**Big Data Analytics**
Section	**A**

TASK

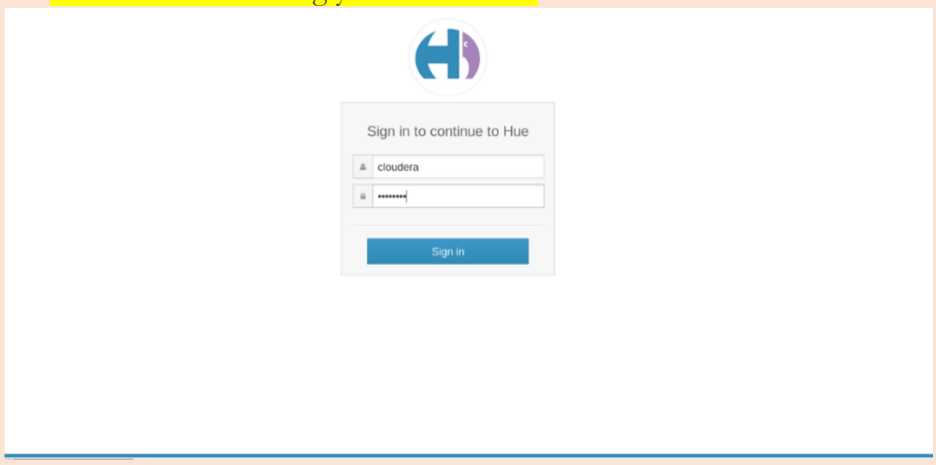
Execute Cloudera or initiate the running of Cloudera:

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



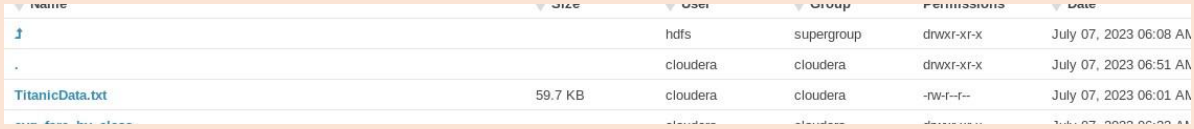
Access the Hue interface or log in to Hue by providing your credentials:

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 tanatnic works, which involves transferring the file to the system or platform you are using.



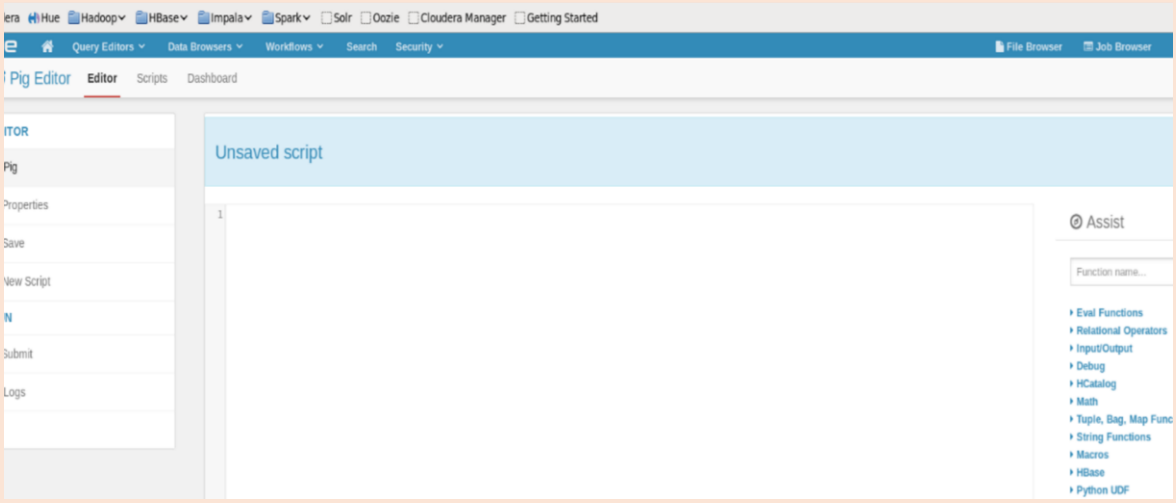
Name	Size	User	Group	Permissions	Date
..		hdfs	supergroup	drwxr-xr-x	July 07, 2023 06:08 AM
.		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:51 AM
TitanicData.txt	59.7 KB	cloudera	cloudera	-rw-r--r--	July 07, 2023 06:01 AM

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.

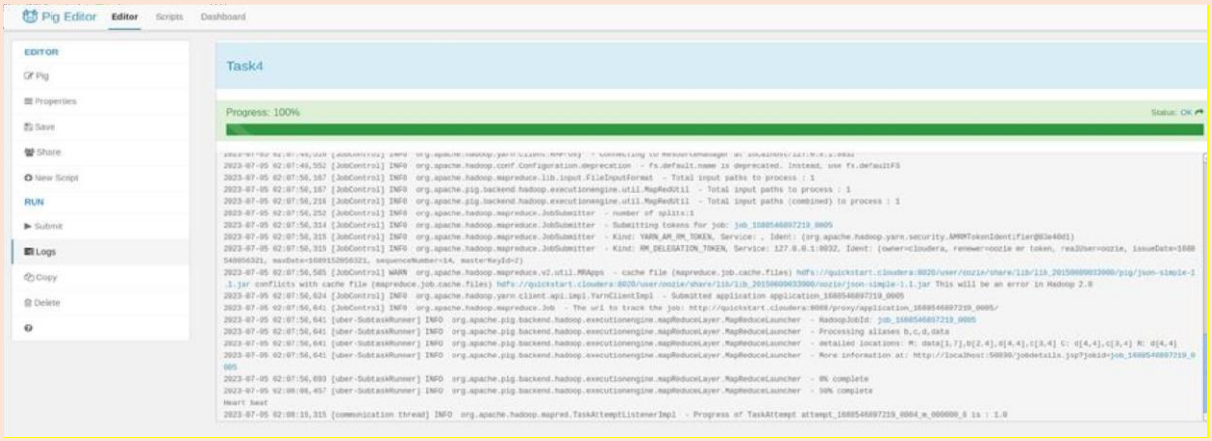
```
STORE age_group_count INTO 'passengers_by_age_group' USING PigStorage(',');
DUMP age_group_count;
```

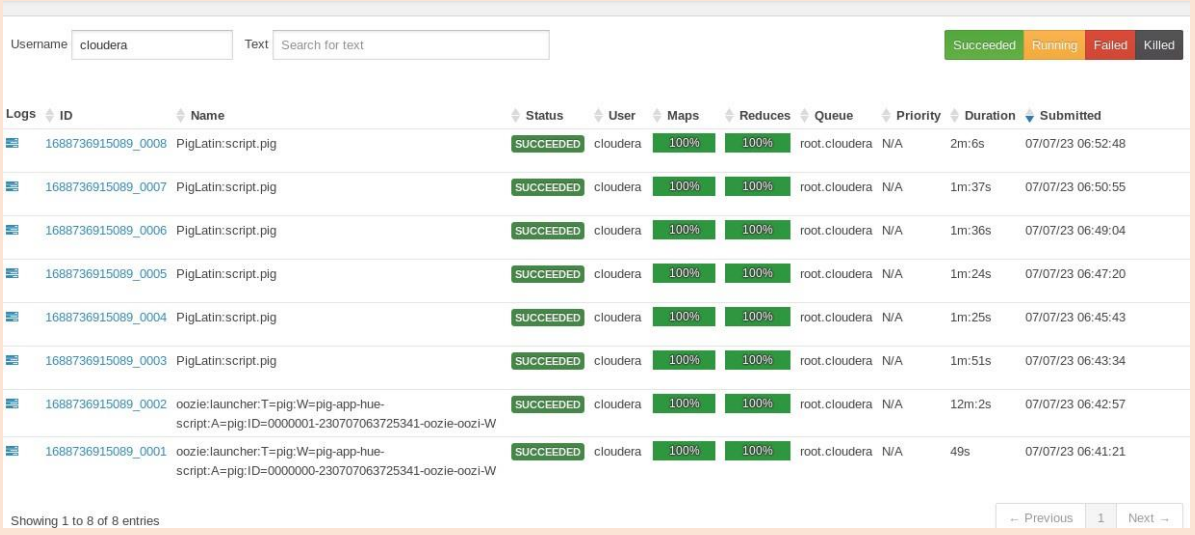
task5a

```
-- Load the Titanic dataset
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
);
-- Calculate the average fare paid by passengers in each passenger class
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;
}

-- Store the result in a relation and display its contents
STORE fare_avg INTO 'avg_fare_by_class1' USING PigStorage(',');
DUMP fare_avg;


-- Count the number of passengers embarked from each port
embarked_count = FOREACH (GROUP titanic BY Embarked) {
    passenger_count = COUNT(titanic);
    GENERATE
        group AS Embarked,
        passenger_count AS PassengerCount;
}
-- Store the result in a relation and display its contents
STORE embarked_count INTO 'passengers_by_embarked' USING PigStorage(',');
```

Check the status of the running job or monitor the progress of the job execution:

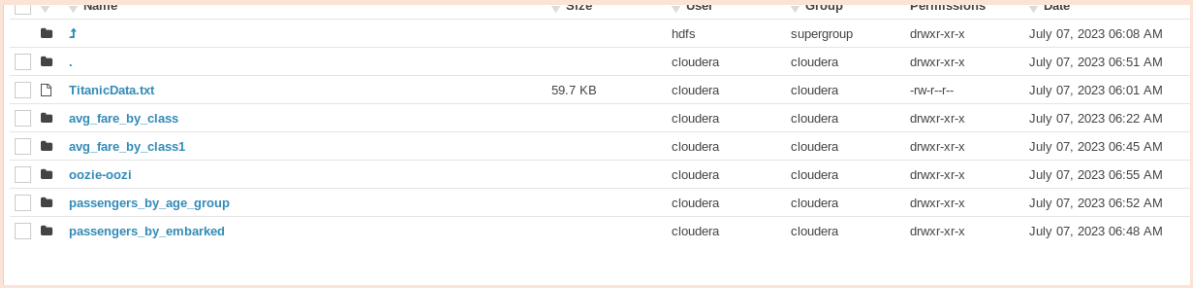
In this step, you will check the status of the running job, which involves monitoring and observing the progress and status of the job execution.



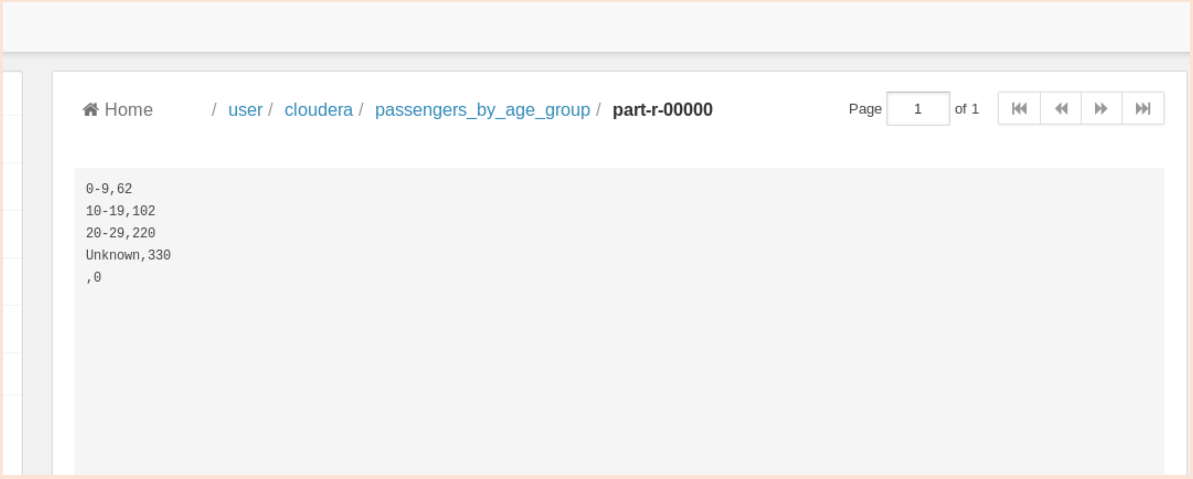


Display Results

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



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
avg_fare_by_class		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:22 AM
avg_fare_by_class1		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:45 AM
oozie-oozi		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:55 AM
passengers_by_age_group		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:52 AM
passengers_by_embarked		cloudera	cloudera	drwxr-xr-x	July 07, 2023 06:48 AM



Home / user / cloudera / passengers_by_embarked / **part-r-00000**

Page 1 of 1

```
C,168
Q,77
S,644
,2
```