Big Data And Hadoop

Session 19 – Assignment 1

Problem Statement:

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.

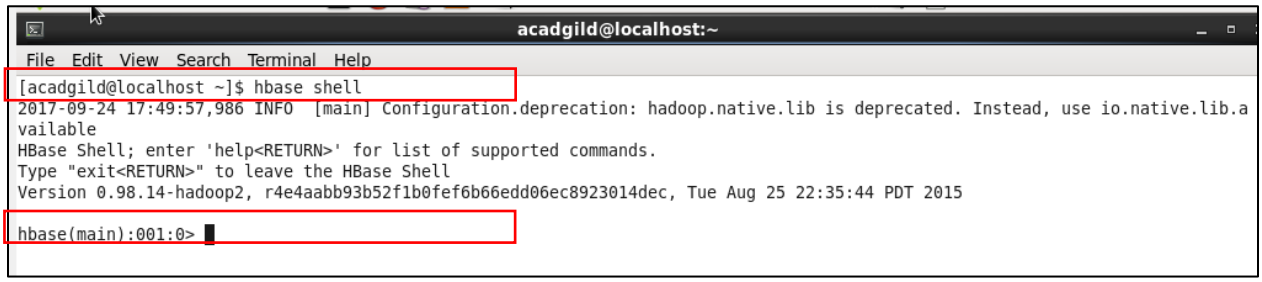
2. Access the customer hbase table from pig and compute the maximum and minimum age among all the customers along with their corresponding name and id.

Solution:

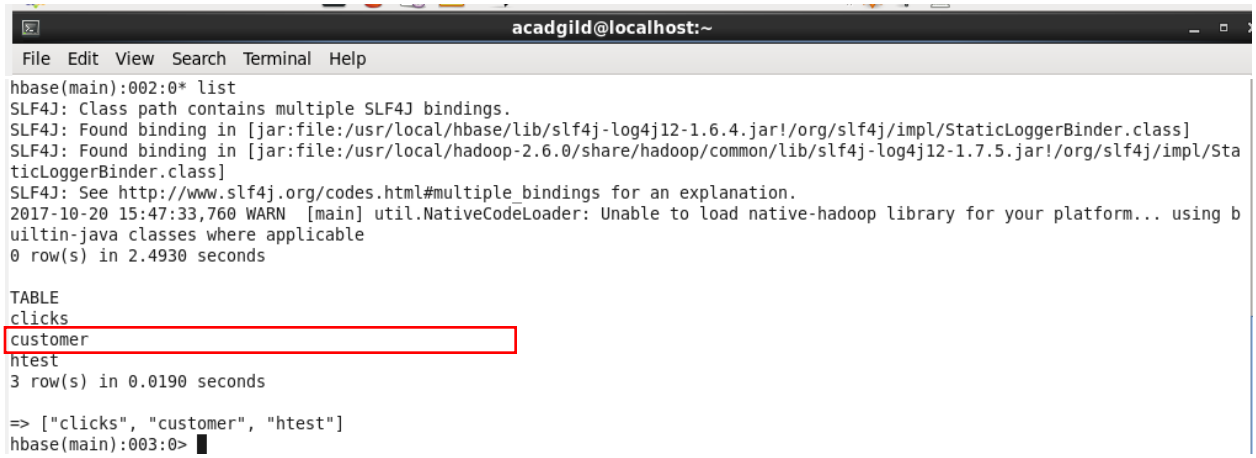
A. Start HBase and view existing table

Step 1: We first start HBase using the command **start-hbase.sh** as follows:

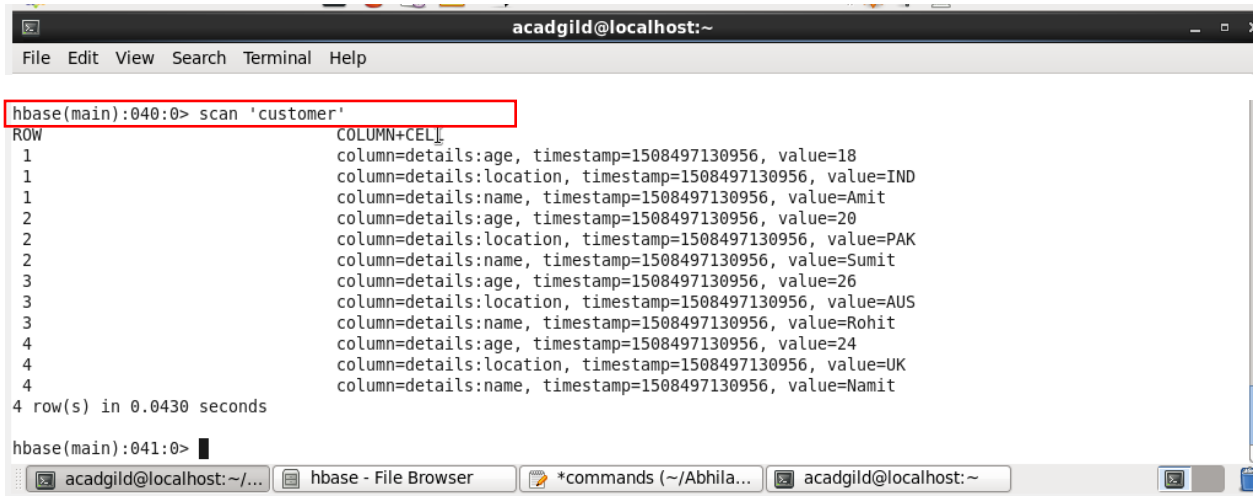
Step 2: Next we start the hbase shell using the command **hbase shell** as follows:



Step 3: Using **list** command, we can see the table **customer** appeared in the list.



Step 4: Data in this table can be viewed using the **scan** command as follows:



B. 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.

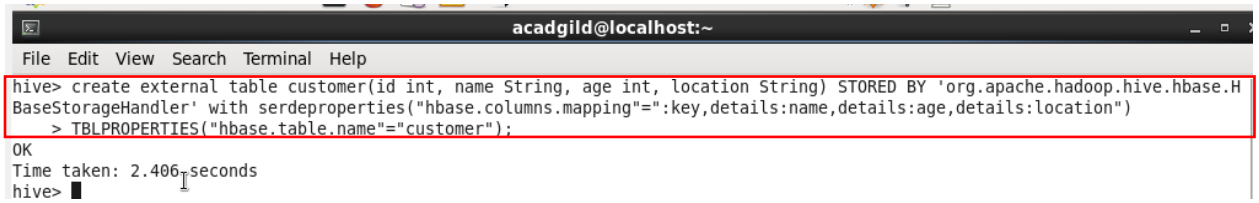
Step1: We start hive command line interface as follows:

Step 2: Create an external table in hive using the command below

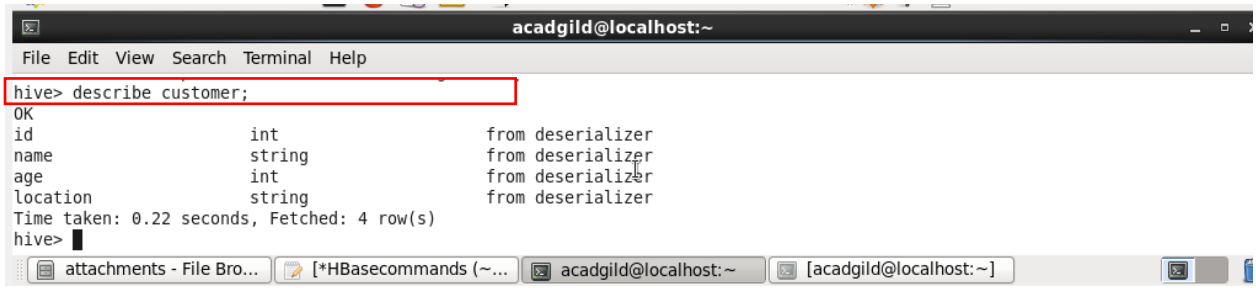
create external table
customer
(
id int,name String,age int,location String
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=":key, details:name, details:age, details:location")
tblproperties("hbase.table.name"="customer");

Here we have used HiveStorageHandler. HiveStorageHandler is the primary interface Hive uses to connect with NoSQL databases

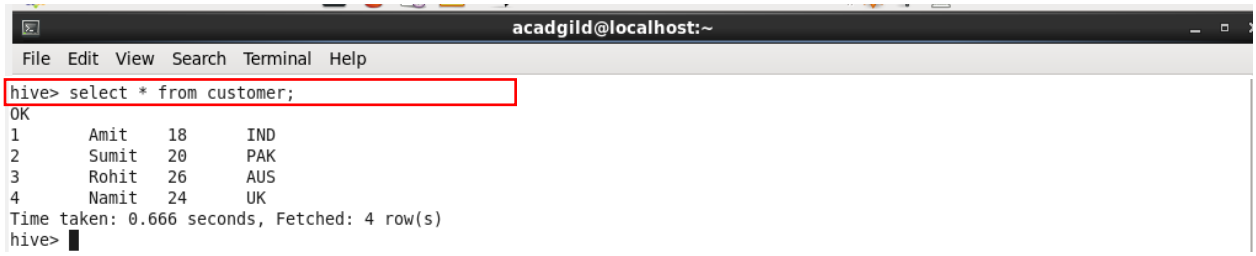
such as HBase, Cassandra, and others. If we use HiveStorageHandler we can create hive table which will point to Hbase table and we can query the data in sql style. For every single column family, one Hive table has to be created.



Step 3: After execution the above command, table in hive is created. Its schema can be seen using the **describe** command as follows:



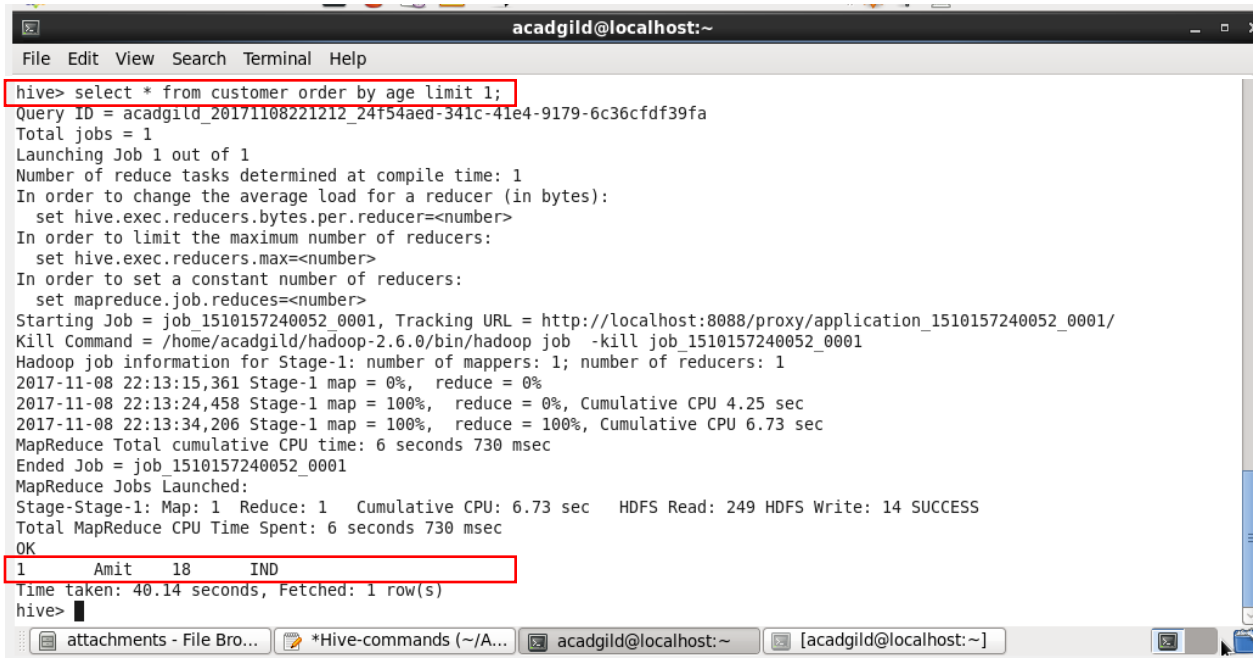
Step 4: Data that got inserted can be seen using **select** query as follows:



Step 5: Now to find the customer with minimum age, we use the following query:

*Select * from customer order by age limit 1;*

Here, we order the records by age using **ORDER BY**. Then we fetch only one record, hence used **LIMIT** clause.

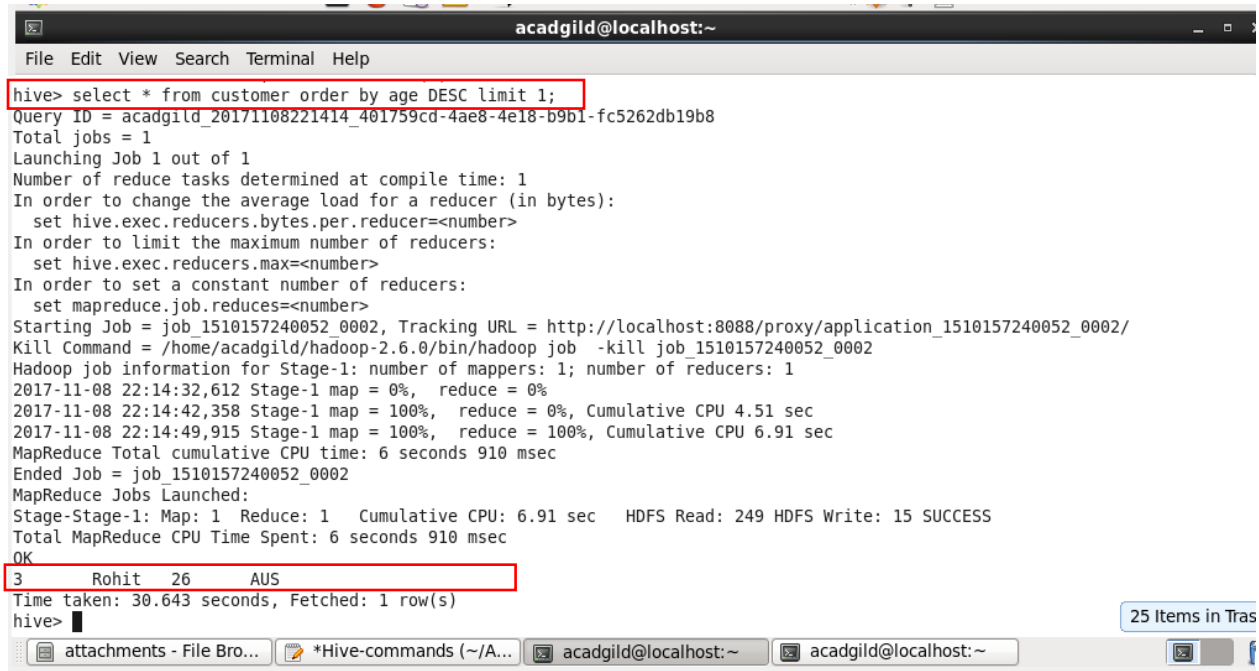


We get the output as shown in the above screenshot.

Step 6: Now to find the customer with maximum age, we use the following query:

*Select * from customer order by age DESC limit 1;*

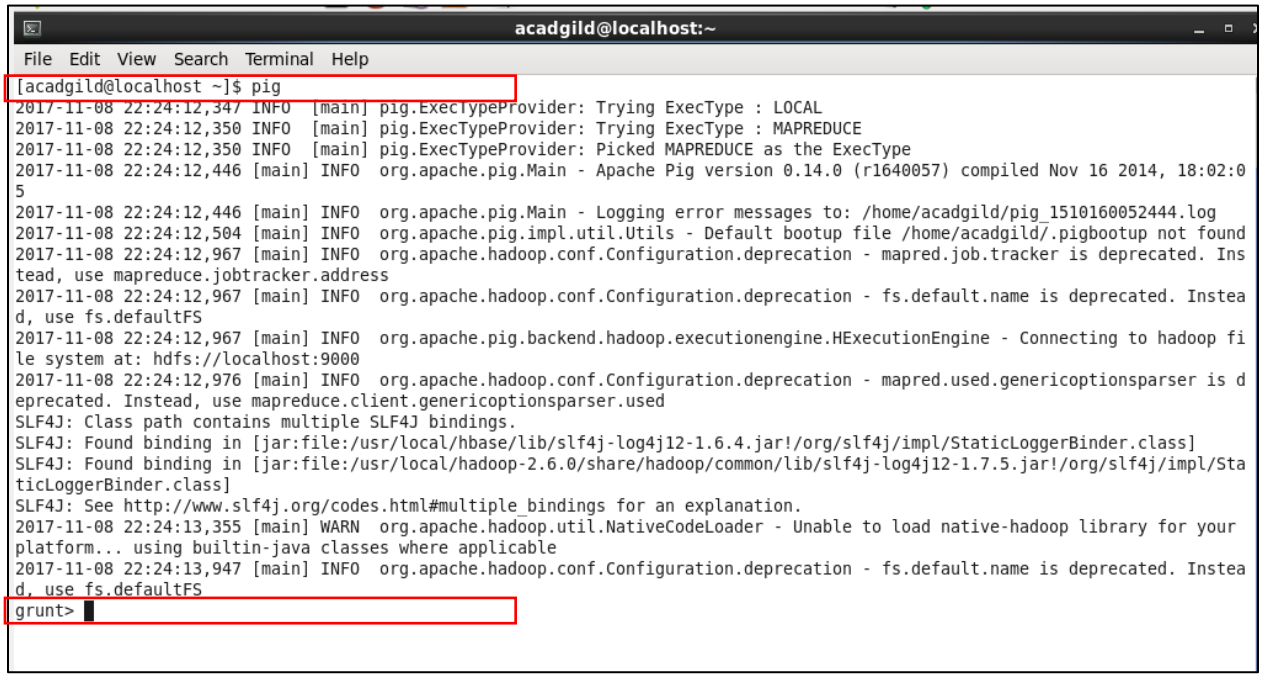
Here, we order the records by age using **ORDER BY**. Also, to get records in descending order, we use the key word **DESC.** Then we fetch only one record, hence used **LIMIT** clause.



We get the output as shown in the above screenshot.

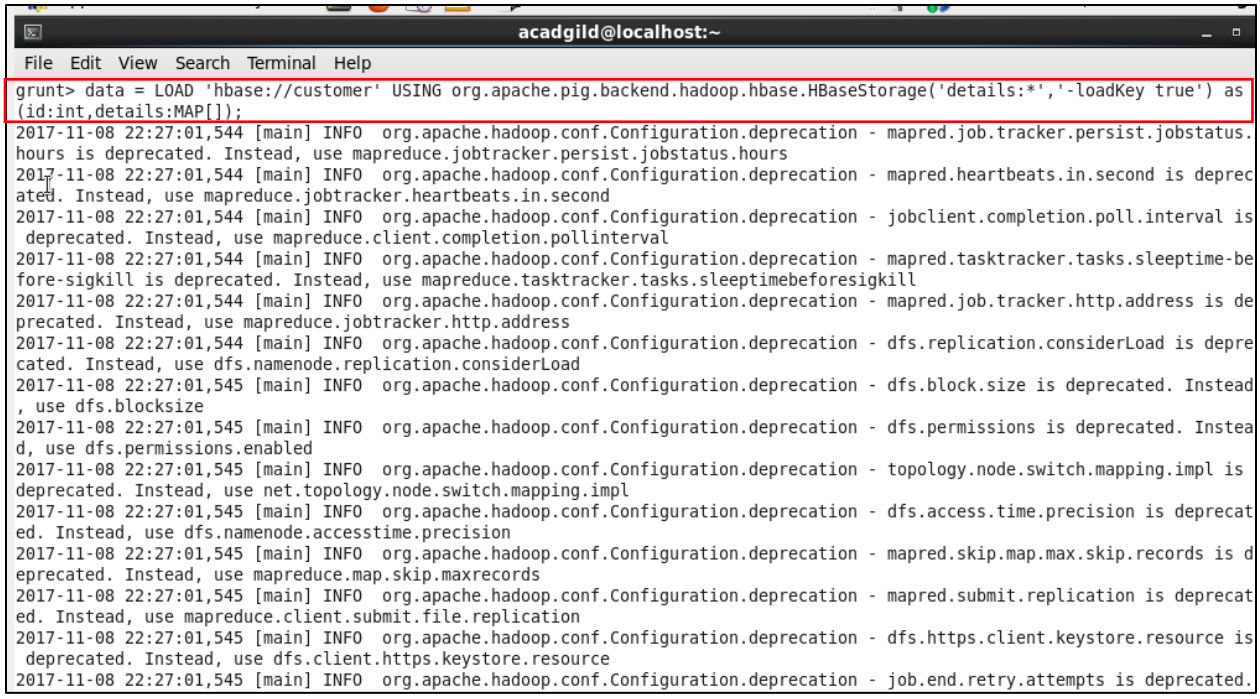
C. Access the customer hbase table from pig and compute the maximum and minimum age among all the customers along with their corresponding name and id.

Step 1: We first start pig command line interface as follows:

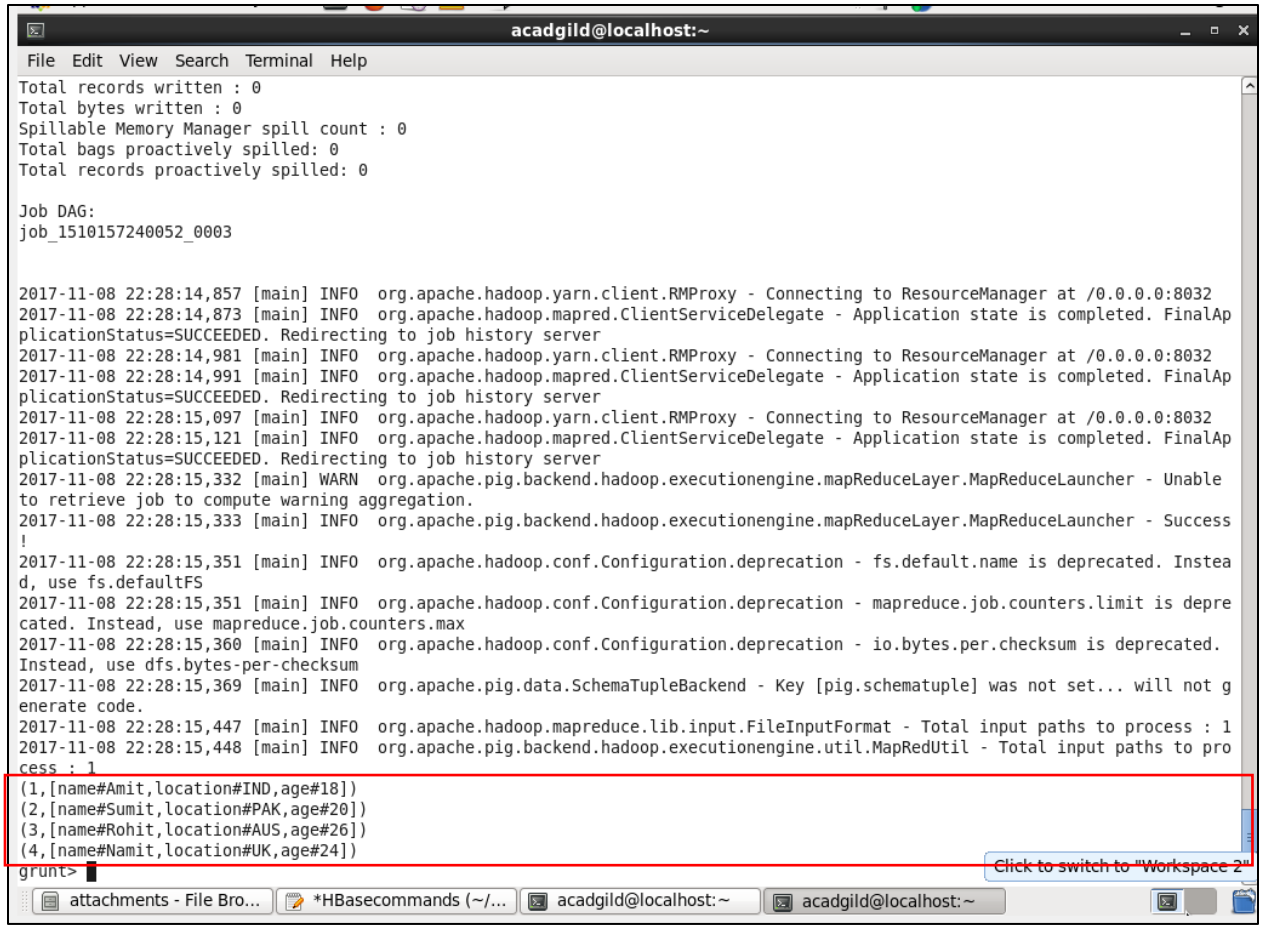


Step 2: Next, we load data from HBase into pig using the command below:

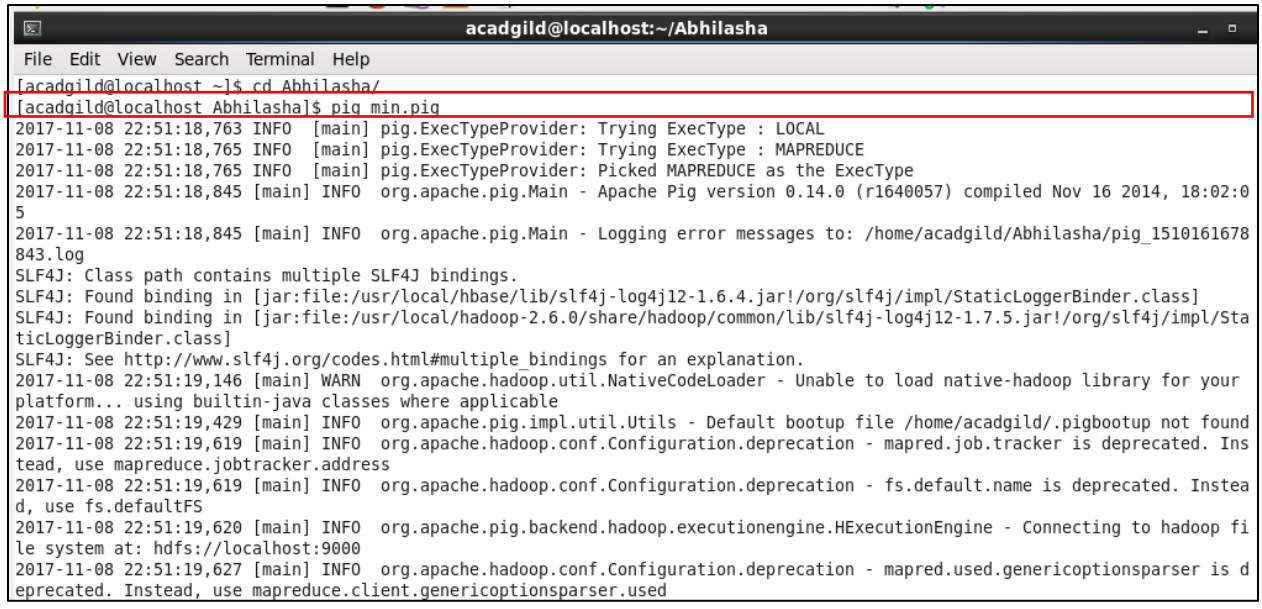
data = LOAD 'hbase://customer' USING org.apache.pig.backend.hadoop.hbase.HBaseStorage('details: ', '-loadKey true') as (id:INT, details:MAP[]);*



Step 3: After the execution of the above command, data got loaded into pig. This can be seen using **scan** command. On executing the command **dump**, we get the following output:



Step 4: Now to find the record with minimum age, we have a pig script that can be run as follows:



The steps used are:

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

   This is to load data from HBase to pig as mentioned in the previous step.

b. *record = FOREACH data GENERATE id,details#'name' AS name,details#'age' AS age;*

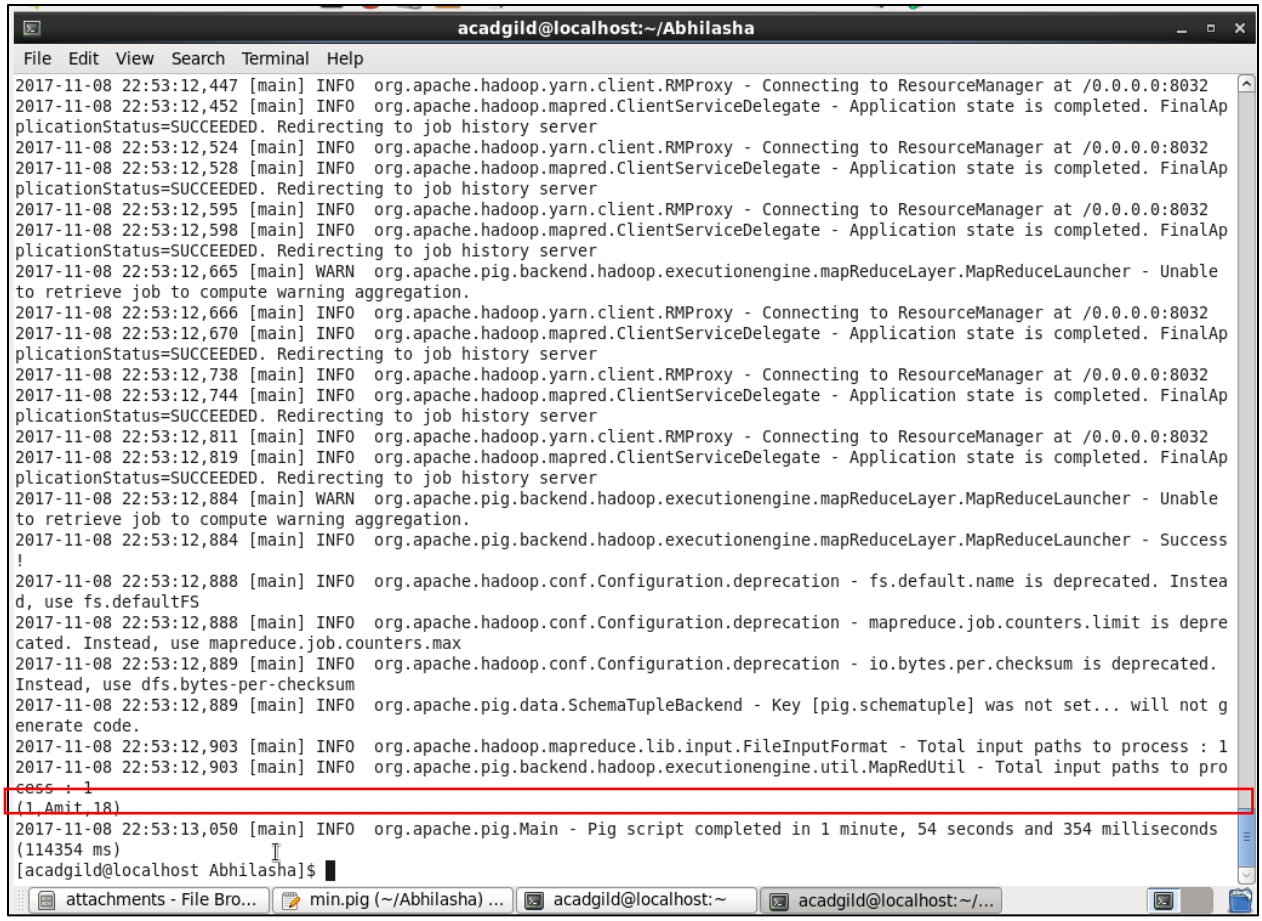
   Here, we have created alias for the columns we need, for our ease.

c. *orderedData = ORDER record BY age;*
   Here, we arrange the records in ascending order of the age.

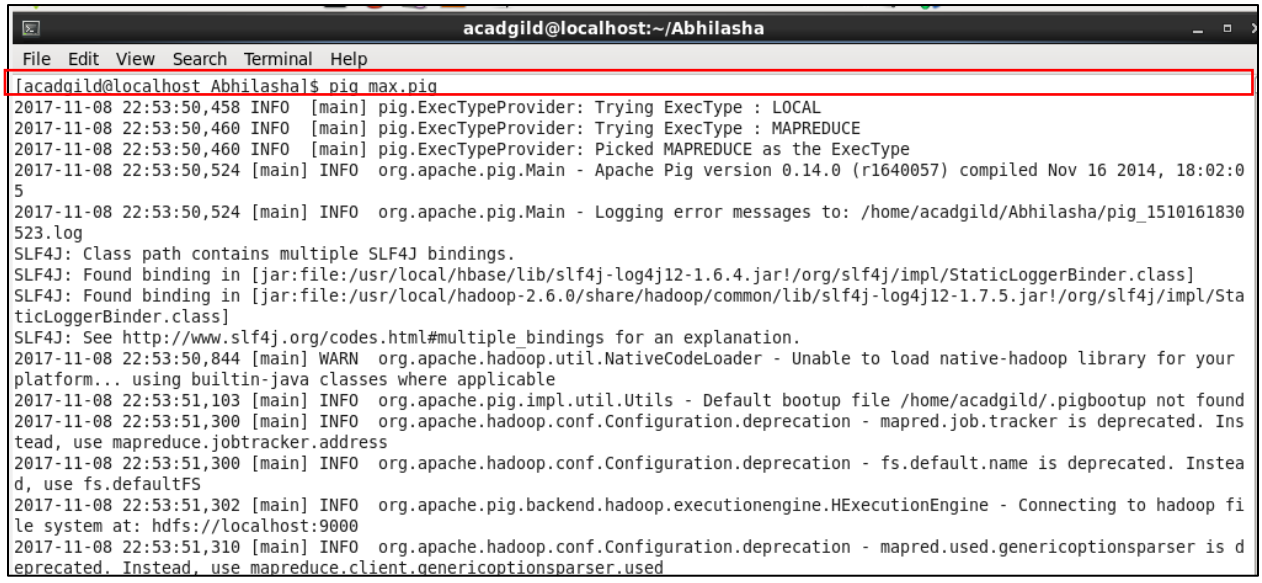
d. *limitedData = LIMIT orderedData 1;*
   Here, we limit the data to only 1 record.

e. *dump limitedData;*
   This is to dump data on console.

The output of this script execution is as follows:



Step 5: Now to find the record with maximum age, we have a pig script that can be run as follows:



The steps used are:

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

   This is to load data from HBase to pig as mentioned in the previous step.

b. *record = FOREACH data GENERATE id,details#'name' AS name,details#'age' AS age;*

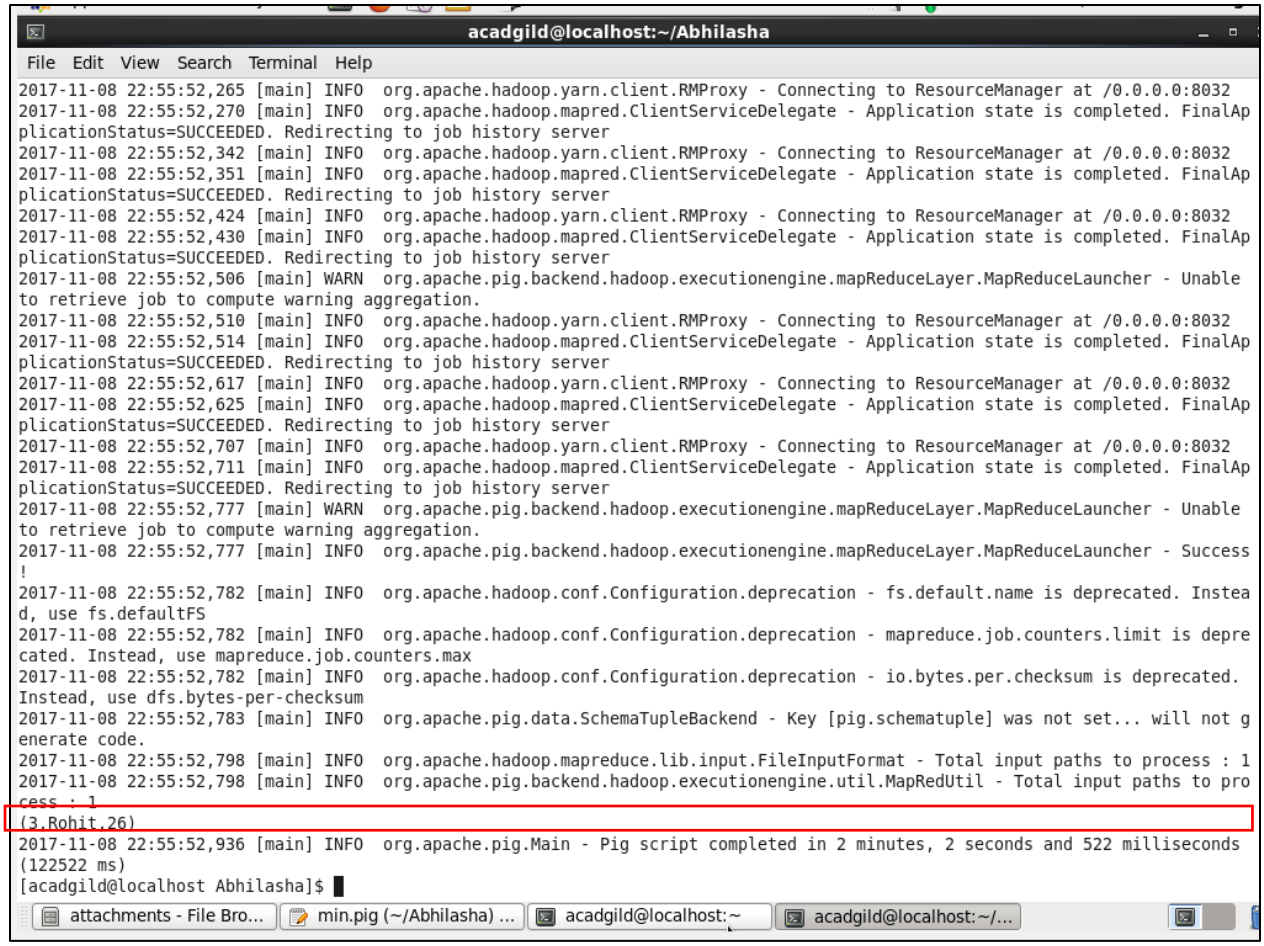
   Here, we have created alias for the columns we need, for our ease.

c. *orderedData = ORDER record BY age DESC;*
   Here, we arrange the records in descending order of the age.

d. *limitedData = LIMIT orderedData 1;*
   Here, we limit the data to only 1 record.

e. *dump limitedData;*
   This is to dump data on console.

The output of this script execution is as follows:


acadgild@localhost:~/Abhilasha
File Edit View Search Terminal Help
2017-11-08 22:55:52,265 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2017-11-08 22:55:52,270 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-11-08 22:55:52,342 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2017-11-08 22:55:52,351 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-11-08 22:55:52,424 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2017-11-08 22:55:52,430 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-11-08 22:55:52,506 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Unable to retrieve job to compute warning aggregation.
2017-11-08 22:55:52,510 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2017-11-08 22:55:52,514 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-11-08 22:55:52,617 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2017-11-08 22:55:52,625 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-11-08 22:55:52,707 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2017-11-08 22:55:52,711 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-11-08 22:55:52,777 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Unable to retrieve job to compute warning aggregation.
2017-11-08 22:55:52,777 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-11-08 22:55:52,782 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-11-08 22:55:52,782 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-11-08 22:55:52,782 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-11-08 22:55:52,783 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2017-11-08 22:55:52,798 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-11-08 22:55:52,798 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(3,Rohit,26)
2017-11-08 22:55:52,936 [main] INFO org.apache.pig.Main - Pig script completed in 2 minutes, 2 seconds and 522 milliseconds (122522 ms)
[acadgild@localhost Abhilasha]$
attachments - File Bro...
min.pig (~/Abhilasha) ...
acadgild@localhost:~
acadgild@localhost:~/...