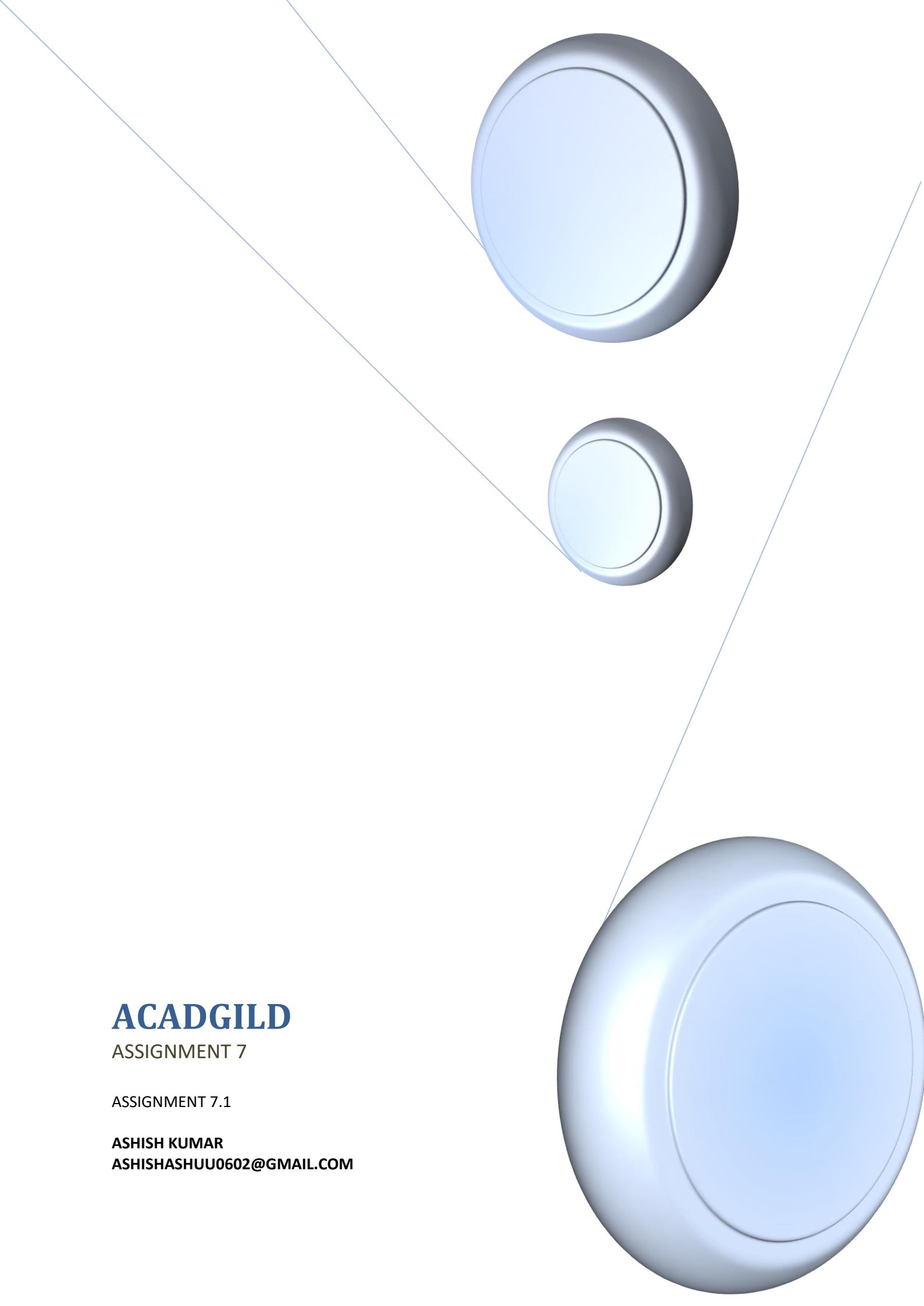


# ACADGILD

ASSIGNMENT 7

ASSIGNMENT 7.1

**ASHISH KUMAR**
**ASHISHASHUU0602@GMAIL.COM**

**Task 1: Write a program to implement wordcount as Pig.**

```
A = load '/hadoopdata/pigassignment/piginp.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;
```

used PIG Script to run the commands.

```
[acadgild@192 Assignment7]$
[acadgild@192 Assignment7]$ pig wordcount.pig
18/08/05 20:08:11 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
18/08/05 20:08:11 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
18/08/05 20:08:11 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2018-08-05 20:08:12,061 [main] INFO  org.apache.pig.Main - Apache Pig version 0.1
2018-08-05 20:08:12,061 [main] INFO  org.apache.pig.Main - Logging error messages
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/shar
```

Input used –

```
[acadgild@192 Assignment7]$ hadoop fs -cat /hadoopdata/pigassignment/piginp.txt
18/08/05 20:14:48 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha
Myself Ashish Sinha[acadgild@192 Assignment7]$
```

Output –

```
2018-08-05 20:08:38,280 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(Sinha,12)
(Ashish,12)
(Myself,12)
2018-08-05 20:08:38,418 [main] INFO  org.apache.pig.Main - Pig script completed in 26 seconds and 580 milliseconds (26580 ms)
[acadgild@192 Assignment7]$
```

**Task 2**: We have employee_details and employee_expenses files. Use local mode while running Pig and write Pig Latin script to get below results:

employee_details (EmpID,Name,Salary,EmployeeRating)
https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/employee_details.txt

employee_expenses(EmpID,Expence)
https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/employee_expenses.txt

## Step :1 Put the both the files in HDFS

```
[acadgild@192 Assignment7]$ hadoop fs -ls /hadoopdata/pigassignment/
18/08/05 20:26:14 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 4 items
-rw-r--r--   1 acadgild supergroup        273 2018-08-05 20:25 /hadoopdata/pigassignment/employee_details.txt
-rw-r--r--   1 acadgild supergroup         79 2018-08-05 20:25 /hadoopdata/pigassignment/employee_expenses.txt
-rw-r--r--   1 acadgild supergroup        250 2018-08-05 19:55 /hadoopdata/pigassignment/piginp.txt
-rw-r--r--   1 acadgild supergroup        388 2018-08-05 19:07 /hadoopdata/pigassignment/test.txt
[acadgild@192 Assignment7]$
```

(a) Top 5 employees (employee id and employee name) with highest rating. (In case two employees have same rating, employee with name coming first in dictionary should get preference)

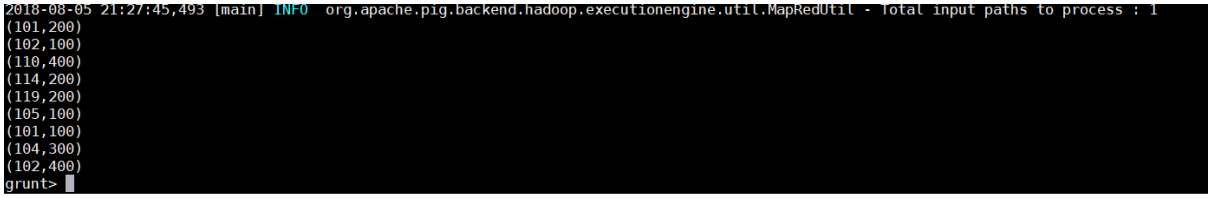
Employee_details.txt

empdet = load '/hadoopdata/pigassignment/employee_details.txt' using PigStorage(',') as (eid:int ename:chararray, esal:int, erating:int);

```
2018-08-05 20:36:56,997 [main] INFO  org.apache
(101,Amitabh,20000,1)
(102,Shahrukh,10000,2)
(103,Akshay,11000,3)
(104,Anubhav,5000,4)
(105,Pawan,2500,5)
(106,Aamir,25000,1)
(107,Salman,17500,2)
(108,Ranbir,14000,3)
(109,Katrina,1000,4)
(110,Priyanka,2000,5)
(111,Tushar,500,1)
(112,Ajay,5000,2)
(113,Jubeen,1000,1)
(114,Madhuri,2000,2)
grunt>
```

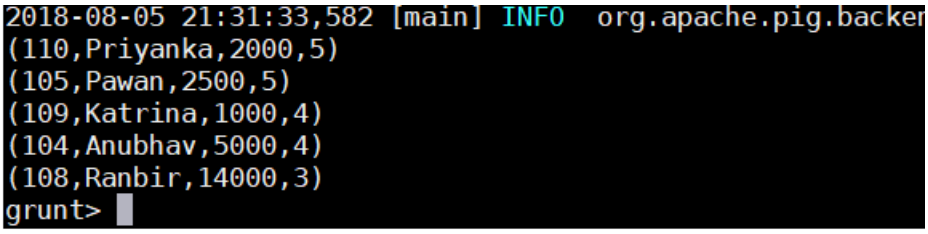
Employee_expenses.txt

empexp = load '/hadoopdata/pigassignment/employee_expenses.txt' as (eid:int, empexpence:int);




Pig Query:

rating = order empdet by erating DESC;
Result = LIMIT rating 5;
Dump Result;



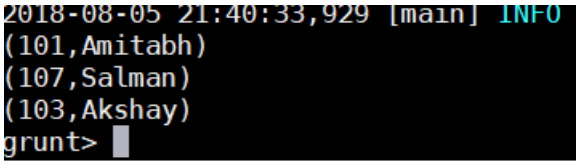

(b) Top 3 employees (employee id and employee name) with highest salary, whose employee id is an odd number. (In case two employees have same salary, employee with name coming first in dictionary should get preference)

Pig Query:

esal_name = order empdet by esal desc;
emp_sal_id = FILTER esal_name by eid%2==1;
emp_final = FOREACH emp_sal_id generate eid,ename;
emp_final_limit = LIMIT emp_final 3;

dump emp_final_limit;

Output –

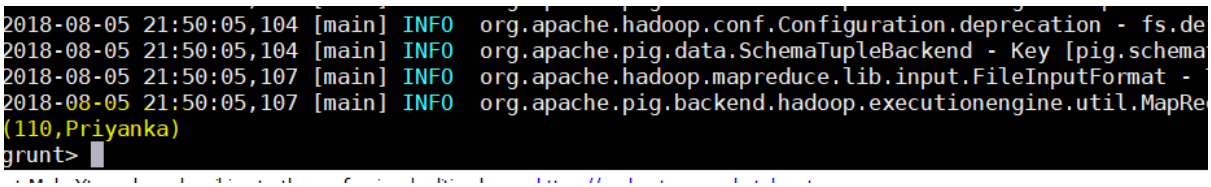



(c) Employee (employee id and employee name) with maximum expense (In case two employees have same expense, employee with name coming first in dictionary should get preference)

Pig_Query:

Joinempempexpense = join empdet by eid, empexp by eid;
maxexpense = ORDER Joinempempexpense by empexp::empexpence desc;

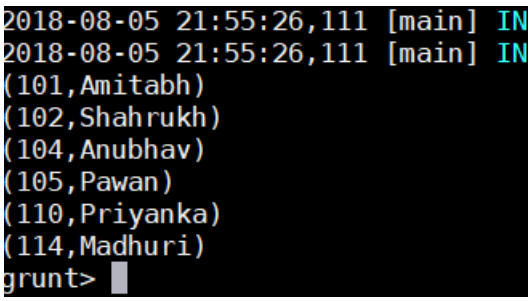
Limitmaxepnse = LIMIT maxexpense 1;
Limitmaxexpensefinal = foreach Limitmaxepnse generate
empdet::eid,empdet::ename;
dump Limitmaxexpensefinal;




(d) List of employees (employee id and employee name) having entries in employee_expenses file.
Pig Query:

emp_with_exp = JOIN empdet BY eid, empexp BY eid;
emp_with_exp_data = FOREACH emp_with_exp GENERATE empdet::eid, empdet::ename;
emp_with_exp_distinct_data = DISTINCT emp_with_exp_data;

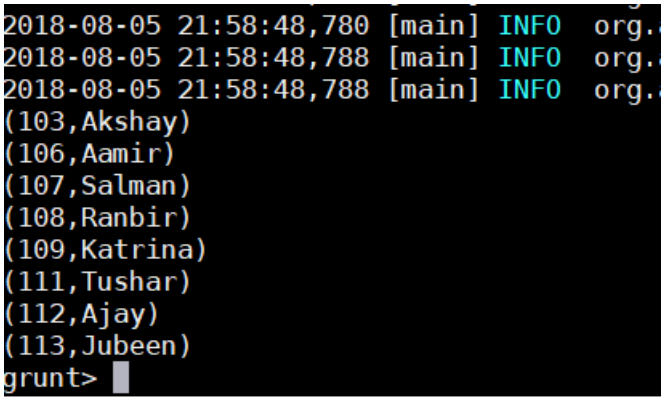
dump emp_with_exp_distinct_data;




(e) List of employees (employee id and employee name) having no entry in employee_expenses file.

Pig Query:

emp_without_exp = JOIN empdet BY eid LEFT OUTER, empexp BY eid;
emp_without_exp_filter = FILTER emp_without_exp BY empexp::eid is null;
emp_without_exp_filter_data = FOREACH emp_without_exp_filter GENERATE empdet::eid, empdet::ename;
DUMP emp_without_exp_filter_data;




**Task 3**

Implement the use case present in below blog link and share the complete steps along with screenshot(s) from your end.

https://acadgild.com/blog/aviation-data-analysis-using-apache-pig/

# Problem Statement 1

We have used local mode by using command : **pig -x local**

Find out the top 5 most visited destinations.

```
REGISTER '/home/acadgild/Desktop/Assignment7/Task3/piggybank.jar';

A = load ' /home/acadgild/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKI
P_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','SKI
P_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;
```

```
grunt> A = load '/home/acadgild/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKI
P_INPUT_HEADER');
2018-08-05 20:27:00,629 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
 dfs.bytes-per-checksum
2018-08-05 20:27:00,629 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> B = foreach A generate (int)$1 as year, (int)$10 as flight_num, (chararray)$17 as origin,(chararray) $18 as dest;
grunt> C = filter B by dest is not null;
grunt> D = group C by dest;
grunt> E = foreach D generate group, COUNT(C.dest);
grunt> F = order E by $1 DESC;
grunt> Result = LIMIT F 5;
grunt> A1 = load '/home/acadgild/airports.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INP
UT_HEADER');
2018-08-05 20:27:56,193 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
 dfs.bytes-per-checksum
2018-08-05 20:27:56,193 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> A2 = foreach A1 generate (chararray)$0 as dest, (chararray)$2 as city, (chararray)$4 as country;
grunt> joined_table = join Result by $0, A2 by dest;
grunt> dump joined_table;
2018-08-05 20:28:08,856 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN,GROUP_BY,ORDE
```

```
2018-08-05 19:17:06,643 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process :
2018-08-05 19:17:06,646 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pr
(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)
2018-08-05 19:17:07,035 [main] INFO  org.apache.pig.Main - Pig script completed in 6 minutes, 3 seconds and 743 milliseconds
```

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

Please find below all steps performed and final output :

```
grunt> REGISTER '/home/acadgild/Desktop/piggybank.jar';
2018-08-05 19:56:49,599 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
 dfs.bytes-per-checksum
2018-08-05 19:56:49,599 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> A = load '/home/acadgild/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKI
P_INPUT_HEADER');
2018-08-05 19:57:05,530 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
 dfs.bytes-per-checksum
2018-08-05 19:57:05,530 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt>
grunt> B = foreach A generate (int)$2 as month,(int)$10 as flight_num,(int)$22 as cancelled,(chararray)$23 as cancel_code;
grunt> C = filter B by cancelled == 1 AND cancel_code =='B';
grunt> D = group C by month;D = group C by month;
grunt> E = foreach D generate group, COUNT(C.cancelled);
grunt> F= order E by $1 DESC;
grunt> Result = limit F 1;
grunt> dump Result;
```

```
2018-08-05 20:00:09,574 [main] INFO  org.apache.hado
, sessionId= - already initialized
2018-08-05 20:00:09,583 [main] INFO  org.apache.hado
, sessionId= - already initialized
2018-08-05 20:00:09,599 [main] INFO  org.apache.hado
, sessionId= - already initialized
2018-08-05 20:00:09,600 [main] INFO  org.apache.hado
, sessionId= - already initialized
2018-08-05 20:00:09,609 [main] INFO  org.apache.hado
, sessionId= - already initialized
2018-08-05 20:00:09,629 [main] INFO  org.apache.pig.
2018-08-05 20:00:09,638 [main] INFO  org.apache.hado
 dfs.bytes-per-checksum
2018-08-05 20:00:09,640 [main] INFO  org.apache.hado
faultFS
2018-08-05 20:00:09,640 [main] WARN  org.apache.pig.
2018-08-05 20:00:09,661 [main] INFO  org.apache.hado
2018-08-05 20:00:09,662 [main] INFO  org.apache.pig.
(12,250)
grunt>
```

**3. Top 10 origins with the highest AVG departure delay.**

Please find below all steps performed and final output :

```
grunt> REGISTER '/home/acadgild/Desktop/piggybank.jar';
grunt> A = load '/home/acadgild/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKI
P_INPUT_HEADER');
2018-08-05 20:08:01,238 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
 dfs.bytes-per-checksum
2018-08-05 20:08:01,238 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> B1 = foreach A generate (int)$16 as dep_delay, (chararray)$17 as origin;
grunt> C1 = filter B1 by (dep_delay is not null) AND (origin is not null);
grunt> D1 = group C1 by origin;
grunt> E1 = foreach D1 generate group, AVG(C1.dep_delay);
grunt> Result = order E1 by $1 DESC;
grunt> Top_ten = limit Result 10;
grunt> Lookup = load '/home/acadgild/airports.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP
_INPUT_HEADER');
2018-08-05 20:09:22,902 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
 dfs.bytes-per-checksum
2018-08-05 20:09:22,905 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> Lookup1 = foreach Lookup generate (chararray)$0 as origin, (chararray)$2 as city, (chararray)$4 as country;
grunt> Joined = join Lookup1 by origin, Top_ten by $0;
grunt> Final = foreach Joined generate $0,$1,$2,$4;
grunt> Final_Result = ORDER Final by $3 DESC;
grunt> dump Final_Result;
```

```
2018-08-05 20:11:22,744 [main] INFO  org.apache.
2018-08-05 20:11:22,744 [main] INFO  org.apache.
(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)
```

**4. Which route (origin & destination) has seen the maximum diversion?**
**Please find below all steps performed and final output :**

```
grunt> REGISTER '/home/acadgild/Desktop/piggybank.jar';
grunt> A = load '/home/acadgild/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKI
P_INPUT_HEADER');
2018-08-05 20:15:38,523 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
 dfs.bytes-per-checksum
2018-08-05 20:15:38,523 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt>
grunt> B = FOREACH A GENERATE (chararray)$17 as origin, (chararray)$18 as dest, (int)$24 as diversion;
grunt> C = FILTER B BY (origin is not null) AND (dest is not null) AND (diversion == 1);
grunt> D = GROUP C by (origin,dest);
grunt> E = FOREACH D generate group, COUNT(C.diversion);
grunt> F = ORDER E BY $1 DESC;
grunt> Result = limit F 10;
grunt> dump Result;
```

```
2018-08-05 20:18:42,272 [main] INFO  org
((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)
grunt>
```