# Hadoop distributed file system (HDFS)

On commodity hardware, the distributed file system HDFS manages enormous data sets. A single Apache Hadoop cluster can be scaled up to hundreds or even thousands of nodes using HDFS. One of Apache Hadoop's key components, along with Map Reduce and YARN, is HDFS.

The dataset on which the analysis is being done in this assignment are:

1) MedianHouseholdIncome2015.csv

2) PercentagePeopleBelowPovertyLevel.csv

3) PercentOver25CompletedHighSchool.csv.

4) PoliceKillingsUS.csv

5) ShareRaceByCity.csv

These datasets are first uploaded in Local storage. Then these files are copied from Local storage to HDFS.

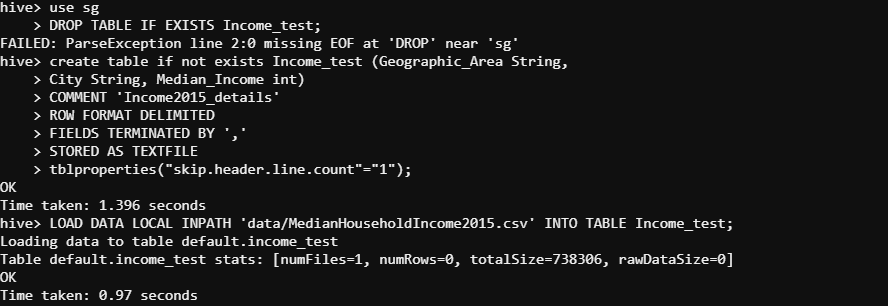
Other commands to perform different other operation on HDFS which were used here are:

# Implementation using Hive, MAPREDUCE, MySQL and Sqoop interface in CloudxLab

## Loading file MedianHouseholdIncome2015.csv file on Hive Interface:

Here, we are using sg Database on Hive and creating table income test.

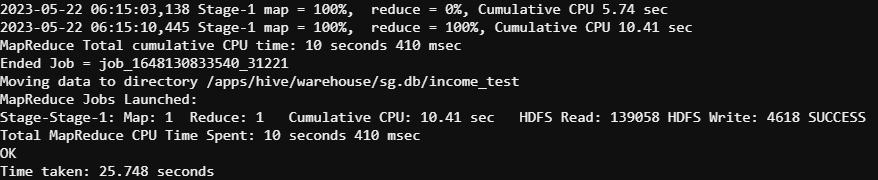
Then we are loading MedianHouseholdIncome2015.csv file into income test table.



Here, we are loading top ten entries from income test table:



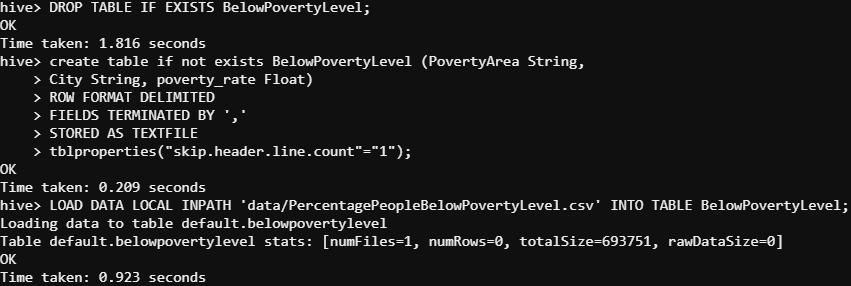
Here, we are overwriting directory '/apps/hive/warehouse/sg.db/income test' with the contents from income test table with the help of MAP REDUCE program running in the background



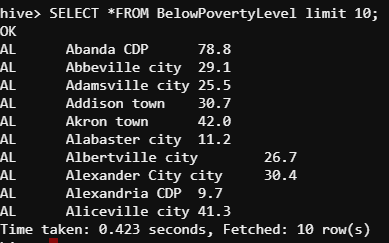
## Loading file PercentagePeopleBelowPovertyLevel.csv file on Hive Interface:

Here, we are using sg Database on Hive and creating table BelowPovertyLevel.

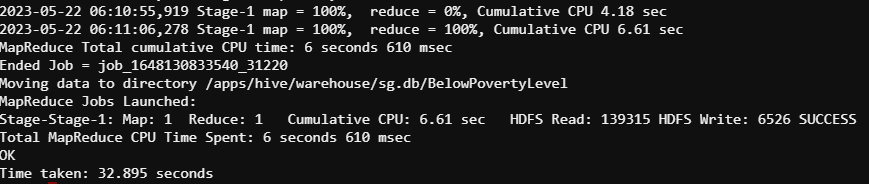
Then we are loading PercentagePeopleBelowPovertyLevel.csv file into BelowPovertyLevel table.



Here, we are loading top ten entries from BelowPovertyLevel table:



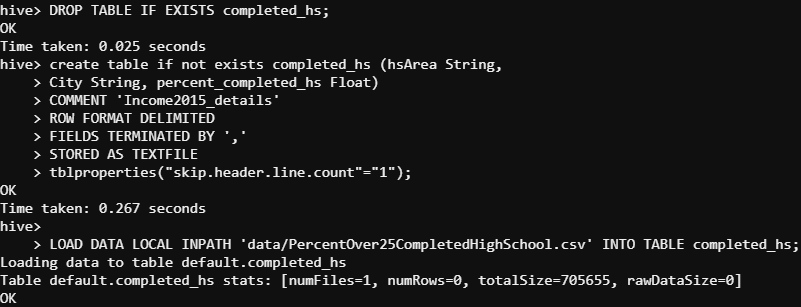
Here, we are overwriting directory ' /apps/hive/warehouse/sg.db/BelowPovertyLevel’ with the contents from BelowPovertyLevel table with the help of MAP REDUCE program running in the background.



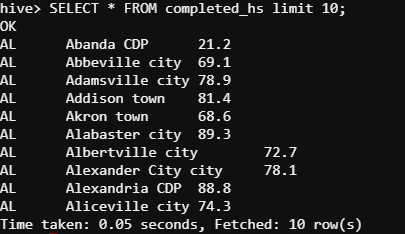
## Loading file PercentOver25CompletedHighSchool.csv file on Hive Interface:

Here, we are using sg Database on Hive and creating table completed\_hs.

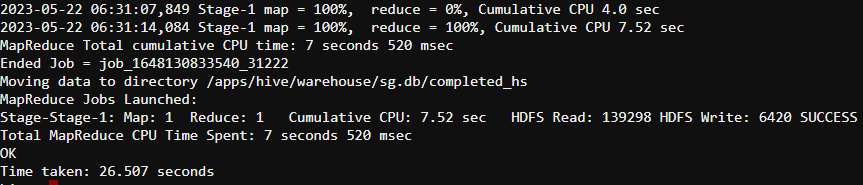
Then we are loading PercentOver25CompletedHighSchool.csv file into completed\_hs table.



Here, we are loading top ten entries from completed\_hs table:

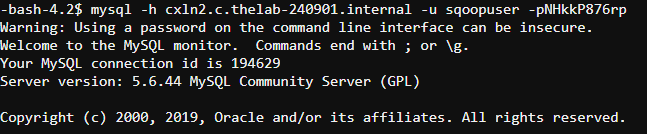


Here, we are overwriting directory ' /apps/hive/warehouse/sg.db/completed\_hs ‘with the contents from completed\_hs with the help of MAP REDUCE program running in the background

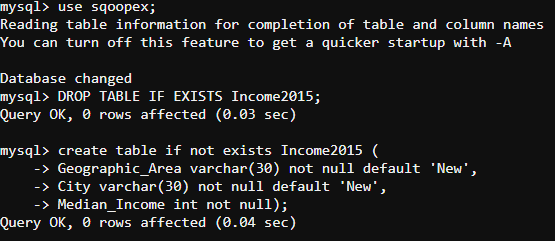


## Loading file MedianHouseholdIncome2015.csv file on MYSQL Interface:

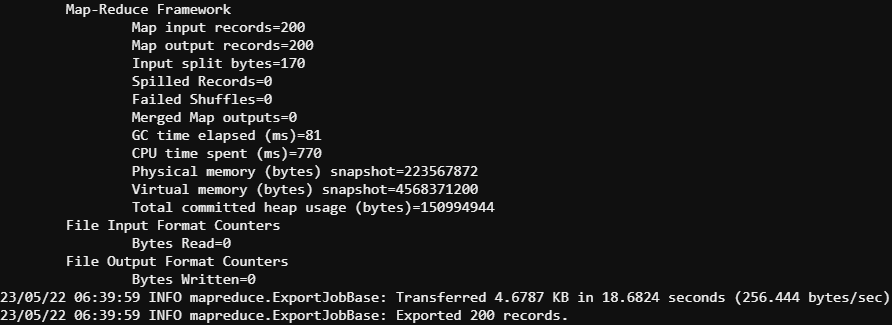
Here, we are launching MySQL in CloudxLab



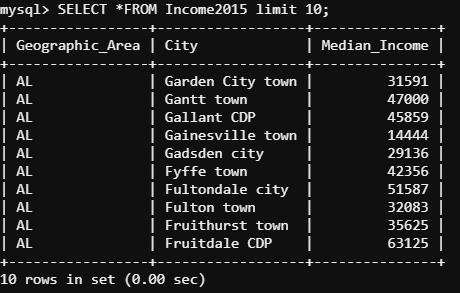
Here we are using sqoopex database and creating Income2015 table in MySQL



Here we are performing Sqoop Export - Hive to MySQL with the help of MAP REDUCE program running in the background:

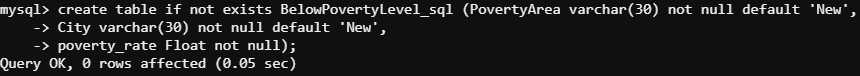


Here, we are loading top ten entries from Income2015table in MySQL interface in CloudxLab:

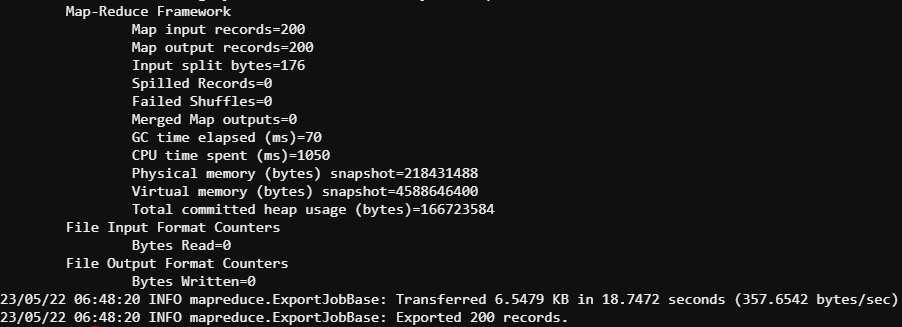


## Loading file PercentagePeopleBelowPovertyLevel.csv file on MYSQL Interface:

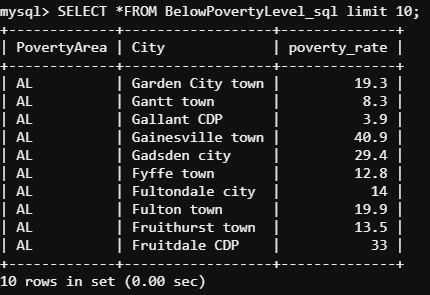
Here we are using sqoopex database and creating BelowPovertyLevel\_sql table in MySQL



Here we are performing Sqoop Export - Hive to MySQL with the help of MAP REDUCE program running in the background:

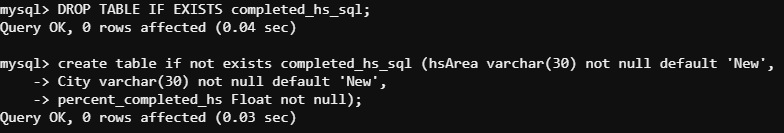


Here, we are loading top ten entries from BelowPovertyLevel\_sql in MySQL interface in CloudxLab:

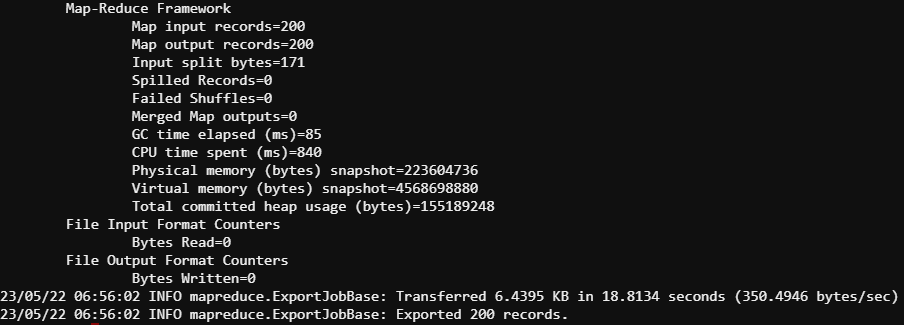


## Loading file PercentOver25CompletedHighSchool.csv file on MYSQL Interface:

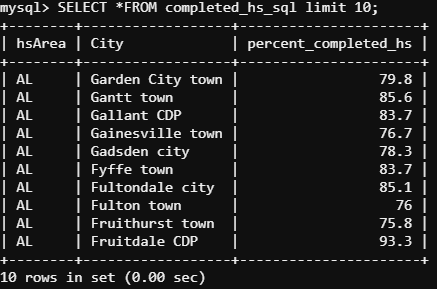
Here we are using sqoopex database and creating completed\_hs\_sql table in MySQL



Here we are performing Sqoop Export - Hive to MySQL with the help of MAP REDUCE program running in the background:



Here, we are loading top ten entries from completed\_hs\_sql in MySQL interface in CloudxLab:



## The insights drawn out of the analysis on MySQL interface in CloudxLab:

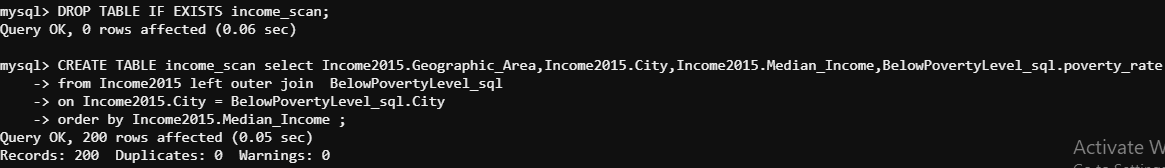
### The dataset on which the analysis is being done:

1) MedianHouseholdIncome2015.csv

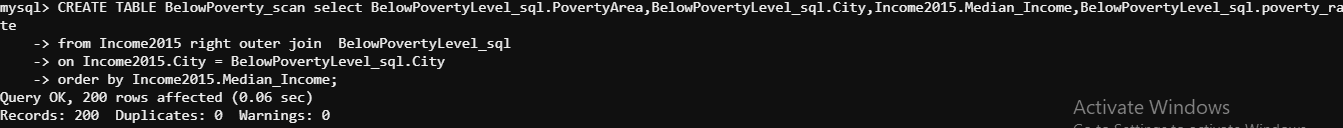
2) PercentagePeopleBelowPovertyLevel.csv

3) PercentOver25CompletedHighSchool.csv.

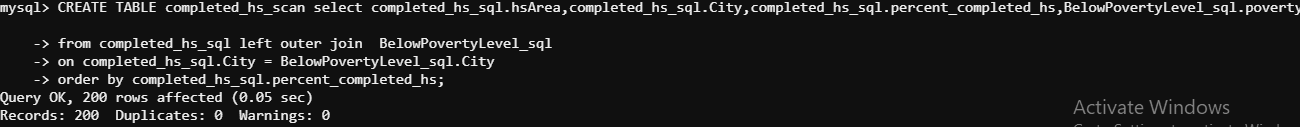
Here we are using sqoopex database and creating income scan table by left outer join of Income2015 and BelowPovertyLevel\_sql table in MySQL



Here we are using sqoopex database and creating BelowPoverty\_scan table by right outer join of Income2015 and BelowPovertyLevel\_sql table in MySQL

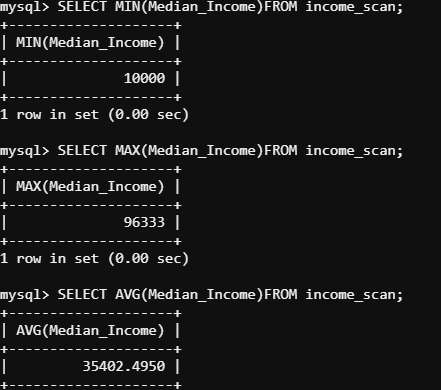


Here we are using sqoopex database and creating completed\_hs\_scan table by left outer join of completed\_hs\_sql and BelowPovertyLevel\_sql table in MySQL

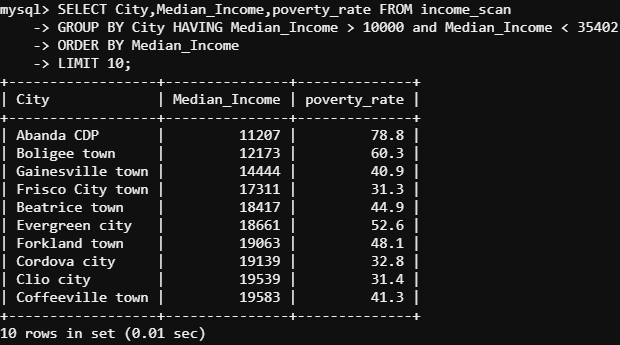


### Analyzing income scan table:

Here, we are querying income scan table to compare and analyze poverty rate and Median Income for people living in cities as shown below:

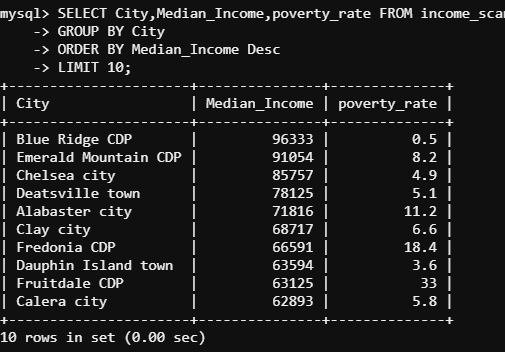


Here, we are querying to extract increasing ten median incomes listings from income scan Table to compare and analyze poverty rate and Median Income for people living in cities as shown below:

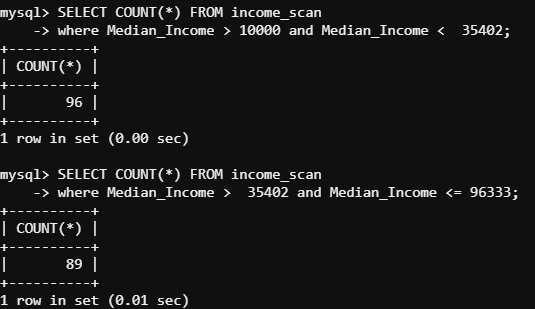
******

***Insights drawn:*** We can see that the query returned ten records with growing Median Income, demonstrating that poverty rates frequently decrease as median income rises. Another conclusion is that based on the general income ranges of the cities, the poverty rate estimate vary for each city. Additionally, the bottom 10 cities with the least median income are Abanda CDP, Boligee Town, Gainesville Town, Frisco City Town, Beatrice Town, Evergreen Town, Forkland Town, Cordova Town, Clio Town, and Coffeeville Town.

Here, we are querying to extract descending ten median incomes listings from income scan Table to compare and analyze poverty rate and Median Income for people living in cities as shown below:

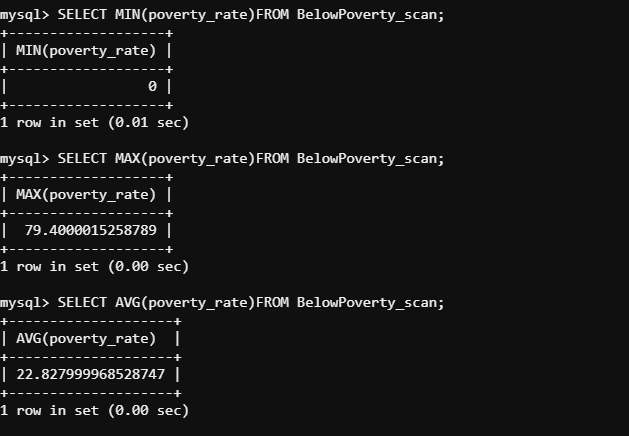


***Insights drawn:*** As we can see, the query returned the top 10 results for Median Income in descending order, showing that cities with higher median incomes had lower poverty rates. Usually, when the median income falls, the poverty rate rises. Another conclusion is that the calculation of the poverty rate differs for each city depending on the general income ranges of the cities. Apart from that, the top ten cities with declining median income are Blue Ridge CDP, Emerald Mountain CDP, Chelsea city, Deatsville town, Alabaster city, Clay city, Fredonia CDP, Duphin Island town, Fruitdale CDP, and Calera city.

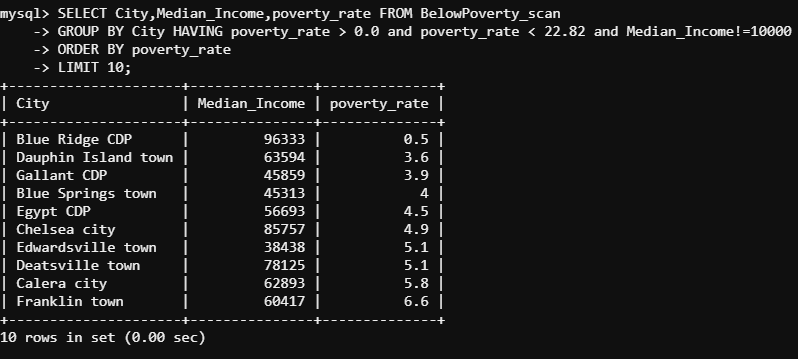


### Analyzing BelowPoverty\_scan table:

Here, we are querying BelowPoverty\_scan table to compare and analyze poverty rate and Median Income for people living in cities as shown below:

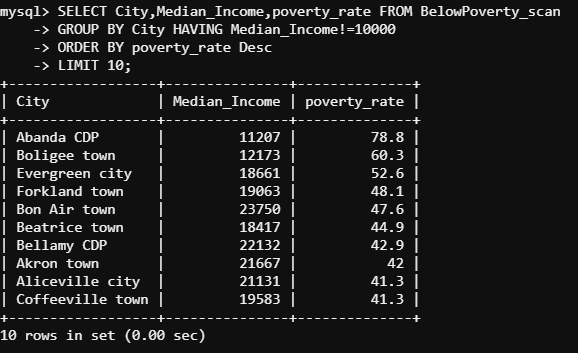


Here, we are querying to extract increasing ten poverty rate listings from BelowPoverty\_scan Table to compare and analyze poverty rate and Median Income for people living in cities as shown below:

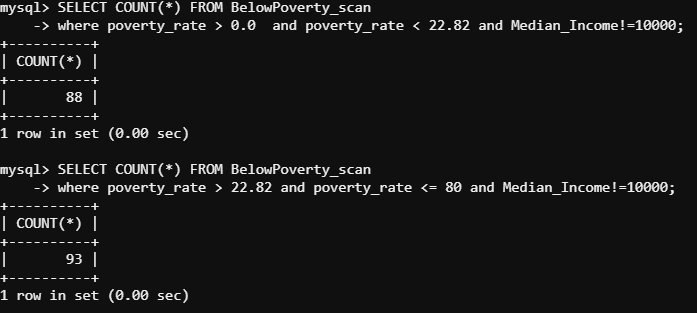


***Insights drawn:*** As we can see, the search produced 10 records with entries for the poverty rate, which should have been rising. However, it is evident from the statistics that based on the overall income ranges of the cities, the calculation of poverty rates varies for each city. Also, the top 10 cities with the lowest rates of poverty are Blue Ridge CDP, Dauphin Island town, Gallant CDP, Blue Springs town, Egypt CDP, Chelsea city, Edwardsville town, Deatsville town, Calera city, and Franklin town.

Here, we are querying to extract descending ten poverty\_rate listings from BelowPoverty\_scan Table to compare and analyze poverty rate and Median Income for people living in cities as shown below:

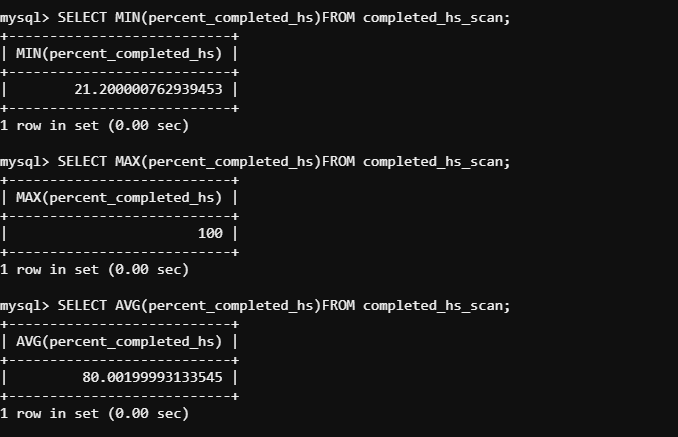


***Insights drawn:*** Typically, the poverty rate increases as the median income declines. However, based on the general income ranges of the cities, the calculation of the poverty rate varies for each one. In addition, the following places are home to the bottom 10 cities with the highest rates of poverty: Abanda CDP, Boligee town, Evergreen city, Forkland town, Bon Air town, Beatrice town, Bellamy CDP, Akron town, Aliceville city, and Coffeeville town

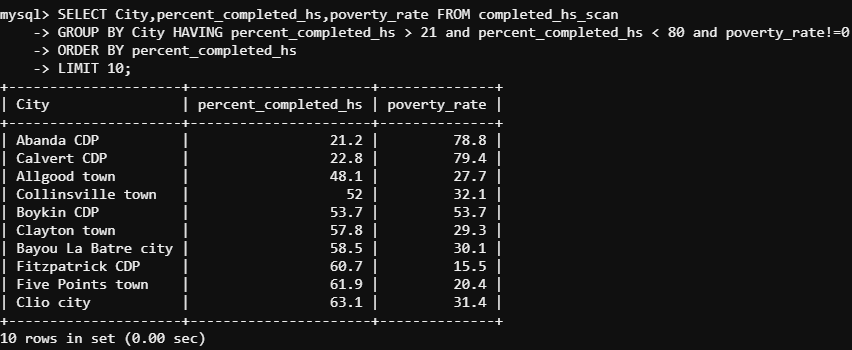


### Analyzing completed\_hs\_scan table:

Here, we are querying completed\_hs\_scan table to compare and analyze poverty rate and Median Income for people living in cities as shown below:

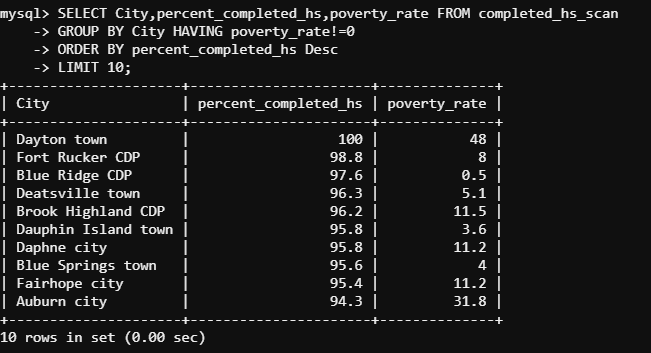


Here, we are querying to extract increasing ten percent\_completed\_hs listings from completed\_hs\_scan Table to compare and analyze poverty rate and percent\_completed\_hs for people living in cities as shown below:

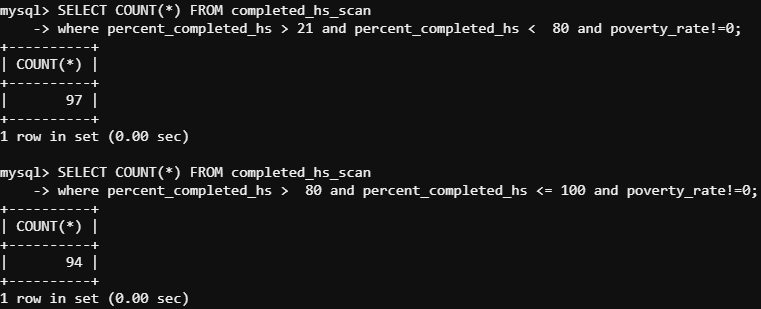


***Insights drawn:*** Based on the findings stated above, we can say that cities with even the lowest rates of poverty have the lowest proportion of residents who have finished their higher secondary education, however occasionally the results are the exact opposite. The bottom 10 cities, which are shown below, are those with the lowest percentages of citizens who have completed a higher education: Abanda CDP, Calvert CDP, Allgood town, Collinsville town, Boykin CDP, Clayton town, Bayou La Batre city, Fitzpatrick CDP, Five Points town, and Clio city.

Here, we are querying to extract descending ten percent\_completed\_hs listings from completed\_hs\_scan Table to compare and analyze poverty rate and percent\_completed\_hs for people living in cities as shown below:

******

***Insights drawn:*** Cities with larger percentages of residents who have completed their higher secondary education typically don't appear to be associated to the poverty rate graph. Additionally, the following are home to the bottom 10 cities with the highest percentage of residents finishing higher secondary education: Dayton town, Fort Rucker CDP, Blue Ridge CDP, Deatsville town, Brook Highland CDP, Dauphin Island town, Daphne city, Blue Springs town, Fairhope city, Auburn city.

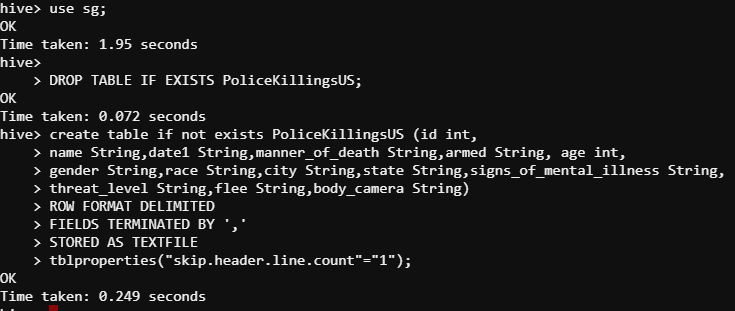


# Implementation using Hive and MAPREDUCE interface in CloudxLab

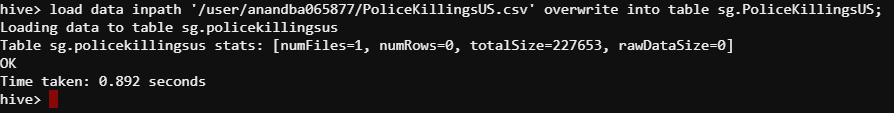
## Loading file PoliceKillingsUS.csv file on Hive Interface:

Here, we are using sg Database on Hive and creating table PoliceKillingsUS.

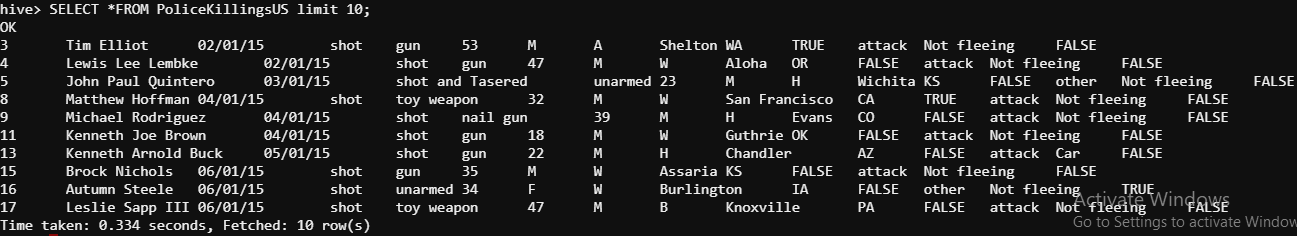
Then we are loading PoliceKillingsUS.csv file into PoliceKillingsUS table.



Then we are loading PoliceKillingsUS.csv from HDFS into our PoliceKillingsUS Table in SG database on Hive interface in CloudxLab.

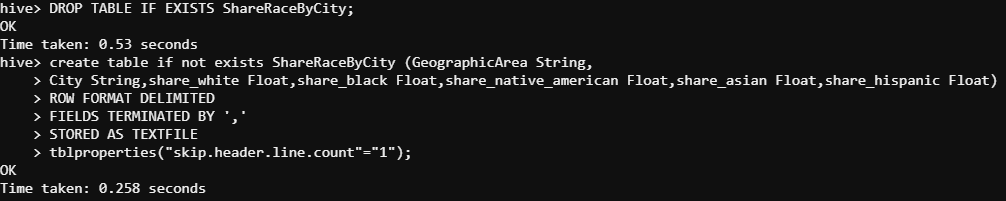


Here, we are loading top ten entries from PoliceKillingsUS table:

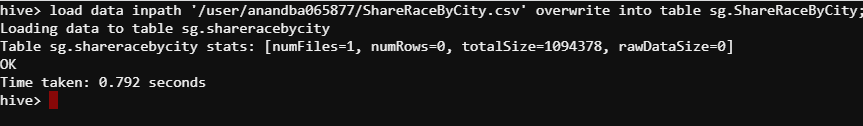


## Loading file ShareRaceByCity.csv file on Hive Interface:

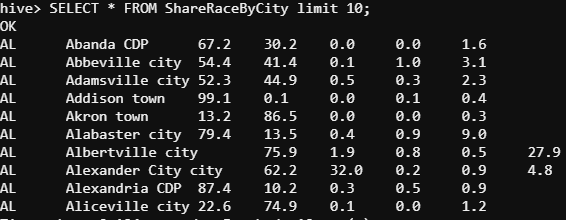
Here, we are using sg Database on Hive and creating table ShareRaceByCity.



Then we are loading ShareRaceByCity.csv file from HDFS into ShareRaceByCity table in SG database on Hive interface in CloudxLab.



Here, we are loading top ten entries from ShareRaceByCity table:



## The insights drawn out of the analysis on Hive interface in CloudxLab:

### The dataset on which the analysis is being done:

1) PoliceKillingsUS.csv

2) ShareRaceByCity.csv.

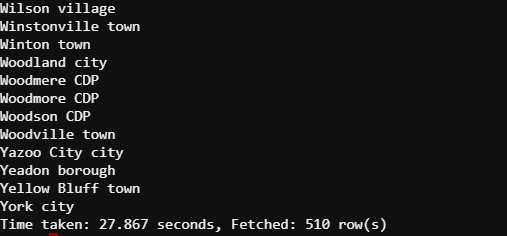
### Analyzing ShareRaceByCity table:

Here, we are querying to find all the cities where share of white Race is more than 70%.



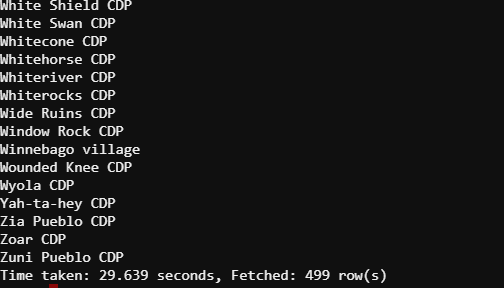
***Insights drawn:*** As we can see that the query has fetched 20242 rows which indicate the count of that many cities where white Race inhabitants are more than 70%.

Here, we are querying to find all the cities where share of Black Race is more than 70%.



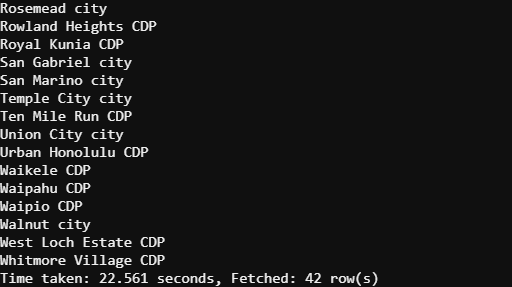
***Insights drawn:*** As we can see that the query has fetched 510 rows which indicate the count of that many cities where Black Race inhabitants are more than 70%.

Here, we are querying to find all the cities where share of Native American Race is more than 70%.



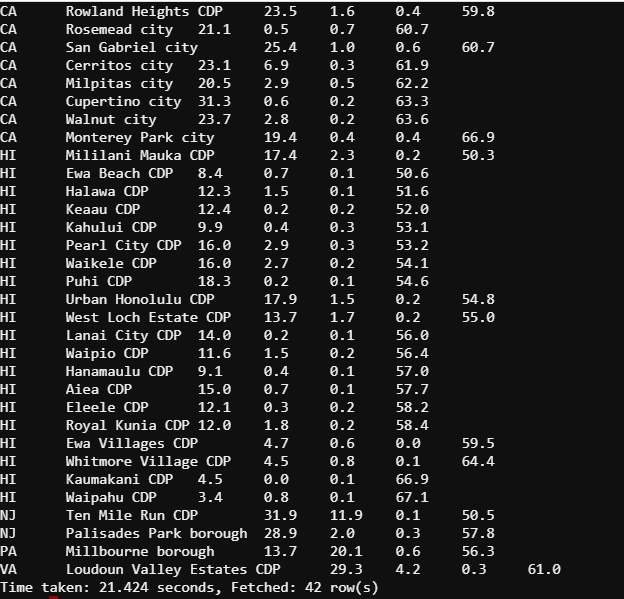
***Insights drawn:*** As we can see that the query has fetched 499 rows which indicate the count of that many cities where Native American Race inhabitants are more than 70%.

Here, we are querying to find all the cities where share of Asian Race is more than 50%.



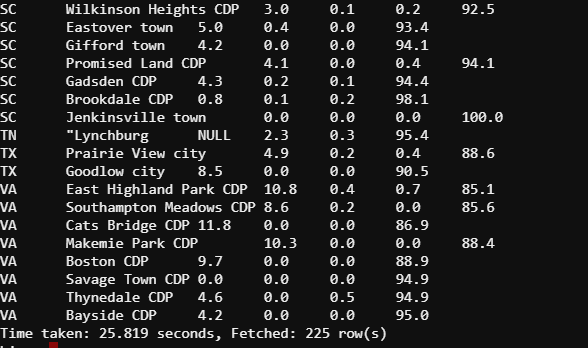
***Insights drawn:*** As we can see that the query has fetched 42 rows which indicate the count of that many cities where Asian Race inhabitants are more than 50%.

Now we are trying to find those geographical areas where Asian are more than 50% and what are the other Race which lives more in numbers at places inhabited by Asians.



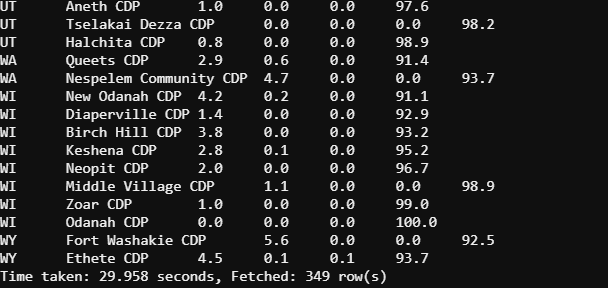
***Insights drawn:*** we can see the following Geographical regions namely CA and HI are the regions where Asians stay more frequently with more than 50% of the population. Also White RACE population is the next higher population of race of peoples staying more frequently with Asians.

Now we are trying to find those geographical areas where Blacks are more than 85% and what is the other Race which lives more in numbers at places inhabited by Blacks.



***Insights drawn:*** we can see the following Geographical regions namely AL, AR, FL, GA, IL, LA, MD, MO,MS,NC,SC, and VA are the regions where Blacks stay more frequently with more than 85% of the population. Also White and Asian populations are race of peoples staying more frequently with Blacks.

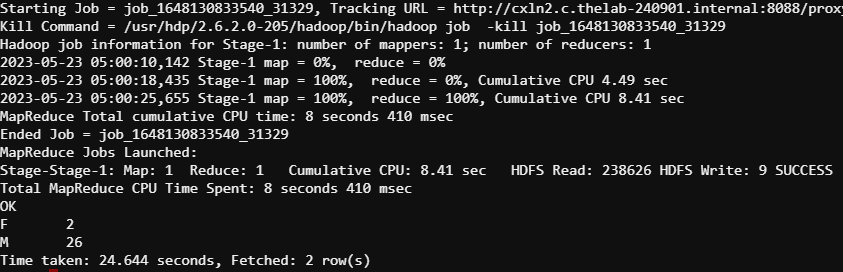
Now we are trying to find those geographical areas where Native Americans are more than 90% and what is the other Race which lives more in numbers at places inhabited by Blacks.



***Insights drawn:*** we can see the following Geographical regions namely AK, AZ, MT, NM, and SD are the regions where Native Americans stay more frequently with more than 90% of the population.

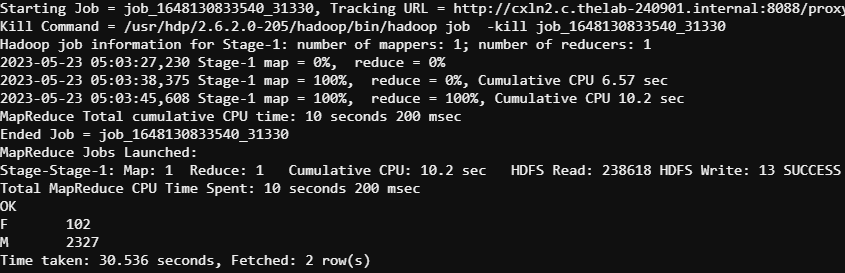
### Analyzing PoliceKillingsUS table:

Here, we are querying to find all those who died in police killings and were less than 16 years old.



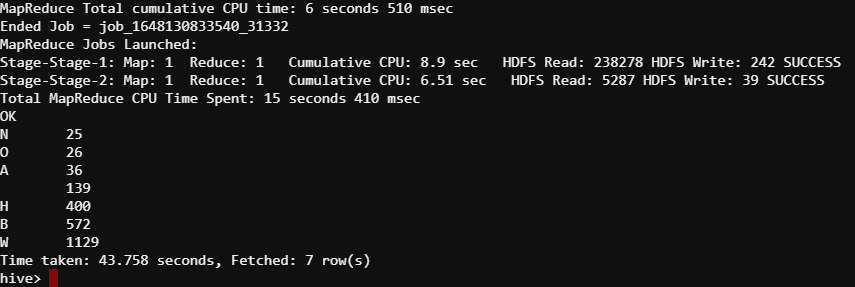
***Insights drawn:*** we can see only 2 females and 26 males who died in police killings and were less than 16 years old.

Here, we are querying to find all those who died in police killings and were more than 16 years old.



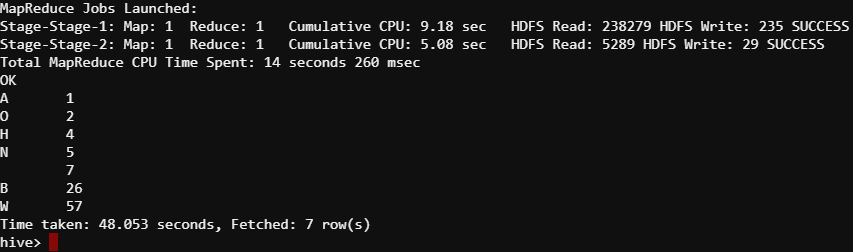
***Insights drawn:*** we can see higher number of males and females having more than 16 years age dying in police killings compared to those who were less than 16 years old.

Here, we are trying to find the categories of race who were males and greater than 16 years age dying in police killings.



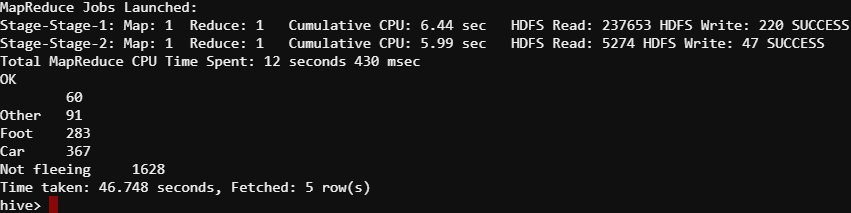
***Insights drawn:*** we can see race W, B, H among males with more than 16 years dying more frequently in police killings compared to other races.

Here, we are trying to find the categories of race who were females and greater than 16 years age dying in police killings.



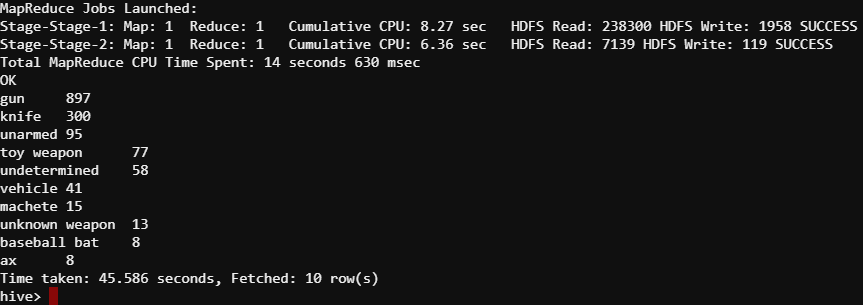
***Insights drawn:*** we can see race B, H among Females with more than 16 years dying more frequently in police killings compared to other races.

Here, we are trying to find all those ways in which the dead tried to flee and died in police killings.



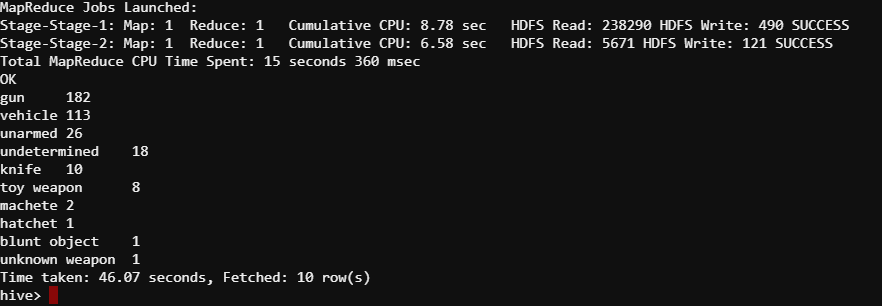
***Insights drawn:*** we can see those who were not fleeing died the most in police killings followed by those who were driving cars and were on foot.

Here, we are trying to find the weapons carried by those who were not fleeing and were greater than 16 years and had died in police killings.



***Insights drawn:*** we can see those who were not fleeing and carried guns and knife more probably to attack died the most in police killings.

Here, we are trying to find the weapons carried by those who were driving cars and were greater than 16 years and had died in police killings.



***Insights drawn:*** we can see those who were driving cars and carried guns more probably to attack died the most in police killings.

Here, we are trying to find the cities where maximum peoples had died in police killings and who were also more than 16 years old.

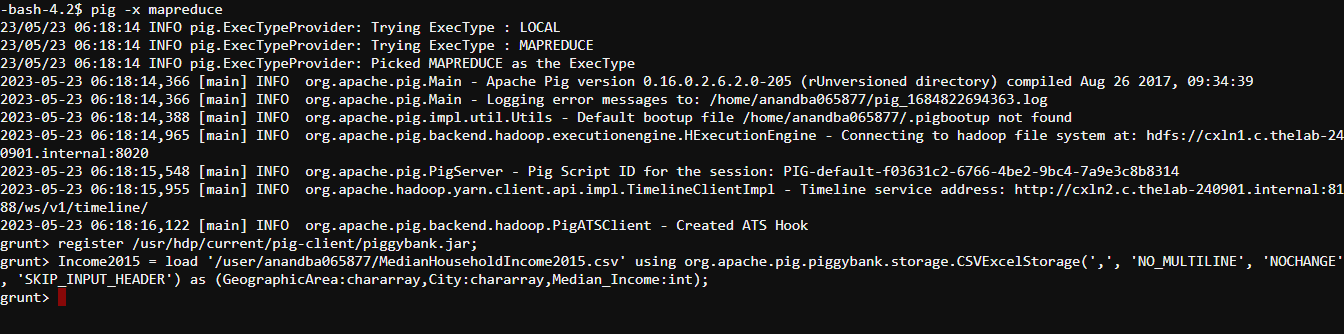


***Insights drawn:*** Los Angeles, Phoenix, Houston, Chicago and Las Vegas were the top 5 cities from where Maximum people died in in police killings.

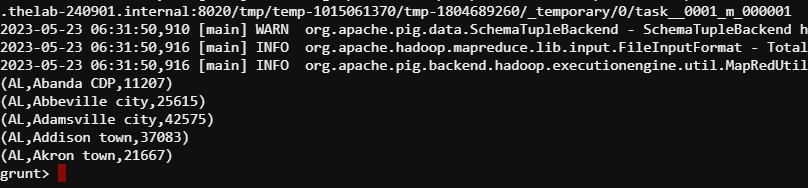
# Implementation using PIG and MAPREDUCE interface in CloudxLab

## Loading file MedianHouseholdIncome2015.csv file on PIG Interface:

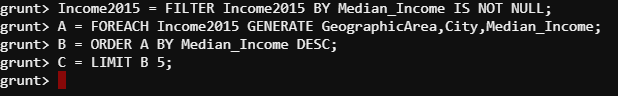
Here, we are loading contents of MedianHouseholdIncome2015.csv file into the object Income2015.



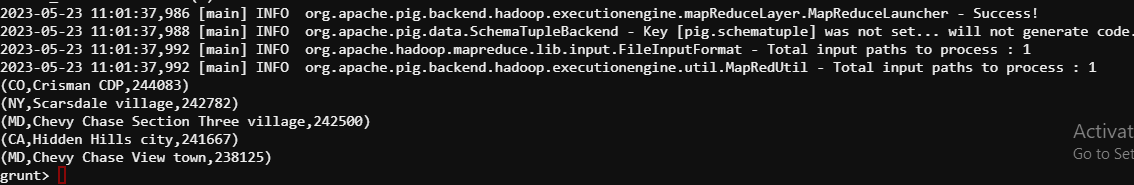
Here, we are loading top five entries from the object Income2015 and determining the data types of the columns being used:



Here, we are filtering null values from the column Median\_Income in the object Income2015 and then follow steps so as load top 5 Median incomes:

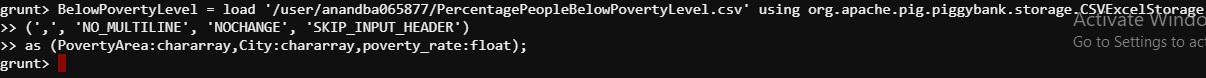


Here, we are loading top five Median incomes and other: entries from the object Income2015:

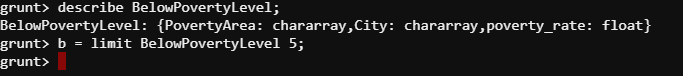


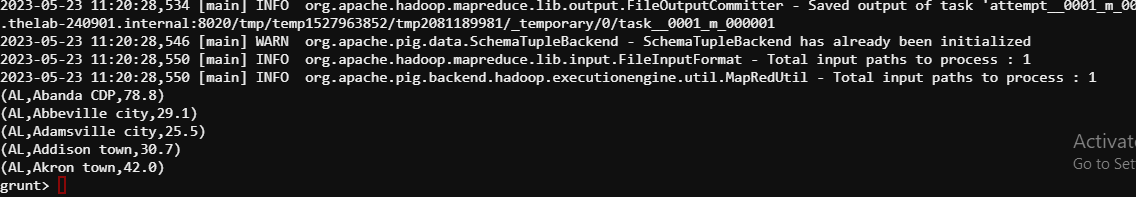
## Loading file PercentagePeopleBelowPovertyLevel.csv file on PIG Interface:

Here, we are loading contents of PercentagePeopleBelowPovertyLevel.csv file into the object BelowPovertyLevel.

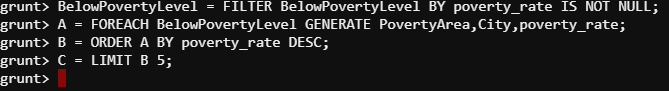


Here, we are loading top five entries from the object BelowPovertyLevel and determining the data types of the columns being used:

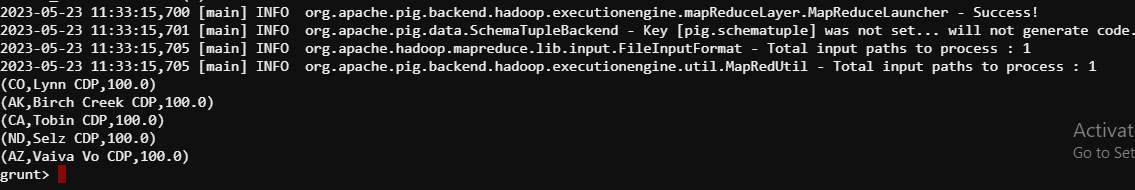




Here, we are filtering null values from the column poverty\_rate in the object BelowPovertyLevel and then follow steps so as load top 5 poverty\_rate:

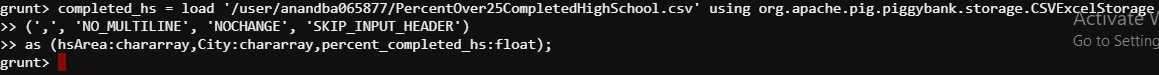


Here, we are loading top five poverty\_rate and other: entries from the object BelowPovertyLevel:

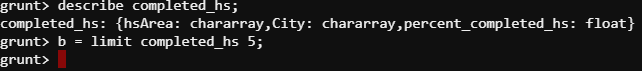


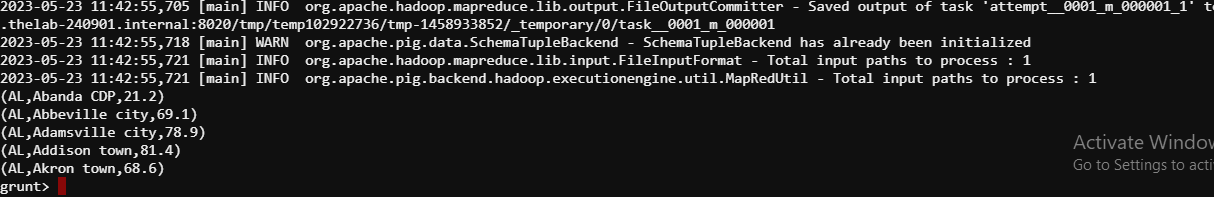
## Loading file PercentOver25CompletedHighSchool.csv file on PIG Interface:

Here, we are loading contents of PercentOver25CompletedHighSchool.csv file into the object completed\_hs.

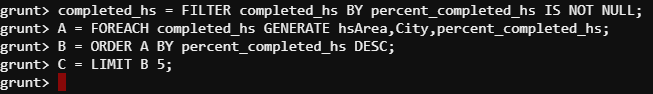


Here, we are loading top five entries from the object completed\_hs and determining the data types of the columns being used:

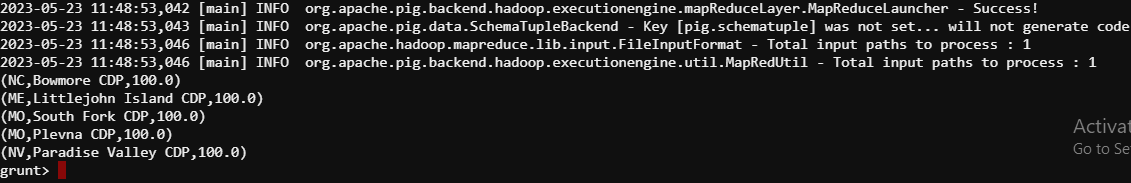




Here, we are filtering null values from the column percent\_completed\_hs in the object completed\_hs and then follow steps so as load top 5 percent\_completed\_hs:



Here, we are loading top five percent\_completed\_hs and other: entries from the object completed\_hs:



### The dataset on which the analysis is being done:

1) MedianHouseholdIncome2015.csv

2) PercentagePeopleBelowPovertyLevel.csv

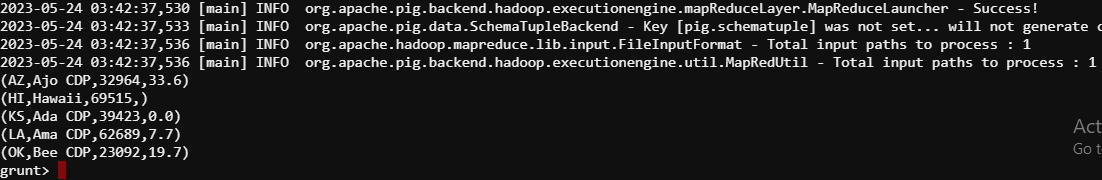
3) PercentOver25CompletedHighSchool.csv.

### Analyzing Income\_scan table:

Here we are creating Income\_scan object by left outer join of Income2015 and BelowPovertyLevel objects in PIG interface in CloudxLab. To prevent duplication of columns we are only selecting column 0, column1, column2 and column 5 of Income\_scan object and reassigning the same back to Income\_scan object



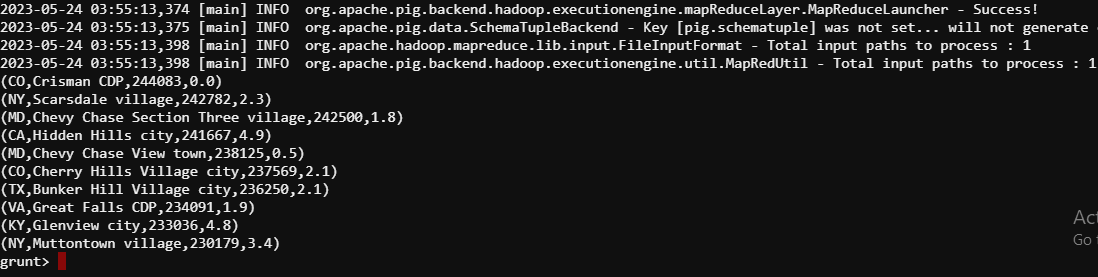
Here, we are loading top five entries from the object Income\_scan:



Here, we are querying to extract descending ten median incomes listings from income scan object to compare and analyze poverty rate and Median Income for people living in cities as shown below:



Here, we are loading descending ten Median Income and other entries from the object income scan:

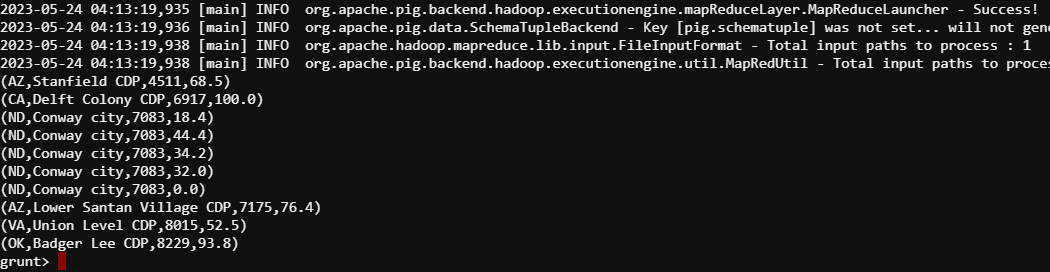


***Insights drawn:*** As we can see, the query returned the descending ten Median Income entries, indicating that poverty rates are lower in cities with greater median incomes. The poverty rate typically increases when median income declines. Another conclusion is that depending on the general income ranges of the cities, the calculation of the poverty rate varies for each city. Also, the top ten cities with decreasing Median Income come from following geographical regions namely CO, NY, MD, CA, MD, CO, TX, VA, KY, and NY.

Here, we are querying to extract increasing ten median incomes listings from income scan object to compare and analyze poverty rate and Median Income for people living in cities as shown below:



Here, we are loading increasing ten Median Income and other entries from the income scan object:

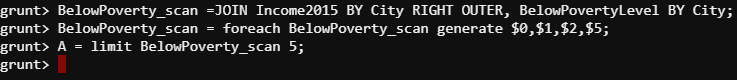


***Insights drawn:*** As we can see, the query produced ten records with increasing Median Income, showing that poverty rates often decline as median income increases. Another conclusion is that the calculation of the poverty rate differs for each city depending on the general income ranges of the cities. Also, the bottom ten cities with least Median Income come from following geographical regions namely AZ, CA, ND, ND, ND, ND, ND, AZ, VA, and OK.

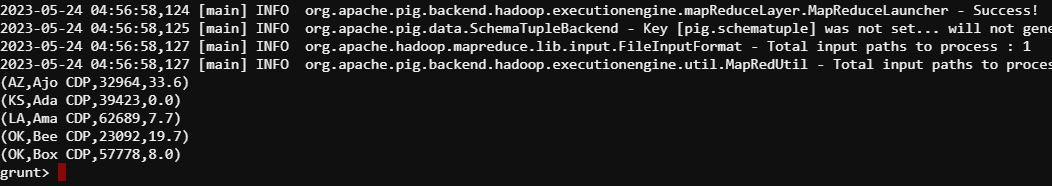
Here, we are storing income scan object into HDFS storage and then loading the HDFS file back into our local storage as well.

### Analyzing BelowPoverty\_scan table:

Here we are creating BelowPoverty\_scan object by right outer join of Income2015 and BelowPovertyLevel objects in PIG interface in CloudxLab. To prevent duplication of columns we are only selecting column 0, column1, column2 and column 5 of BelowPoverty\_scan object and reassigning the same back to BelowPoverty\_scan.



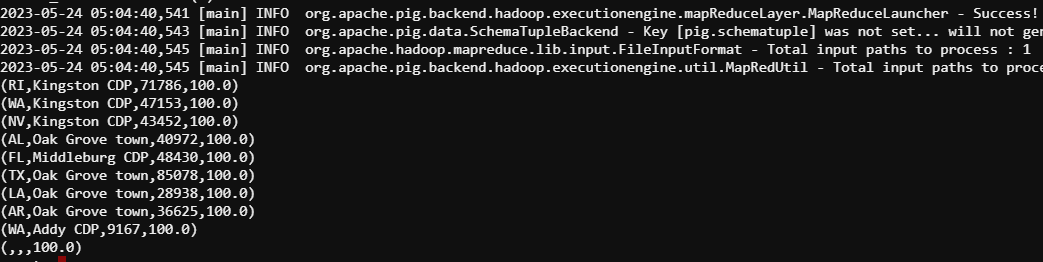
Here, we are loading top five entries from the BelowPoverty\_scan object:



Here, we are querying to extract descending ten poverty\_rate listings from BelowPoverty\_scan object to compare and analyze poverty rate and Median Income for people living in cities as shown below:



Here, we are loading descending ten poverty\_rate listings and other entries from BelowPoverty\_scan object:

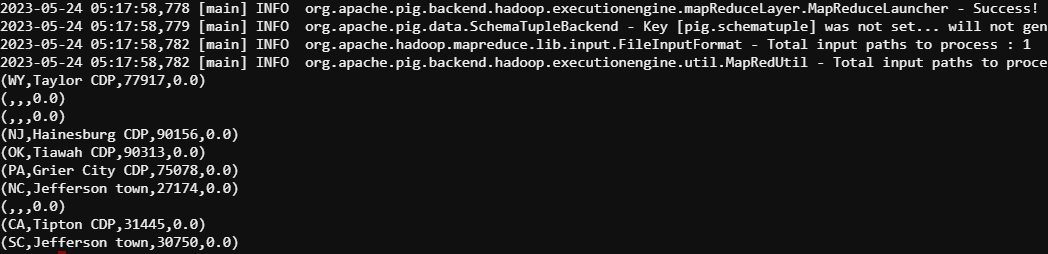


***Insights drawn:*** Usually, when the median income falls, the poverty rate rises. The determination of the poverty rate, however, differs for each city depending on the general income ranges of the cities. Additionally, the bottom 10 cities with the highest poverty rates are located in the following regions: RI, WA, NV, AL, FL, TX, LA, AR, and WA.

Here, we are querying to extract increasing ten poverty rate listings from BelowPoverty\_scan object to compare and analyze poverty rate and Median Income for people living in cities as shown below:



Here, we are loading increasing ten poverty rate and other entries from the BelowPoverty\_scan object:

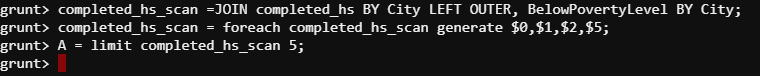


***Insights drawn:*** As we can see, the search returned 10 records with poverty rate listings that ought to have been increasing However; it is clear from the data that the calculation of poverty rates differs for each city depending on the general income ranges of the cities. WY, NJ, OK, PA, NC, CA, and SC are among the states where the top 10 cities with the lowest rates of poverty are situated.

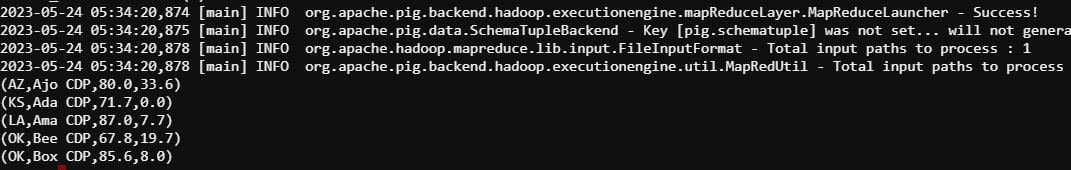
Here, we are storing BelowPoverty\_scan object into HDFS storage and then loading the HDFS file back into our local storage as well.

### Analyzing completed\_hs\_scan table:

Here we are creating completed\_hs\_scan object by left outer join of completed\_hs and BelowPovertyLevel objects in PIG interface in CloudxLab. To prevent duplication of columns we are only selecting column 0, column1, column2 and column 5 of completed\_hs\_scan object and reassigning the same back to completed\_hs\_scan.



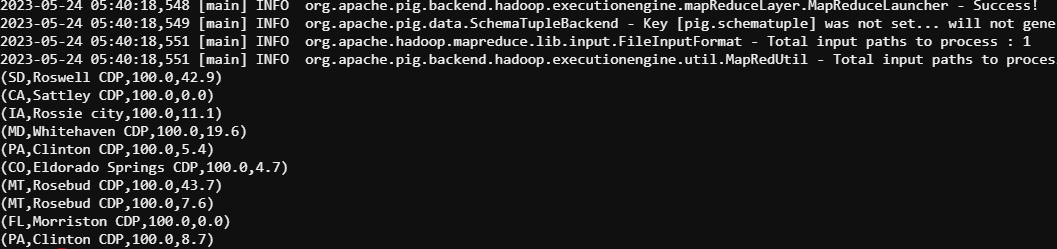
Here, we are loading top five entries from the completed\_hs\_scan object:



Here, we are querying to extract descending ten percent\_completed\_hs listings from completed\_hs\_scan object to compare and analyze poverty rate and percent\_completed\_hs for people living in cities as shown below:



Here, we are loading descending ten percent\_completed\_hs listings and other entries from completed\_hs\_scan object:

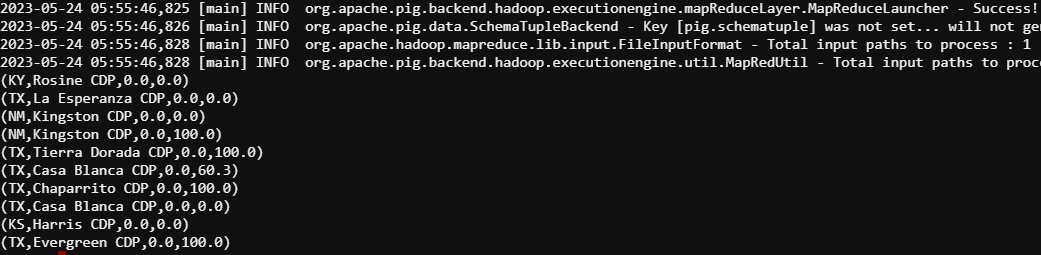


***Insights drawn:*** Cities with larger percentages of residents who have completed their higher secondary education typically don't appear to be associated to the poverty rate graph. Additionally, the following regions are home to the bottom 10 cities with the highest percentage of residents finishing higher secondary education: IA, MD, PA, CO, MT, MT, FL, and PA

Here, we are querying to extract increasing ten percent\_completed\_hs from completed\_hs\_scan object to compare and analyze poverty rate and percent\_completed\_hs for people living in cities as shown below:



Here, we are loading increasing ten percent\_completed\_hs and other entries from the completed\_hs\_scan object:



***Insights drawn:*** We can conclude from the observations listed above that cities with even the lowest rates of poverty have the lowest percentage of citizens who have completed their higher secondary school, but sometimes the results are exactly the contrary. Additionally, The bottom 10 cities with the lowest percentage of residents completing higher secondary education are located in the following regions : KY,TX,NM,NM,TX,TX,TX,TX,KS, and TX.

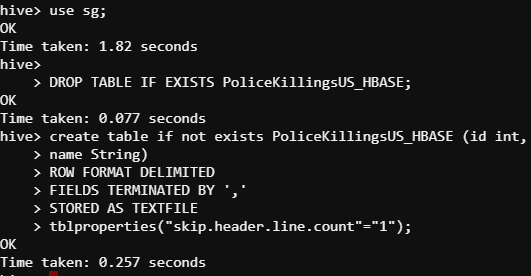
Here, we are storing completed\_hs\_scan object into HDFS storage and then loading the HDFS file back into our local storage as well.

# SQOOP

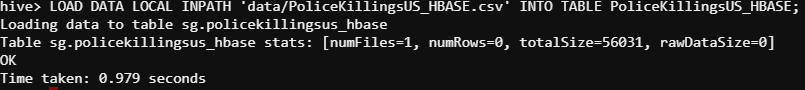
It is an Open source tool to efficiently transferring bulk data between Hadoop components (HDFS, Hive, H base) and structured data stores such as MySQL, Oracle, and PostgreSQL.

## Sqoop Export - Hive to MySQL

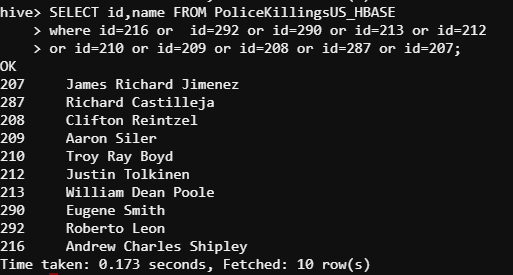
Here, we are using sg Database on Hive and creating table PoliceKillingsUS\_HBASE.



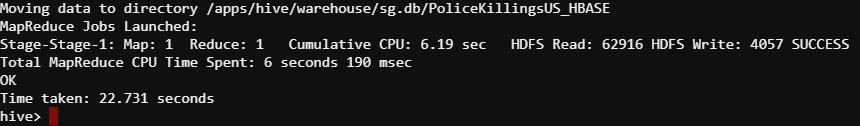
Then we are loading PoliceKillingsUS\_HBASE.csv file into PoliceKillingsUS\_HBASE table.



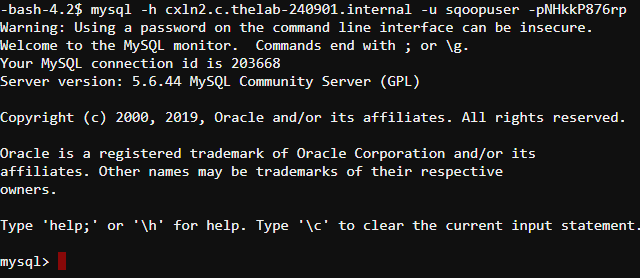
Here, we are loading top ten entries from PoliceKillingsUS\_HBASE table:



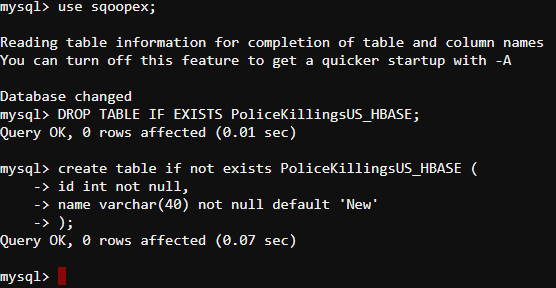
Here, we are overwriting directory ' /apps/hive/warehouse/sg.db/PoliceKillingsUS\_HBASE' with the contents from PoliceKillingsUS\_HBASE table with the help of MAP REDUCE program running in the background



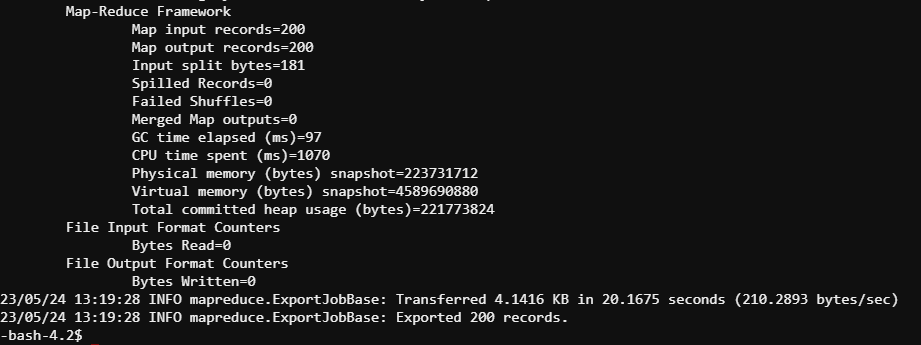
Here, we are launching MySQL in CloudxLab



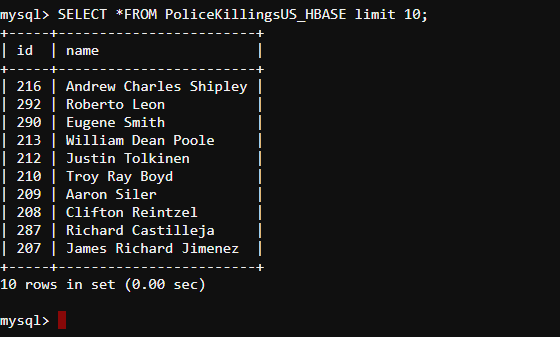
Here we are using sqoopex database and creating PoliceKillingsUS\_HBASE table in MySQL



Here we are performing Sqoop Export - Hive to MySQL with the help of MAP REDUCE program running in the background:

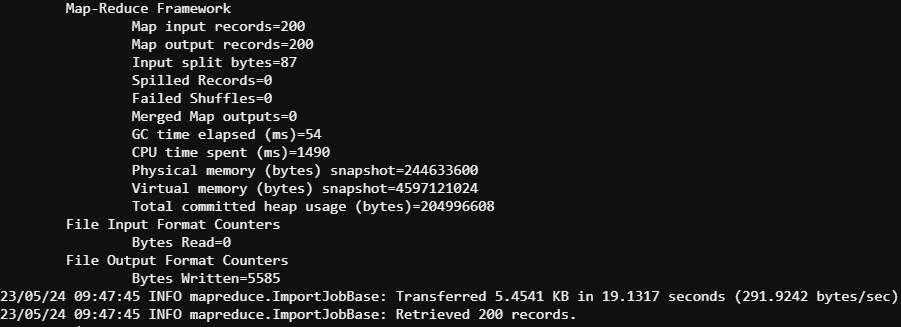


Here, we are loading top ten entries from PoliceKillingsUS\_HBASE table in MySQL interface in CloudxLab:

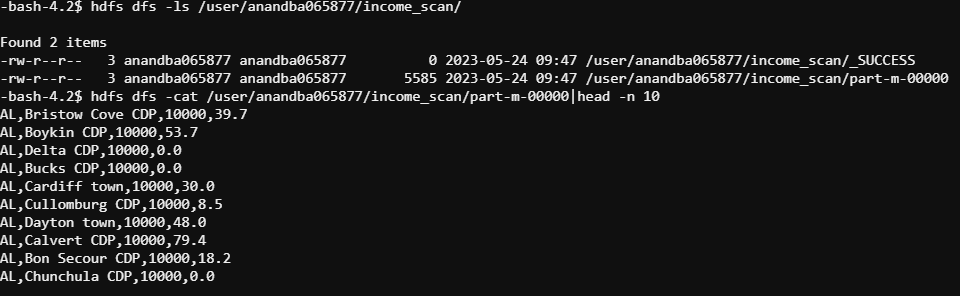


## Sqoop Import - MySQL to HDFS

Importing income\_scan Table from sqoopex database in MYSQL to HDFS at the location /user/anandba065877/income\_scan

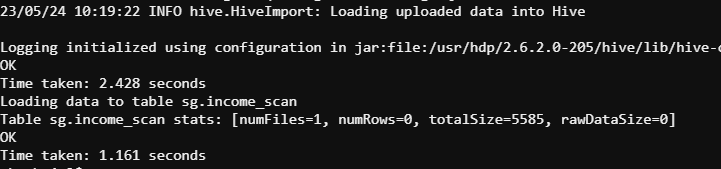


Displaying top 10 entries of the income\_scan Table imported from sqoopex database in MYSQL to HDFS at the location /user/anandba065877/income\_scan

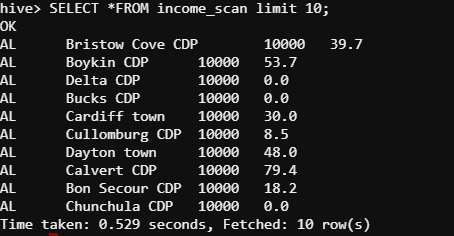


## Sqoop Import - MySQL to HIVE

Importing income\_scan Table from sqoopex database in MYSQL to HIVE in sg database:

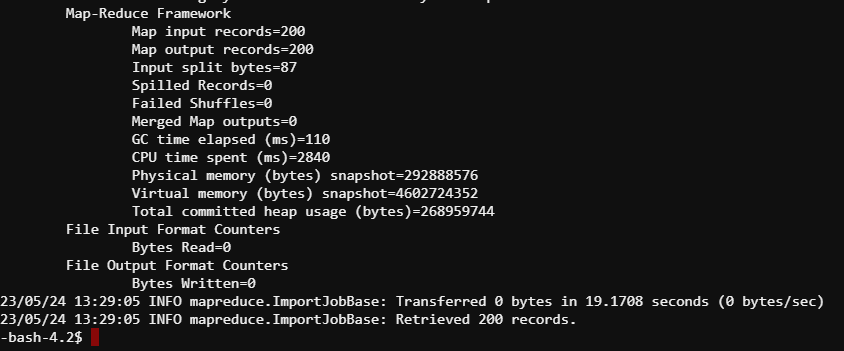


Displaying top 10 entries of the income\_scan Table imported from sqoopex database in MYSQL to HIVE in sg Database:

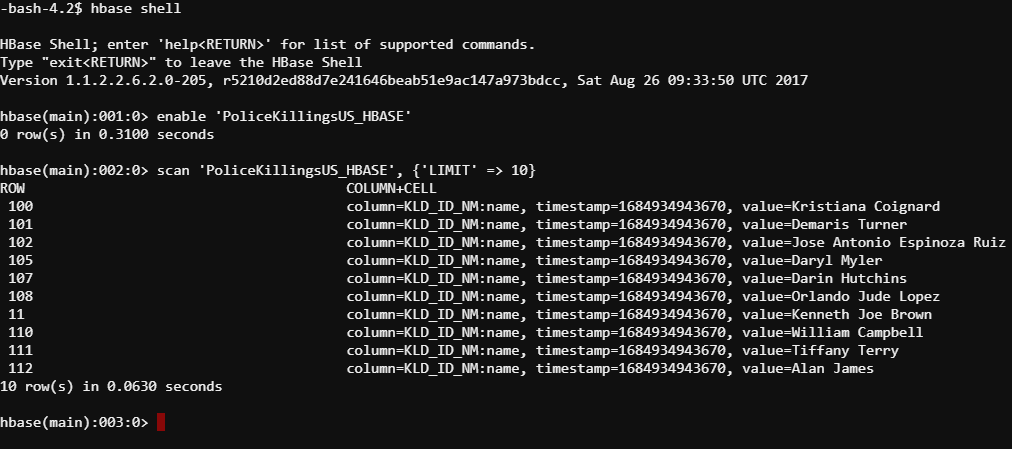


## Sqoop Import - MySQL to HBase

Importing PoliceKillingsUS\_HBASE Table from sqoopex database in MYSQL to HBase



Displaying top 10 entries of the PoliceKillingsUS\_HBASE Table imported from sqoopex database in MYSQL to HBase:



# HBASE

It is a column family-oriented data store. It is great for storing data having 100s of millions of records or more. It is based on Google's paper on Big Table. Hbase runs on top of Hadoop meaning it stores data files in HDFS and it can process the data using Map Reduce.

Commands to perform different other operation on HDFS which were used here are:

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////

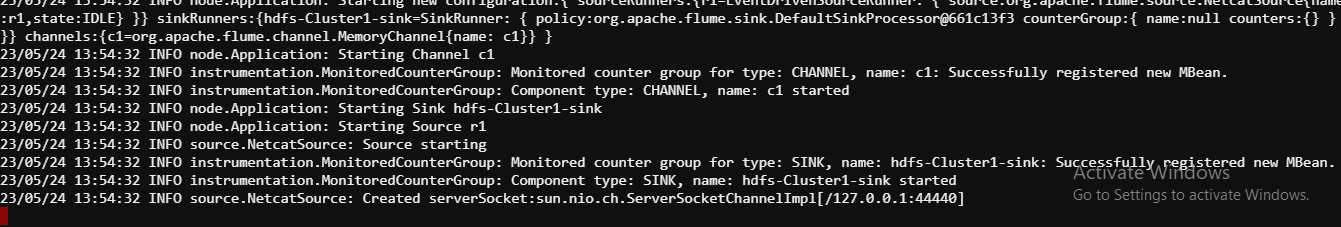
# Flume

Flume is a simple, robust and extensible tool for data ingestion from various data sources into Hadoop. It is used for collecting, aggregating and transporting a large amount of streaming data such as events and logs from various sources to a centralized data store such as HDFS.

#getting a copy of sample flume conf from common data

# Change the port to 44440 and location to hdfs://10.142.1.1/user/anandba065877/flume\_webdata in HDFS

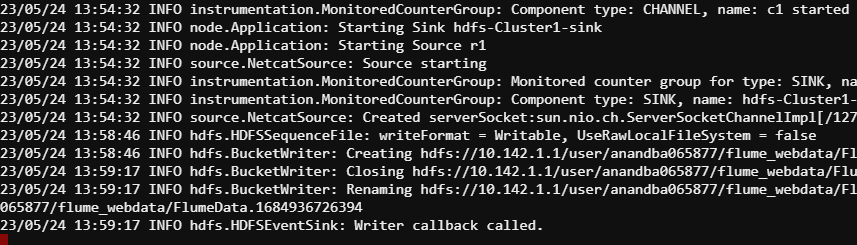
#Launch the flume agent



# Open a new console and Connect to the same port that you defined in config

# Generate some data .Type something in the console.





#Open a new console and Check in HDFS using the following commands:

