ASSIGNMENT 5

PART 3 - Execute 2 HBase commands from each of the 6 groups (Total 12 commands), and place the screenshots into a word file, and upload to Blackboard.
https://learnhbase.wordpress.com/2013/03/02/hbase-shell-commands/

1) General HBase shell commands
a- hbase> status

b- hbase> version

```
hbase(main):004:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 2.0000 average load
Took 0.7835 seconds
hbase(main):005:0> version
2.2.0, rUnknown, Tue Jun 11 04:30:30 UTC 2019
Took 0.0167 seconds
hbase(main):006:0> whoami
ankit (auth:SIMPLE)
    groups: ankit, adm, cdrom, sudo, dip, plugdev, lpadmin, sambashare
Took 0.0522 seconds
hbase(main):007:0>
```

2) Tables Management commands

a- create

```
hbase(main):007:0> create 'table1', {NAME=> 'f1', VERSIONS=>'5'}
Created table table1
Took 2.7647 seconds
=> Hbase::Table - table1
hbase(main):008:0>
```

b- describe

```
hbase(main):008:0> describe 'table1'
Table table1 is ENABLED
table1
COLUMN FAMILIES DESCRIPTION
{NAME => 'f1', VERSIONS => '5', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOR
EVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'false', COM
PRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536'}

1 row(s)

QUOTAS
0 row(s)
Took 0.4364 seconds
hbase(main):009:0>
```

3) Data Manipulation commands

a- get

```
hbase(main):009:0> get 'table1', 'r1'
COLUMN                                          CELL
0 row(s)
Took 0.1700 seconds
```

b- count

```
hbase(main):011:0> count 'table1', INTERVAL=>100000
0 row(s)
Took 0.1303 seconds
=> 0
hbase(main):012:0>
```

4) HBase surgery tools

a- balancer

```
hbase(main):012:0> balancer
true
Took 0.0549 seconds
=> 1
hbase(main):013:0>
```

b- compact

```
hbase(main):013:0> compact 'table1'
Took 0.1307 seconds
hbase(main):014:0>
```

5) Cluster replication tools

a- list_peers

```
hbase(main):014:0> list_peers
 PEER_ID CLUSTER_KEY ENDPOINT_CLASSNAME STATE REPLICATE_ALL NAMESPACES TABLE_CFS BANDWIDTH SERIAL
0 row(s)
Took 0.1353 seconds
=> #<Java::JavaUtil::ArrayList:0x661db63e>
```

b- enable_peer

```
hbase(main):019:0> enable_peer '1'

ERROR: Replication peer 1 does not exist
```

6) Security tools

a- grant

```
hbase(main):020:0> grant 'ankit', 'RW', 'table1'

ERROR: DISABLED: Security features are not available
```

b- user_permission

```
hbase(main):021:0> user_permission 'table1'
User                                          Namespace,Table,Family,Qualifier:Permission

ERROR: DISABLED: Security features are not available
```

PART 5 - Programming Assignment

Execute one function of your choice from each group of commands, i.e., one function from Eval Functions, one function from Load/Store functions, one function from Math functions, etc., from the Official Pig website. Every time you execute a command copy-paste the screenshot, including the output, to a word document, and submit with your assignment.

http://pig.apache.org/docs/r0.17.0/func.html

1) EVAL Functions

COUNT

```
grunt> B= GROUP ratingData1 by UserId;
grunt> X = FOREACH B GENERATE COUNT(ratingData1);
```

2) Load/Store Functions

PIGSTORAGE

```
grunt> ratingData1 = LOAD '/home/ankit/Downloads/mi-data/ratings.csv' USING PigStorage(',') AS(UserId,MovieId,Rating,TimeStamp);
2019-07-25 14:03:27,060 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> x = LIMIT ratingData1 7;
grunt> dump x;
2019-07-25 14:03:37,806 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: LIMIT
2019-07-25 14:03:37,825 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2019-07-25 14:03:37,825 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2019-07-25 14:03:37,825 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, ConstantCalculator, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NestedLimitOptimizer, PartitionFilterOptimizer, PredicatePushdownOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter]}
2019-07-25 14:03:37,838 [main] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - File Output Committer Algorithm version is 1
2019-07-25 14:03:37,895 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2019-07-25 14:03:37,896 [main] INFO  org.apache.pig.builtin.PigStorage - Using PigTextInputFormat
2019-07-25 14:03:37,897 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2019-07-25 14:03:37,908 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2019-07-25 14:03:37,997 [main] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - Saved output of task 'attempt__0001_m_000001_1' to file:/tmp/temp-474577091/tmp1809679108/_temporary/0/task__0001_m_000001
2019-07-25 14:03:38,036 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2019-07-25 14:03:38,064 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2019-07-25 14:03:38,064 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(userId,movieId,rating,timestamp)
(1,1,4.0,964982703)
(1,3,4.0,964981247)
(1,6,4.0,964982224)
```

3) Math Functions

CBRT

```
grunt> math_data = LOAD '/home/ankit/Downloads/mi-data/math.txt' USING PigStorage(',') AS(data:float);
grunt> dump math_data;
2019-07-25 14:31:53,572 [main] INFO  o
(5.0)
(16.0)
(9.0)
(2.5)
(5.9)
(3.1)
```

```
grunt> cbrt_data = foreach math_data generate (data), CBRT(data);
grunt> dump cbrt_data;
```

```
(5.0,1.709975946676697)
(16.0,2.5198420997897464)
(9.0,2.080083823051904)
(2.5,1.3572088082974532)
(5.9,1.8069688790571206)
(3.1,1.4580997208745365)
grunt>
```

4) String Functions

ENDSWITH()

```
grunt> emp_data = LOAD '/home/ankit/Downloads/mi-data/emp.txt' USING PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);
grunt> dump emp_data;
(1,Robin,22,newyork)
(2,BOB,23,Kolkata)
(3,Maya,23,Tokyo)
(4,Sara,25,London )
(5,David,23,Bhuwaneshwar )
(6,Maggy,22,Chennai)
(7,Robert,22,newyork )
(8,Syam,23,Kolkata)
(9,Mary,25,Tokyo)
(10,Saran,25,London )
(11,Stacy,25,Bhuwaneshwar )
(12,Kelly,22,Chennai)
grunt> emp_endswith_a = FOREACH emp_data GENERATE (id,name),ENDSWITH ( name, 'a' );
grunt> dump emp_endswith_a;
((1,Robin),false)
((2,BOB),false)
((3,Maya),true)
((4,Sara),true)
((5,David),false)
((6,Maggy),false)
((7,Robert),false)
((8,Syam),false)
((9,Mary),false)
((10,Saran),false)
((11,Stacy),false)
((12,Kelly),false)
grunt>
```

5) Datetime Functions

GetDay()

```
grunt> date_data = LOAD '/home/ankit/Downloads/mi-data/date.txt' USING PigStorage(',') as (id:int, date:chararray);
grunt> dump date_data;
(1,1989/09/26 09:00:00)
(2,1980/06/20 10:22:00)
(3,1990/12/19 03:11:44)
```

```
grunt> todate_data = foreach date_data generate ToDate(date,'yyyy/MM/dd HH:mm:ss')as (date_time:DateTime );
2019-07-25 14:49:28,526 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED_FUNCTION 3 time(s).
2019-07-25 14:49:28,526 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 3 time(s).
grunt> getday_data = foreach todate_data generate(date_time), GetDay(date_time);
2019-07-25 14:49:41,415 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED_FUNCTION 3 time(s).
2019-07-25 14:49:41,415 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 3 time(s).
grunt> Dump getday_data;
(1989-09-26T09:00:00.000-04:00,26)
(1980-06-20T10:22:00.000-04:00,20)
(1990-12-19T03:11:44.000-05:00,19)
```

6) Tuple,Bag,Map Functions

TOMAP()

```
grunt> tomap = FOREACH emp_data GENERATE TOMAP(name, age);
2019-07-25 15:03:46,635 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED_FUNCTION 3 time(s).
2019-07-25 15:03:46,635 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 3 time(s).
grunt> dump tomap;
([Robin#22])
([BOB#23])
([Maya#23])
([Sara#25])
([David#23])
([Maggy#22])
([Robert#22])
([Syam#23])
([Mary#25])
([Saran#25])
([Stacy#25])
([Kelly#22])
```

PART 6 - Programming Assignment - Apache Pig (Use .pig scripts)

(1 million ratings from 6000 users on 4000 movies).

http://grouplens.org/datasets/movielens/

Task 1. Write a Pig Script to find the top 25 rated movies in the movieLens dataset

```
grunt> describe joined;
joined: {ratings::userId: bytearray,ratings::temp: bytearray,ratings::movieId: bytearray,ratings::temp1: bytearray,ratings::rating: bytearray,ratings::temp2: bytearray,ratings::timestamp: bytearray,movies
::movieId: bytearray,movies::temp1: bytearray,movies::titles: bytearray,movies::temp2: bytearray,movies::genres: bytearray}
grunt> grp_movies = GROUP joined by titles;
grunt> total_rating = FOREACH grp_movies GENERATE group, COUNT(joined) AS rating;
grunt> desc_order = ORDER total_rating BY rating DESC;
grunt> top25 = LIMIT desc_order 25;
grunt> STORE top25 INTO '/movielens_output/part2';
```

OUTPUT:

```
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -ls /movielens_output/part2
Found 2 items
-rw-r--r--   1 ankit supergroup          0 2019-07-26 17:31 /movielens_output/part2/_SUCCESS
-rw-r--r--   1 ankit supergroup        665 2019-07-26 17:31 /movielens_output/part2/part-r-00000
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -cat /movielens_output/part2/part-r-00000
Star Wars       11114
Star Trek       5347
American Beauty (1999)  3428
Mission 2840
Jurassic Park (1993)    2672
Saving Private Ryan (1998)      2653
Terminator 2    2649
Austin Powers   2639
Matrix, The (1999)      2590
Back to the Future (1985)       2583
Silence of the Lambs, The (1991)        2578
Men in Black (1997)     2538
Raiders of the Lost Ark (1981)  2514
Fargo (1996)    2513
Godfather       2466
Sixth Sense, The (1999) 2459
Braveheart (1995)       2443
Shakespeare in Love (1998)      2369
Princess Bride, The (1987)      2318
Schindler's List (1993) 2304
L.A. Confidential (1997)        2288
Groundhog Day (1993)    2278
E.T. the Extra-Terrestrial (1982)       2269
Being John Malkovich (1999)     2241
Shawshank Redemption, The (1994)        2227
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$
```

Task 2. Write a Pig Script to find the number of males and females in the movielens dataset

```
grunt> users = LOAD '/movielens/users.dat' USING PigStorage(':') as (userId,temp,gender,temp1,age,temp2,occupation,temp3,zip);
grunt> grp_gender = GROUP users BY gender;
grunt> count_gender = FOREACH grp_gender GENERATE group, COUNT(users);
grunt> STORE count_gender INTO 'movielens_output/part1';
```

OUTPUT:

```
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -ls /user/ankit/movielens_output/part1
Found 2 items
-rw-r--r--   1 ankit supergroup          0 2019-07-26 17:07 /user/ankit/movielens_output/part1/_SUCCESS
-rw-r--r--   1 ankit supergroup         14 2019-07-26 17:07 /user/ankit/movielens_output/part1/part-r-00000
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -cat /user/ankit/movielens_output/part1/part-r-00000
F       1709
M       4331
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$
```

Task 3. Write a Pig Script to find the number of movies rated by different users

```
grunt> ratings = LOAD '/movielens/ratings.dat' USING PigStorage(':') as (userId,temp,movieId,temp1,rating,temp2,timestamp);
grunt> grp_rating = GROUP ratings BY userId;
grunt> count_users = FOREACH grp_rating GENERATE group, COUNT(ratings);
grunt> STORE count_users INTO '/movielens_output/part3';
```

OUTPUT:

```
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -ls /movielens_output/part3
Found 2 items
-rw-r--r--   1 ankit supergroup          0 2019-07-26 17:14 /movielens_output/part3/_SUCCESS
-rw-r--r--   1 ankit supergroup      50199 2019-07-26 17:14 /movielens_output/part3/part-r-00000
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -cat /movielens_output/part3/part-r-00000
1       53
2       129
3       51
4       21
5       198
6       71
7       31
8       139
9       106
10      401
11      137
12      23
13      108
14      25
15      201
16      35
17      211
18      305
19      255
20      24
21      22
22      297
23      304
24      136
25      85
26      400
27      70
```

PART 7 - Programming Assignment - Apache Pig (Use GRUNT Shell)

Copy the 'access.log' file, used in previous assignments, into HDFS under /logs directory.
Using the access.log file stored in HDFS, implement Pig Script to find the number of times each IP accessed the website.

PIG Script:

```
grunt> logs = LOAD '/logs' USING PigStorage(' ') as (ip);
grunt> grpip = GROUP logs BY ip;
grunt> count_ip = FOREACH grpip GENERATE group, COUNT(logs);
grunt> store count_ip into '/logs_output';
```

Output:

```
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -ls /logs_output
Found 2 items
-rw-r--r--   1 ankit supergroup          0 2019-07-26 16:41 /logs_output/_SUCCESS
-rw-r--r--   1 ankit supergroup      77367 2019-07-26 16:41 /logs_output/part-r-00000
ankit@ankit-VirtualBox:/usr/local/bin/hadoop-2.8.5/bin$ hadoop fs -cat /logs_output/part-r-00000
127.0.0.1       368
27.4.0.57       10
5.39.81.6       1
5.9.40.86       1
60.7.80.2       2
1.22.56.96      10
1.234.2.41      24
10.15.10.5      2
10.15.11.5      1
10.15.8.23      19
10.15.8.72      1
10.15.8.85      3
10.15.9.18      3
10.15.9.75      2
108.7.47.4      29
116.8.66.2      1
124.89.8.5      2
180.76.5.7      2
180.76.5.8      2
187.5.67.6      2
192.71.7.1      27
199.38.8.5      1
23.20.27.1      1
24.60.3.88      4
```

ILLUSTRATE Command:

```
--------------------------------
| logs     | ip:bytearray     |
--------------------------------
|          | 111.67.206.26    |
|          | 111.67.206.26    |
--------------------------------
-------------------------------------------------------------------------------
| grpip    | group:bytearray     | logs:bag{:tuple(ip:bytearray)}             |
-------------------------------------------------------------------------------
|          | 111.67.206.26       | {}                                         |
|          | 111.67.206.26       | {}                                         |
-------------------------------------------------------------------------------
-------------------------------------------------
| count_ip     | group:bytearray     | :long     |
-------------------------------------------------
|              | 111.67.206.26       | 2         |
-------------------------------------------------

grunt>
```