Assignment 19.1

1. Create a customer_hive table on the top of 'customer' table created in the last session.

Calculate the maximum and minimum age of customer from the table.

Step1: scan hbase table in hbase shell:

Scan 'customer'

```
hbase(main):003:0> list
TABLE
clicks
customer
2 row(s) in 0.0200 seconds

=> ["clicks", "customer"]
hbase(main):004:0> scan 'customer'
ROW                                   COLUMN+CELL
 1                                    column=details:age, timestamp=1503131028718, value=18
 1                                    column=details:location, timestamp=1503131028718, value=IND
 1                                    column=details:name, timestamp=1503131028718, value=Amit
 2                                    column=details:age, timestamp=1503131028718, value=20
 2                                    column=details:location, timestamp=1503131028718, value=PAK
 2                                    column=details:name, timestamp=1503131028718, value=Sumit
 3                                    column=details:age, timestamp=1503131028718, value=26
 3                                    column=details:location, timestamp=1503131028718, value=AUS
 3                                    column=details:name, timestamp=1503131028718, value=Rohit
 4                                    column=details:age, timestamp=1503131028718, value=24
 4                                    column=details:location, timestamp=1503131028718, value=UK
 4                                    column=details:name, timestamp=1503131028718, value=Namit
4 row(s) in 0.0530 seconds

hbase(main):005:0>
```

Step 2: create customer_hive table in hive shell.

create external table customer_hive

(

id INT,

name STRING,

location STRING,

age INT

)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

with serdeproperties ("hbase.columns.mapping"=":key, details:name, details:location, details:age")

tblproperties("hbase.table.name"="customer");

describe customer_hive;

*select * form customer_hive;*

```
hive> create external table customer_hive
    > (
    > id INT,
    > name STRING,
    > location STRING,
    > age INT
    > )
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > with serdeproperties ("hbase.columns.mapping"=":key, details:name, details:location, details:age")
    > tblproperties("hbase.table.name"="customer");
OK
Time taken: 2.109 seconds
hive> describe customer_hive
    > ;
OK
id                      int                     from deserializer
name                    string                  from deserializer
location                string                  from deserializer
age                     int                     from deserializer
Time taken: 0.335 seconds, Fetched: 4 row(s)
hive> select * from customer_hive;
OK
1       Amit    IND     18
2       Sumit   PAK     20
3       Rohit   AUS     26
4       Namit   UK      24
Time taken: 0.491 seconds, Fetched: 4 row(s)
hive>
```

Step 3: calculate the min and max age of the customer:

select MIN(age), MAX(age) from customer_hive;

```
hive> select MIN(age), MAX(age) from customer_hive;
Query ID = acadgild_20170820144040_47ae8951-8b78-4cfb-8cf8-51d09c8e93cb
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1503216555391_0003, Tracking URL = http://localhost:8088/proxy/application_1503216555391_0003/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1503216555391_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-08-20 14:40:48,834 Stage-1 map = 0%,  reduce = 0%
2017-08-20 14:40:55,222 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.74 sec
2017-08-20 14:41:02,697 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.9 sec
MapReduce Total cumulative CPU time: 2 seconds 900 msec
Ended Job = job_1503216555391_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.9 sec   HDFS Read: 254 HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 900 msec
OK
18      26
Time taken: 24.469 seconds, Fetched: 1 row(s)
```

2. Access the customer hbase table from pig and compute the maximum and minimum age among all the customers along with their corresponding name and id.

Step 1: Load HBase table in PIG relation in pig shell.

A = LOAD 'hbase://customer' USING org.apache.pig.backend.hadoop.hbase.HBaseStorage('details:', '-loadKey true') as (id:INT,details:MAP[]);*

Describe A;

```
grunt> describe A;
A: {id: int,details: map[]}
```

Step 2: adjust the data types of the columns.

B = FOREACH A GENERATE id, (CHARARRAY)details#'name' as name, (CHARARRAY)details#'location' as location, (INT)details#'age' as age;

Describe B;

```
grunt> B = FOREACH A GENERATE id, (CHARARRAY)details#'name' as name, (CHARARRAY)details#'location' as location, (INT)details#'age' as age;
grunt> Describe B;
B: {id: int,name: chararray,location: chararray,age: int}
grunt>
```

Step 3: Check data is loaded.

Dump B;

```
2017-08-20 16:25:00,539 [main] INFO  org.apach
2017-08-20 16:25:00,579 [main] INFO  org.apach
2017-08-20 16:25:00,579 [main] INFO  org.apach
(1,Amit,IND,18)
(2,Sumit,PAK,20)
(3,Rohit,AUS,26)
(4,Namit,UK,24)
grunt>
```

Step 4: Calculate the max age.

C = GROUP B all;

D = FOREACH C GENERATE MAX(B.age) as max_age;

```
grunt> C = GROUP B all;
grunt> D = FOREACH C GENERATE MAX(B.age) as max_age;
grunt>
```

Step 5: get the record details.

E = FILTER B BY age == (int)D.max_age;

Dump E;

```
2017-08-20 16:29:06,529 [main] INFO
2017-08-20 16:29:06,536 [main] INFO
2017-08-20 16:29:06,536 [main] INFO
(3,Rohit,AUS,26)
grunt>
grunt>
```

Step 6: calculate the min age.

C = GROUP B all;

D = FOREACH C GENERATE MIN(B.age) as min_age;

```
grunt> C = GROUP B all;
grunt> D = FOREACH C GENERATE MIN(B.age) as min_age;
grunt>
```

Step 7: get the record details.

E = FILTER B BY age == (int)D.min_age;

Dump E;

```
-per-checksum
2017-08-20 16:30:33,411 [main
2017-08-20 16:30:33,415 [main
2017-08-20 16:30:33,415 [main
(1,Amit,IND,18)
grunt>
```