Assignment 7.1(Apache Pig)

Task 1 :

To write a program for wordcount using pig script:

1. Write the below script using nano editor and save the file as wordcount.pig in local and move it to hdfs .
2. execute the pig script as below:

exec wordcount.pig in grunt shell as below:

grunt> exec wordcount.pig

wordcount.pig:

```
A = load '/hadoopdata/test.txt';
B = foreach A generate flatten(TOKENIZE((chararray)$0)) as word;
C = group B by word;
D = foreach C generate group , COUNT(B);
dump D;
```

Script Explanation:

1.A is used to loads the i/p file for counting the words in it.
2. B will will order all the words in the file in one single column.
3. C will will group the same words together in a tuple like (THis,{(THis),(THis)}).
4. D will will count the number of same words and creates a wordcount as (is,3) .
5. Dump will display the final output.

Output of wordcount:

```
: 1
(is,3)
(BDHS,3)
(THis,2)
(This,1)
(from,1)
(Lavanya,1)
(training.,3)
```

Task 2 :

Given 2 files namely employee_details(Empid, name,salary,Rating) and employee_expenses(Empid, Expenses) to write a pig script and execute in local mode

Local Mode is nothing but executing the pig script in local filesystem. both the i/p files and scripts are available in local filesystems.

a) To write a pig script to get the top 5 employees(empid, name) with highest ratings incase 2 employee has same rating , employee with name coming first in dictionary should be displayed.

Steps to create a pig file and execute in local mode:
----
1. Create a Emp_HighRating.pig file using nano editor,write pig scripts as highlighted in red below and save it in local.
2. Copy the input Files, Employee_details, Employee_expense to the local filesystem.
3. execute in local mode using below command:
pig -x local Emp_HighRating.pig

Emp_HighRating.pig script:

```
A = load  '/pig/employee_details.txt' using PigStorage(',');
B = foreach A generate (int)$0 as EmpId, (chararray)$1 as EmpName, (int)$3 as rating;
C = order B by $2  DESC, $1 ASC;
D = LIMIT C 5;
dump D;
```

Script Explanation:

1.A is used to load the input file which is delimited using ','
2. B will create something like a table with data in 0th column as Empid ,data in the 1st column as EmpName and data in 2nd column as salary.
3. C will arrange the data in 2nd column(rating) in descending order and data in 1st column(name) in ascending order.
4. D will will limit the data to top 5 employees with High Rating.
5. Dump will display the final output.

Output after running the script:
```
(105,Pawan,5)
(110,Priyanka,5)
(104,Anubhav,4)
(109,Katrina,4)
(103,Akshay,3)
```

b) To write a pig script to get the top 3 employees(empid, name) with highest salary whose employee id is an odd number.

Steps To create a pig file and execute in local mode:
----

1. Create a Emp_HighSalary.pig file using nano editor,write pig scripts as highlighted in red below and save it in local.
2. Copy the input Files, Employee_details, Employee_expense to the local filesystem.
3. execute in local mode using below command:
pig -x local Emp_HighSalary.pig

Emp_HighSalary.pig script:

```
A = load  '/pig/employee_details.txt' using PigStorage(',');
B = foreach A generate (int)$0 as EmpId, (chararray)$1 as EmpName, (int)$2 as salary;
C = order B by $2  DESC;
D = filter C by ($0%2 !=0);
E = LIMIT D 3;
dump E;
```

Script Explanation:

1.A is used to load the input file which is delimited using ','
2. B will create something like a table with data in 0th column as Empid ,data in the 1st column as EmpName and data in 2nd column as salary.
3. C will order the 2nd column salary in descending order..
4. D will filter the employees whose id is odd number.
5. E will limit the top 3 employees having high salary..
6. Dump will display the final output.

Output:

```
(101,Amitabh,20000)
(107,Salman,17500)
(103,Akshay,11000)
```

c) To write a pig script to get the employees(empid, name) with highest expenses.

Steps To create a pig file and execute in local mode:
----
1. Create a Emp_High_Expense.pig file using nano editor,write pig scripts as highlighted in red below and save it in local.
2. Copy the input Files, Employee_details, Employee_expense to the local filesystem.
3. execute in local mode using below command:
pig -x local Emp_High_Expense.pig

Emp_High_Expense.pig script:

```
A = load '/pig/employee_details.txt' using PigStorage(',');
B = foreach A generate (int)$0 as EmpId, (chararray)$1 as Name;
A1 = load '/pig/employee_expenses.txt' using PigStorage('\t');
B1 = foreach A1 generate (int)$0 as EmpId, (int)$1 as expense;
C1 = group B1 by EmpId;
D1 = foreach C1 generate group as EmpId, SUM(B1.expense) as tot_expense;
joined_table = join B by $0, D1 by EmpId;
final = foreach joined_table generate $0, $1, $3;
final_order = order final by $2 DESC, $1 ASC;
dump final_order;
```

Script Explanation:

1.A is used to load the input file which is delimited using ','
2. B will create something like a table with data in 0th column as Empid ,data in the 1st column as EmpName
3. A1 is used to load the another input file which is delimited using '\t'
4. B1 will create something like a table with data in 0th column as Empid ,data in the 1st column as Expense.
5. C1 will group B1 based on EmpId as tuple .
6. D1 forms a group as empId , summed up expense as 101, (200+100) = (101,300)..
7. joined_table joins both the table based on EmpId.
8. final will create a something like a table containing 0th column Id, 1st column EmpName and 3rd column as summed up Expense.
9. final_order will order the table based on salary in descending order followed by name in ascending order.
10. Dump will display the final output.

Output:

```
(102,Shahrukh,500)
(110,Priyanka,400)
(101,Amitabh,300)
(104,Anubhav,300)
(114,Madhuri,200)
(105,Pawan,100)
```

d) To write a pig script to get the employees(empid, name) which is available in expense_details.

Steps To create a pig file and execute in local mode:

----

1. Create a Emp_Details_InExpense.pig file using nano editor,write pig scripts as highlighted in red below and save it in local.
2. Copy the input Files, Employee_details, Employee_expense to the local filesystem.
3. execute in local mode using below command:
pig -x local Emp_Details_InExpense.pig

Emp_Details_InExpense.pig script:

```
A = load '/home/acadgild/employee_details.txt' using PigStorage(',');
B = foreach A generate (int)$0 as EmpId, (chararray)$1 as Name;
A1 = load '/home/acadgild/employee_expenses.txt' using PigStorage('\t');
B1 = foreach A1 generate (int)$0 as EmpId;
C1 = group B1 by EmpId;
D1 = foreach C1 generate group as EmpId;
joined_table = join B by $0 full, D1 by EmpId;
C = FILTER joined_table by $2 is not null ;
final = foreach C generate $2, $1;
dump final;
```

Script Explanation:

1.A is used to load the input file which is delimited using ','
2. B will create something like a table with data in 0th column as Empid ,data in the 1st column as EmpName
3. A1 is used to load the another input file which is delimited using '\t'
4. B1 will create something like a table with data in 0th column as Empid.
5. C1 will group B1 based on EmpId as tuple .
6. D1 works on tuple to form a group based on the empId.
7. joined_table joins both the table based on EmpId.
8. C filters the joined table such that expense tables empId is not NULL.
9. final creates a group with empId and EmpName.
10. Dump will display the final output.

Output:

```
(101,Amitabh)
(102,Shahrukh)
(104,Anubhav)
(105,Pawan)
(110,Priyanka)
(114,Madhuri)
(119,)
```

e) To write a pig script to get the employees(empid, name) which is not available in expense_details.

Steps To create a pig file and execute in local mode:
----
1. Create a Emp_Details_NotInExpense.pig file using nano editor,write pig scripts as highlighted in red below and save it in local.
2. Copy the input Files, Employee_details, Employee_expense to the local filesystem.
3. execute in local mode using below command:
pig -x local Emp_Details_NotInExpense.pig

Emp_Details_NotInExpenses.pig script:

```
A = load '/home/acadgild/employee_details.txt' using PigStorage(',');
B = foreach A generate (int)$0 as EmpId, (chararray)$1 as Name, (int)$2 as salary ,(int)$3 as rating;
A1 = load '/home/acadgild/employee_expenses.txt' using PigStorage('\t');
B1 = foreach A1 generate (int)$0 as EmpId, (int)$1 as Expense;
C1 = group B1 by EmpId;
D1 = foreach C1 generate group as EmpId;
joined_table = Join B by $0 full, D1 by EmpId;
C = FILTER Joined_table by $4 is null ;
final = foreach C generate $0, $1;
dump final;
```

Script Explanation:

1.A is used to load the input file which is delimited using ','
2. B will create something like a table with data in 0th column as Empid ,data in the 1st column as EmpName and data in 2nd column as salary and data in 3rd column as rating.
3. A1 is used to load the another input file which is delimited using '\t'
4. B1 will create something like a table with data in 0th column as Empid ,data in the 1st column as Expense.
5. C1 will group B1 based on EmpId as tuple .
6. D1 works on tuple to form a group based on the empId.
7. joined_table joins both the table based on EmpId.
8. C filters the joined table such that expense tables empId is NULL.
9. final creates a group with empId and EmpName.
10. Dump will display the final output.

Output:

```
(103,Akshay)
(106,Aamir)
(107,Salman)
(108,Ranbir)
(109,Katrina)
(111,Tushar)
(112,Ajay)
(113,Jubeen)
```

Task 3:

To Implement the aviation usecase given in the blog with complete steps and screen shots.

There are 2 different datasets, i.e., Delayed_Flights.csv and Airports.csv -
Delayed_Flights.csv Datasets -

Problem Statement 1:
Find out the top 5 most visited destinations:

Highest_Visited_Destination.pig script:

```
REGISTER '/home/acadgild/piggybank.jar';
 A = load '/home/acadgild/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_
HEADER');
 B = foreach A generate (int)$1 as year, (int)$10 as flight_num, (chararray)$17 as
origin,(chararray) $18 as dest;
 C = filter B by dest is not null;
 D = group C by dest;
 E = foreach D generate group, COUNT(C.dest);
 F = order E by $1 DESC;
 Result = LIMIT F 5;
 A1 = load '/home/acadgild/airports.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_
HEADER');
 A2 = foreach A1 generate (chararray)$0 as dest, (chararray)$2 as city, (chararray)$4 as
country;
 joined_table = join Result by $0, A2 by dest;
 dump joined_table;
```

Steps to find the 5 most visited destinations:
----
1. Register the piggybank jar in order to use the CSVExcelStorage class.
2. In A, we are loading the dataset using CSVExcelStorage.
3. In B, we are generating the columns that are required for processing .
4. In C, we are filtering the null values from the “dest” column.
5. In D, we are grouping relation C by “dest.”
6. In E, we are generating the grouped column and the count of each.
7. Relation F and Result is used to order and limit the result to top 5.

We will be using another table to find the city name and country as well.

8. In A1, we are loading another table to which we will look-up and find the city as well as the country.
9. In relation A2, we are generating dest, city, and country from the previous relation.
10. In joined_table, we are joining Result and A2 based on a common column, i.e., “dest”
11. Finally, using dump, we are printing the result.

Output:

```
(ATL,106898,ATL,Atlanta,USA)
(DEN,63003,DEN,Denver,USA)
(DFW,70657,DFW,Dallas-Fort Worth,USA)
(LAX,59969,LAX,Los Angeles,USA)
(ORD,108984,ORD,Chicago,USA)
```

Problem Statement 2:

Which month has seen the most number of cancellations due to bad weather?

High_Cancellation_BadWeather.pig:

```
REGISTER '/home/acadgild/piggybank.jar';
 A = load '/home/acadgild/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_
HEADER');
 B = foreach A generate (int)$2 as month,(int)$10 as flight_num,(int)$22 as
cancelled,(chararray)$23 as cancel_code;
 C = filter B by cancelled == 1 AND cancel_code =='B';
 D = group C by month;
 E = foreach D generate group, COUNT(C.cancelled);
 F= order E by $1 DESC;
 Result = limit F 1;
 dump Result;
```

Steps to find out the most number of cancellation due to bad
----
weather:
----
1. Registering piggybank jar in order to use the CSVExcelStorage class.
2. In A, we are loading the dataset using CSVExcelStorage.
3. In B, we are generating the columns which are required for processing.
4. In C, we are filtering the data based on cancellation and cancellation code, i.e., canceled = 1 means flight have been canceled and cancel_code = 'B' means the reason for cancellation is "weather." So relation C will point to the data which consists of canceled flights due to bad weather.
4. In D, we are grouping the relation C based on every month.
5. In relation E, we are finding the count of canceled flights every month.
6. Relation F and Result is for ordering and finding the top month based on cancellation

Output:

```
2018-05-15
 : 1
(12,250)
```

Problem Statement 3:

Top ten origins with the highest AVG departure delay

Highest_AVG_Departure_Delay.pig script:

```
REGISTER '/home/acadgild/piggybank.jar';
 A = load '/home/acadgild/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_
HEADER');
 B1 = foreach A generate (int)$16 as dep_delay, (chararray)$17 as origin;
 C1 = filter B1 by (dep_delay is not null) AND (origin is not null);
 D1 = group C1 by origin;
 E1 = foreach D1 generate group, AVG(C1.dep_delay);
 Result = order E1 by $1 DESC;
 Top_ten = limit Result 10;
 Lookup = load '/home/acadgild/airports.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_
HEADER');
 Lookup1 = foreach Lookup generate (chararray)$0 as origin, (chararray)$2 as city,
(chararray)$4 as country;
 Joined = join Lookup1 by origin, Top_ten by $0;
```

```
Final = foreach Joined generate $0,$1,$2,$4;
Final_Result = ORDER Final by $3 DESC;
dump Final_Result;
```

Steps to find out top 10 AVG departure delay:

----

1. Register the piggybank jar in order to use the CSVExcelStorage class.
2. In A, we are loading the dataset using CSVExcelStorage.
3. In B, we are generating the columns that are required for processing .
4. In C1, we are removing the null values fields present if any.
5. In D1, we are grouping the data based on column “origin.”
6. In E1, we are finding average delay from each unique origin.
7. Relations named Result and Top_ten are ordering the results in descending order and printing the top ten values.

We will be following a few more steps to find some more details like country and city.

8. In Lookup, we are loading another table to which we will look up and find the city as well as the country.
9. In Lookup1, we are generating the destination, city, and country from the previous relation.
10. In Joined, we are joining relation Top_ten and Lookup1 based on common a column, i.e., “origin.”
11. In Final, we are generating required columns from the Joined table.
12. Finally, we are ordering and printing the results.

Output:

```
(CMX,Hancock,USA,116.1470588235294)
(PLN,Pellston,USA,93.76190476190476)
(SPI,Springfield,USA,83.84873949579831)
(ALO,Waterloo,USA,82.2258064516129)
(MQT,NA,USA,79.55665024630542)
(ACY,Atlantic City,USA,79.3103448275862)
(MOT,Minot,USA,78.66165413533835)
(HHH,NA,USA,76.53005464480874)
(EGE,Eagle,USA,74.12891986062718)
(BGM,Binghamton,USA,73.15533980582525)
```

Problem Statement 4:

Which route (origin & destination) has seen the maximum diversion?

Max_diverted_route.pig script:

```
REGISTER '/home/acadgild/piggybank.jar';
```

```
 A = load '/home/acadgild/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_
HEADER');
 B = FOREACH A GENERATE (chararray)$17 as origin, (chararray)$18 as dest, (int)$24 as
diversion;
 C = FILTER B BY (origin is not null) AND (dest is not null) AND (diversion == 1);
 D = GROUP C by (origin,dest);
 E = FOREACH D generate group, COUNT(C.diversion);
 F = ORDER E BY $1 DESC;
 Result = limit F 10;
 dump Result;
```

Steps to find out the maximum diverted route:

1.Registering piggybank jar in order to use CSVExcelStorage class.
2. In A, we are loading the dataset using CSVExcelStorage.
3. In B, we are generating the columns which are required for processing.
4. In C, we are filtering the data based on “not null” and diversion =1. This will remove the null records, if any, and give the data corresponding to the diversion taken.
5. In D, we are grouping the data based on origin and destination.
6. D finds the count of diversion taken per unique origin and destination.
7. Relations F and Result orders the result and produces top 10 results.

Output:

```
((ORD,LGA),39)
((DAL,HOU),35)
((DFW,LGA),33)
((ATL,LGA),32)
((ORD,SNA),31)
((SLC,SUN),31)
((MIA,LGA),31)
((BUR,JFK),29)
((HRL,HOU),28)
((BUR,DFW),25)
```