**Major project:**

**1. Executive Summary:**

1.1 Project Overview to develop the System to analyze the log data (In XML format) of government progress of various development activities.

1.2 Purpose and Scope of this Specification The purpose of this project is to capture the data for analyzing the progress of various activities. In scope the following requirement will be addressed in phase 1 of Project:

• Developing system to handle the incoming log feed and store the information in Hadoop Cluster (Flume)

• Analyze the data and understand the progress

• Store the results in Hbase/RDBMS Out of scope we can use this data and visualization and get more insights

**2. Product/Service Description:**

2.1 Assumptions Log will be generated in XML format and stored in a server

2.2 Constraints describe any item that will constrain the design options, including

• This system may not be used for searching for now. But it will be used for analysis and saving the relevant information as of now

• System will be using Hbase as a database

**3. Requirements:**

• The FLUME job which will format the data and place the data to HDFS

• Pig/MapReduce job for parsing the XML data.

• Create Pig scripts/MapReduce jobs to analyze the data

• Create the Sqoop job to store the data in database Priority Definitions The following definitions are intended as a guideline to prioritize requirements.

• Priority 1 – Create FLUME job for fetching log files from spool directory the data

• Priority 2 – MapReduce/pig job to preprocess Download the dataset using the below link: Link:

<https://drive.google.com/file/d/0Bxr27gVaXO5sUjd2RWFQS3hQQUE/view?usp=sharing>

Refer the below steps to understand the actual steps to create the above project.

Step 1: Copy dataset from local file system to HDFS using flume.

Note: use the conf file by downloading from below link.

Click here to download Command:

flume-agent agent –n agent1 –c conf –f

Step 2: Input file is in the XML format use Map reduce or pig to parse the data and get the results for the below problem statements.

**4. Problem statement:**

1.

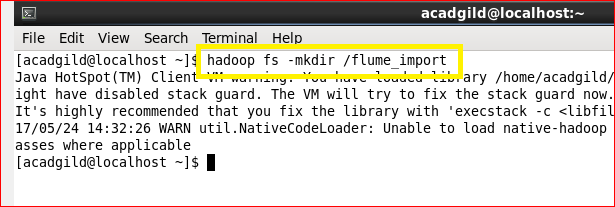
Find out the districts who achieved 100 percent objective in BPL cards Export the results to MySQL using sqoop

2.

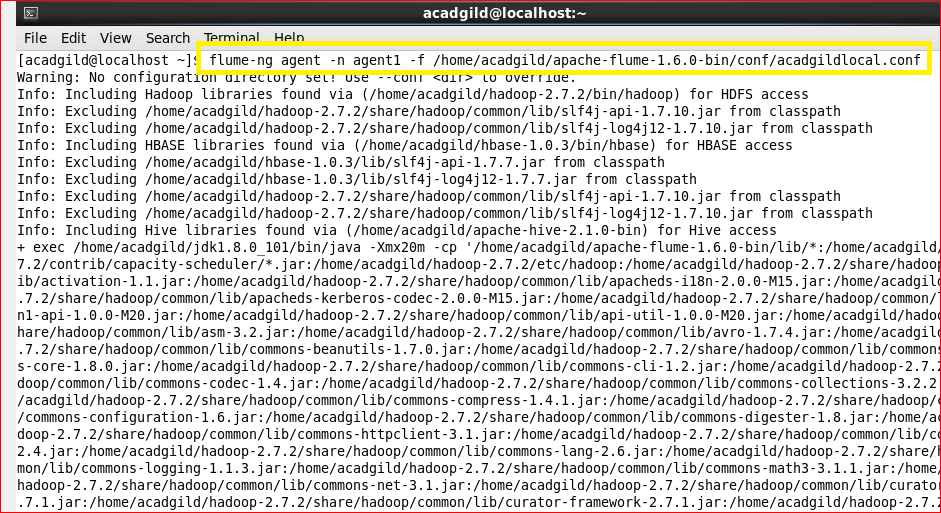
Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards. Export the results to MySQL using Sqoop.

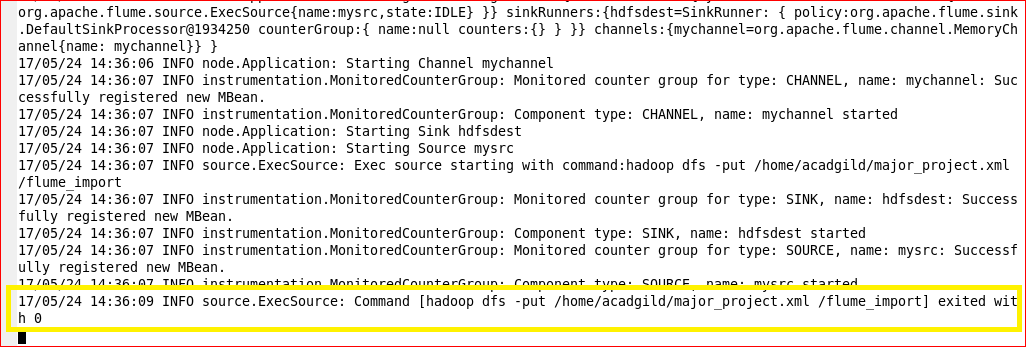
**Data Streaming from Local File System to HDFS using flume Spool directory:**

1.  We need to make destination directory inside HDFS as mentioned in acadgildlocal.conf.



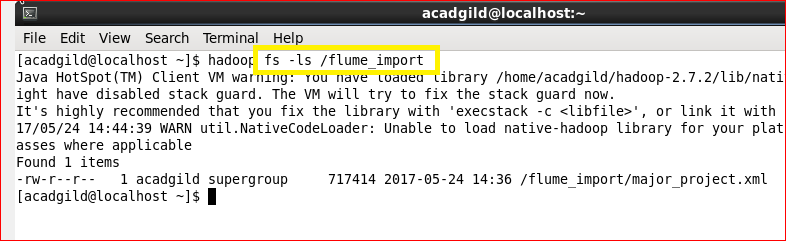
2. Start flume agent by the following command  
Command: flume-ng agent –n agent1 –f /home/acadgild/apache-flume-1.6.0-bin/conf/acadgildlocal.conf





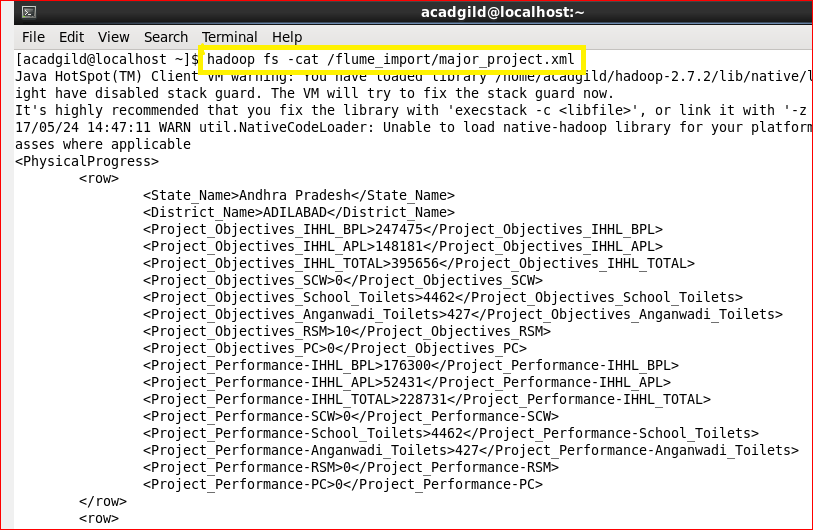
Exited with ‘0’ will indicate that the streaming is completed.

3. We can now check the resultant temporary file inside HDFS at destination path generated by Flume.

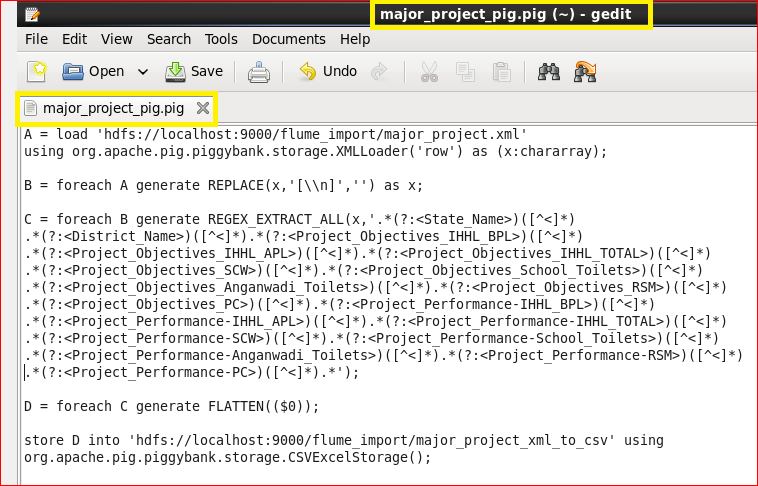


4. The contents streamed into hdfs directory are as follows:

The contents are in .xml format.



**Converting xml file into csv file using pig:**



Step 1: Here we will load the xml file using the default XML loader available in pig.

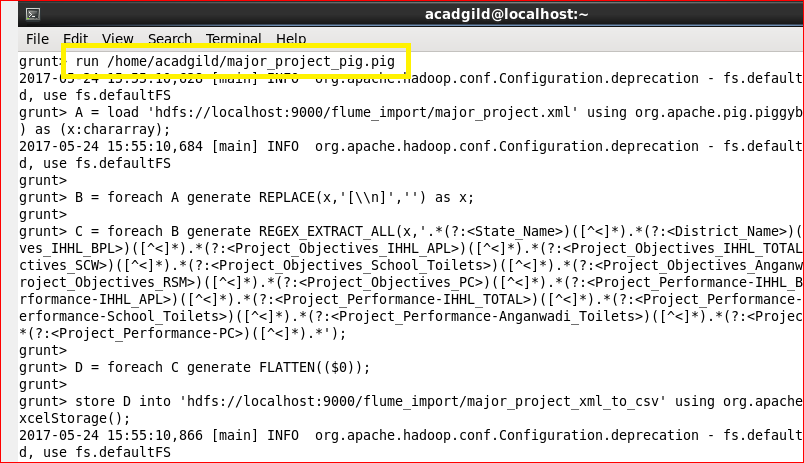
 Inside the XML loader we are specifying that our root element is row and we are storing the whole thing with an alias name *x*as chararray.

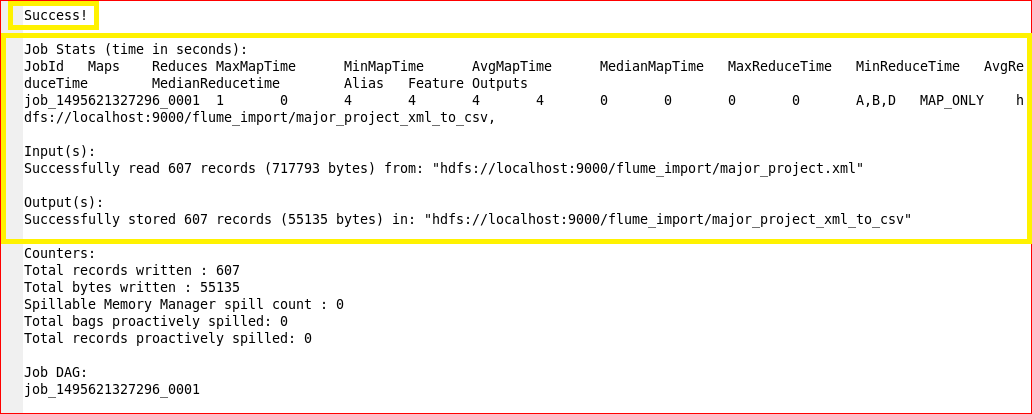
Step 2: Here we are bringing the contents between the *row*tags in one line.

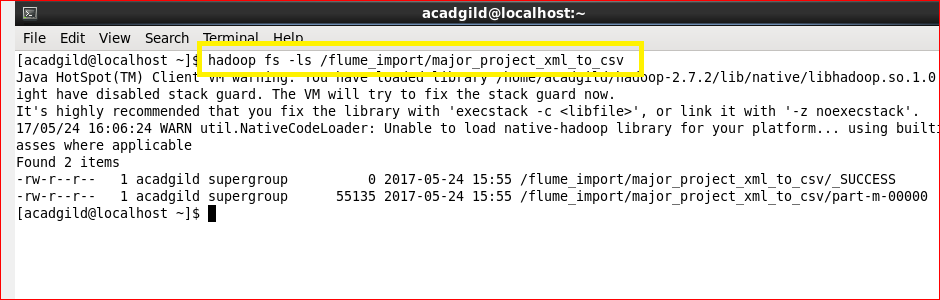
Step 3: Now we are removing the brackets by using the above mentioned regular expression.

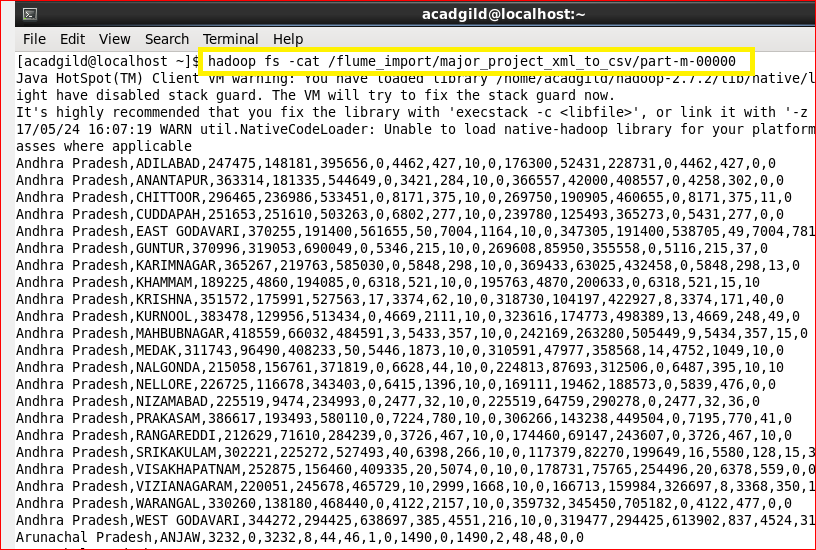
Step 4: Here by using flatten it will remove the remaining brackets. Now the Final result looks like this.

Step 5: This output is stored in the location   /flume\_import/major\_project\_xml\_to\_csv with name part-m-00000 of HDFS.







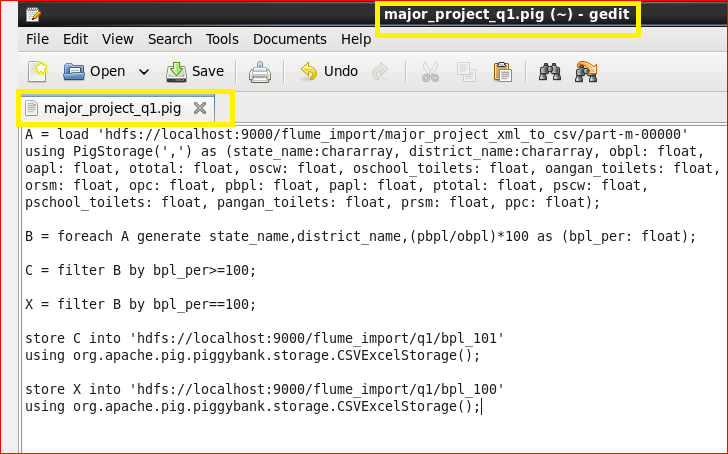


**Problem statement 1:**

Find out the districts who achieved 100 percent objective in BPL cards Export the results to mysql using sqoop

Approach 1: districts which achieved 100% and more than 100%:

Approach 2: districts which achieved exactly 100%:



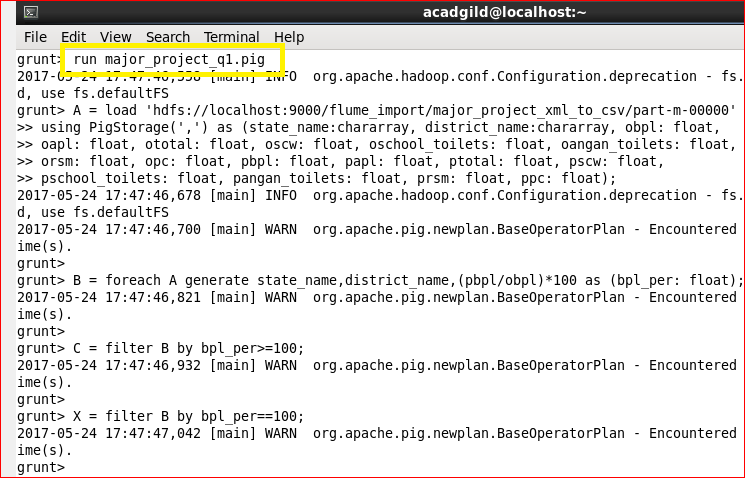
Step 1: load data of csv file into relation A.

Step 2: calculated BPL percentage as (performance\_BPL/objective\_BPL)\*100.

Step 3: filtered relation according to the problem approach.

Step 4: stored result on HDFS.

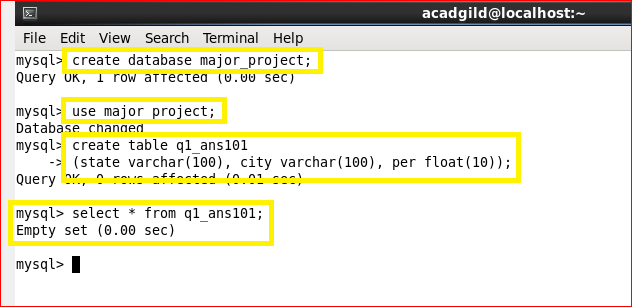
Running pig script:

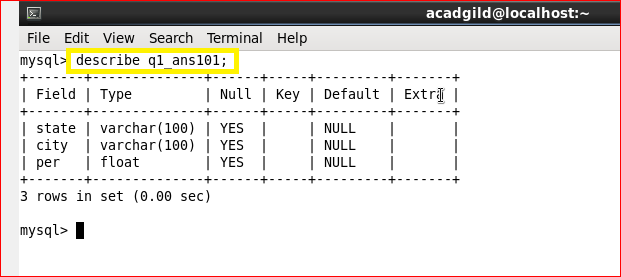


**Exporting data from HDFS to MySQL table:**

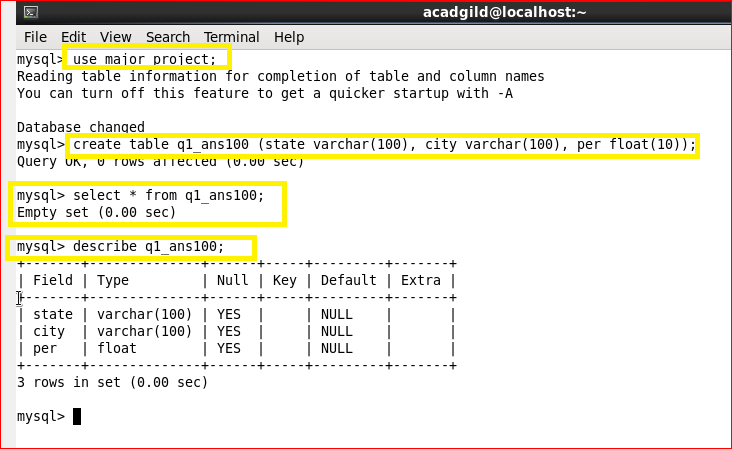
Step 1: first we created table to store result.

1. q1\_ans101 table to stored result of approach 1.



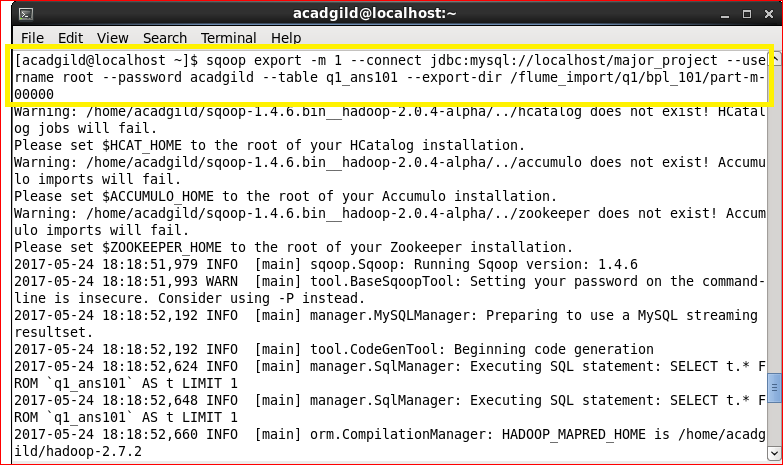


2. q1\_ans100 table to stored result of approach 2.

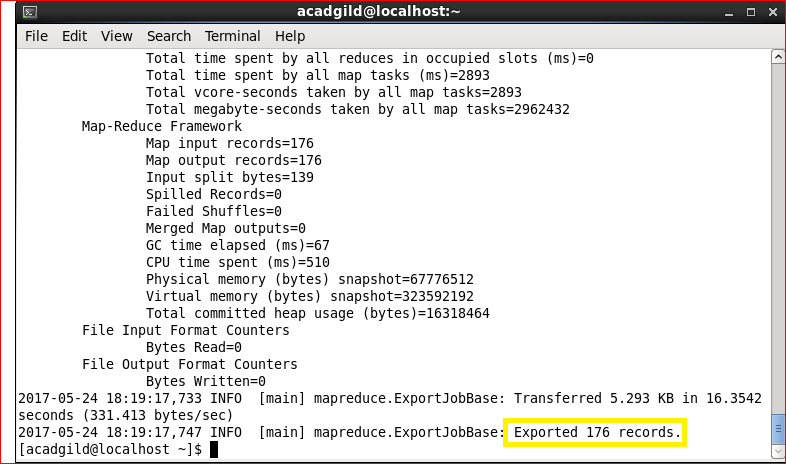


Step 2: now exporting contents from HDFS to MySQL using Sqoop:

Approach 1:

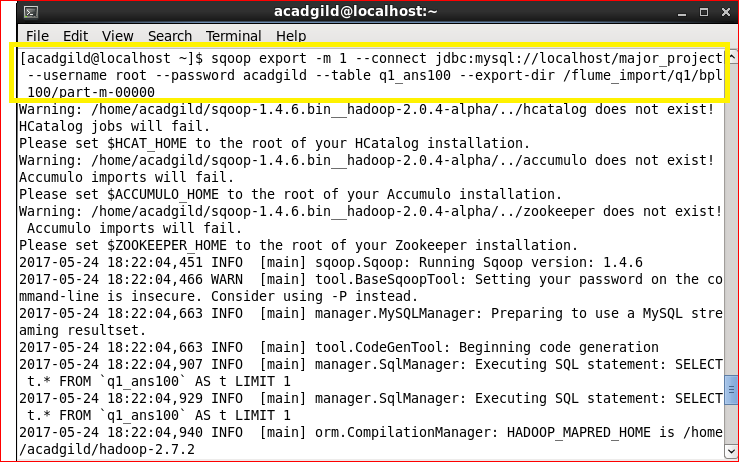


Here –m 1 indicated only one mappers are restricted to 1.

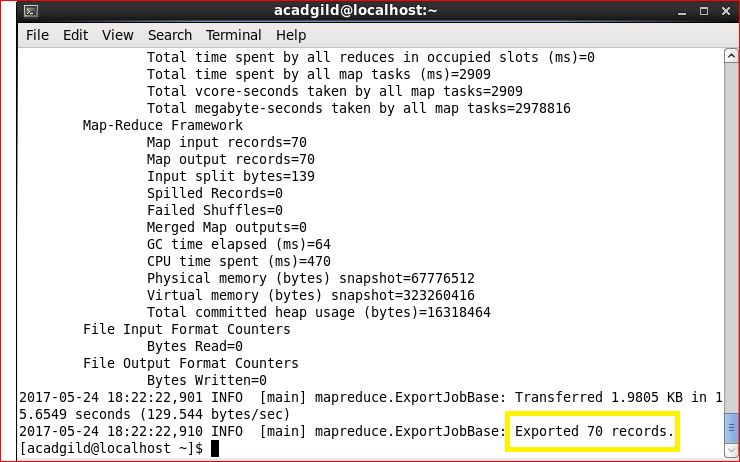


Total 176 records are exported into table q1\_ans101.

Approach 2:



Here –m 1 indicated only one mappers are restricted to 1.

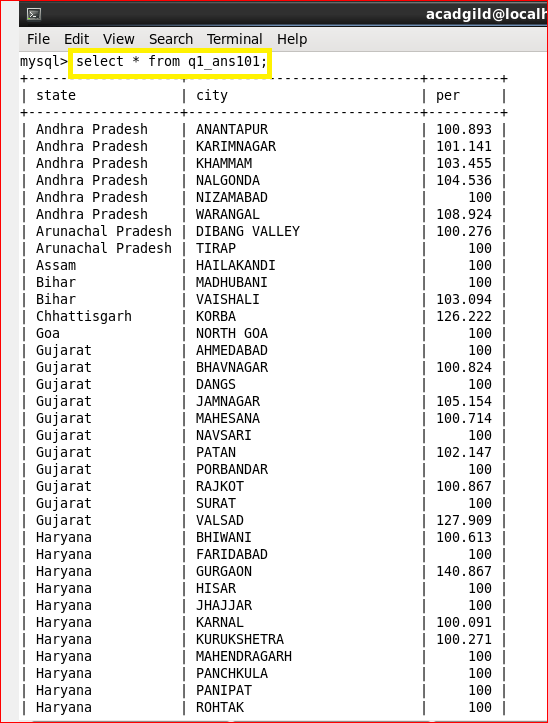


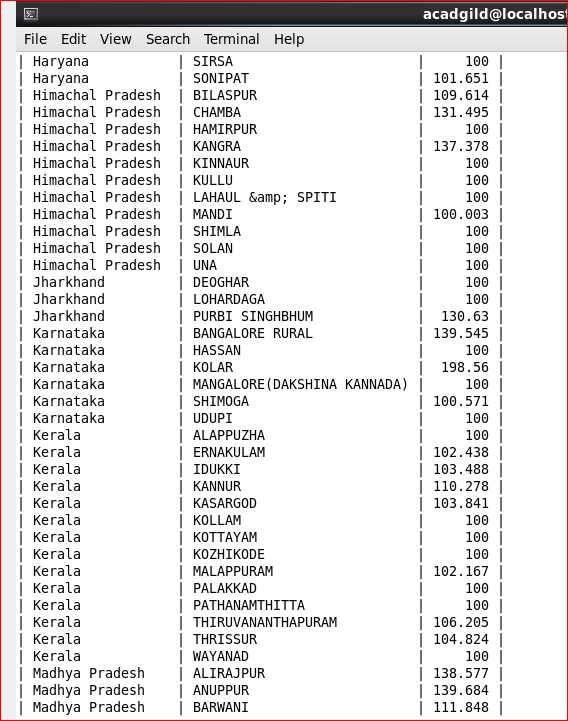
Here 70 records are exported into table q1\_ans100.

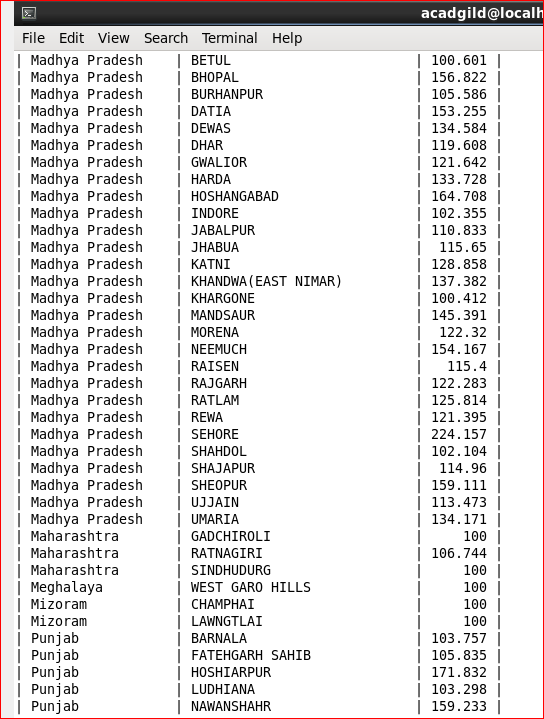
**Outputs:**

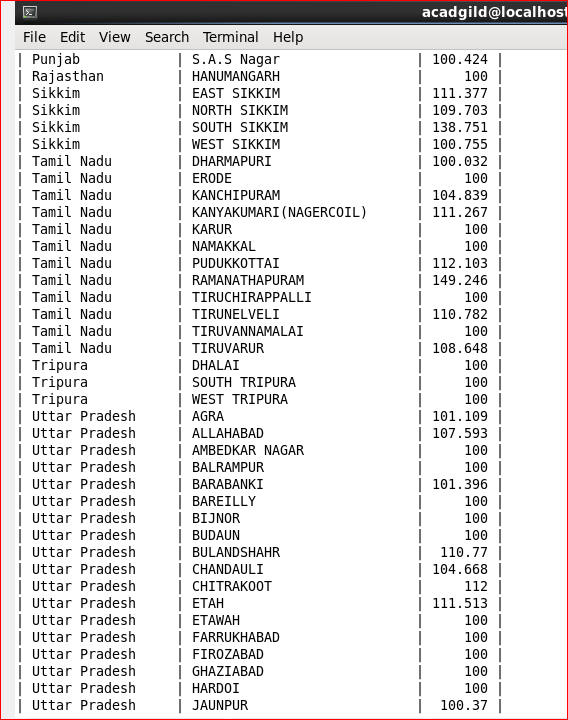
Approach 1: districts which achieved 100% and more than 100%:

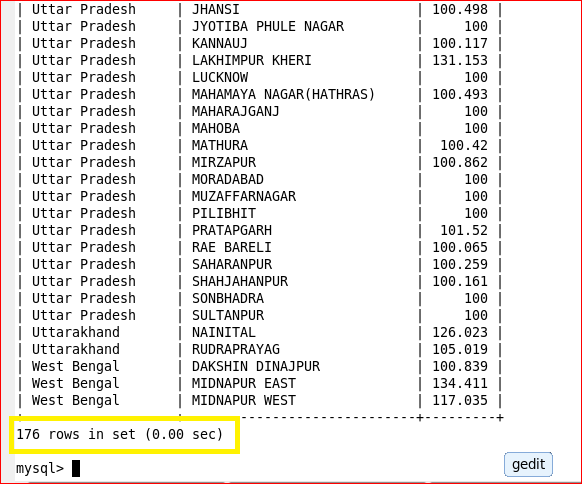
**Total 176 records:**





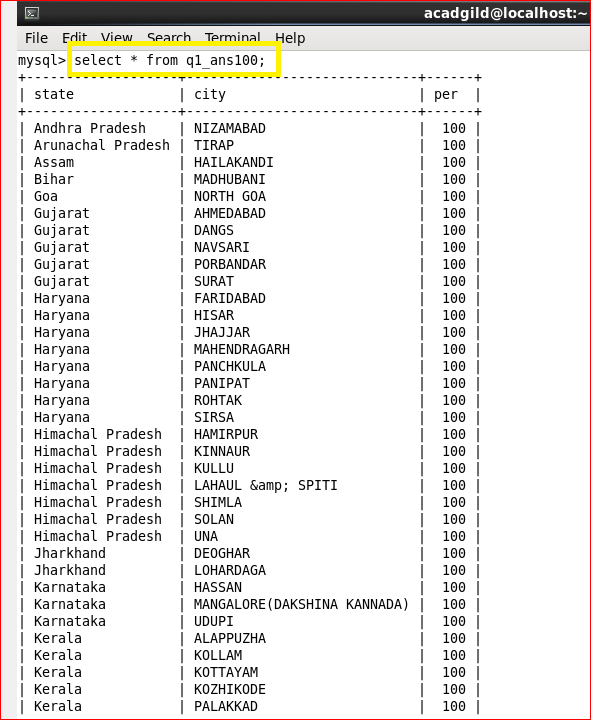


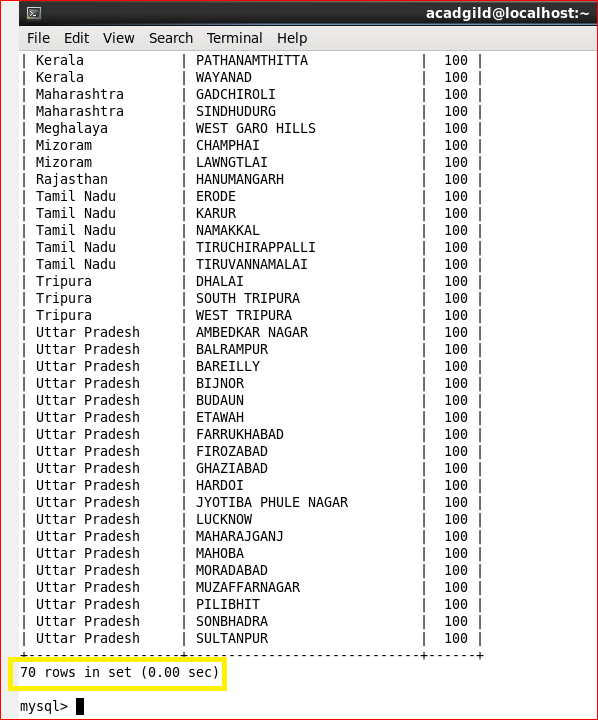




Approach 2: districts which achieved exactly 100%:

**Total 70 records.**

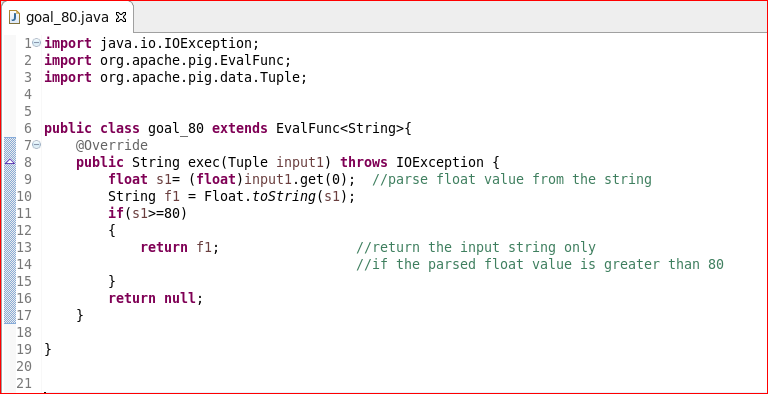


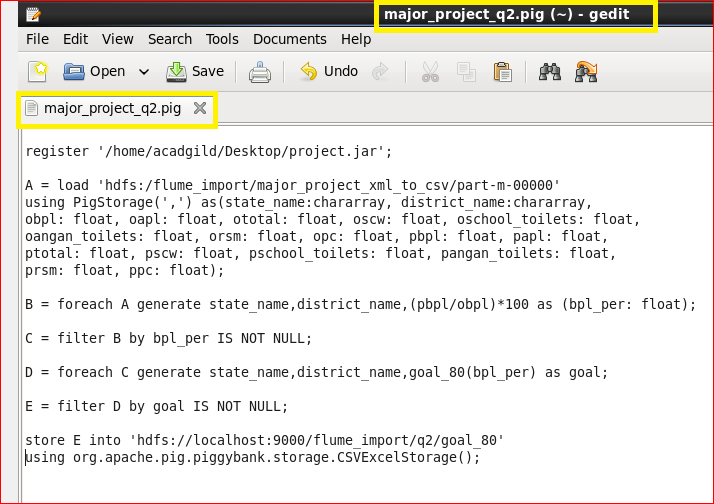


**Problem statement 2:**

Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards. Export the results to MySQL using Sqoop.

UDF class:





Registering the jar for udf.

Step 1: load data into relation A.

Step 2: we calculated BPL percentage in this step.

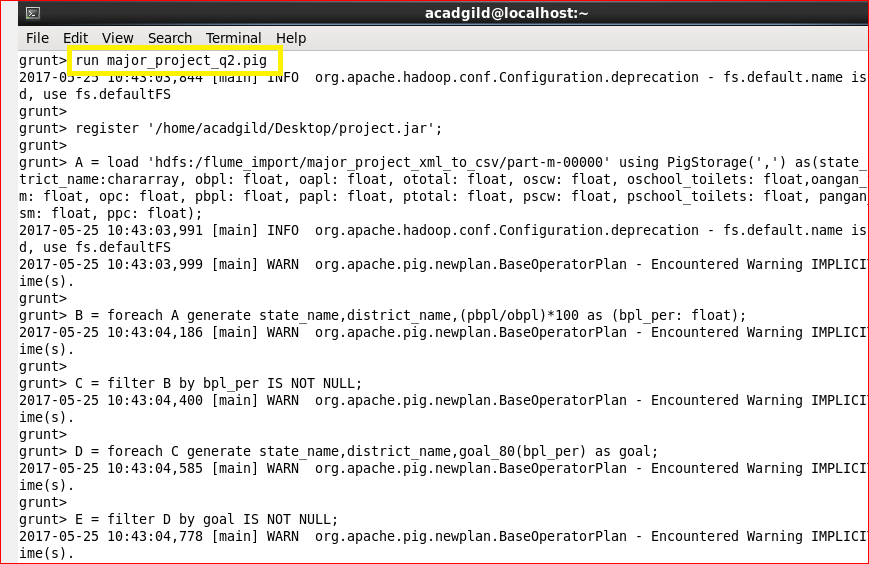
Step 3: filter relation for the null values.

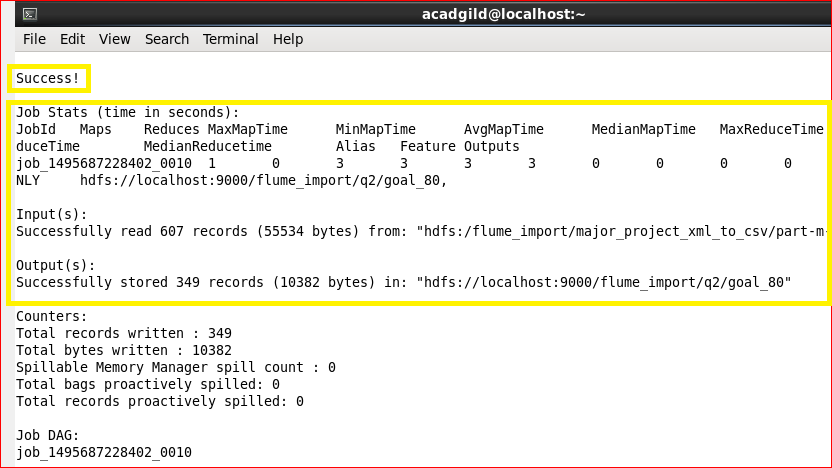
Step 4: now we are using the UDF function that we have created to retain records we have achieved the goal of 80% BPL.

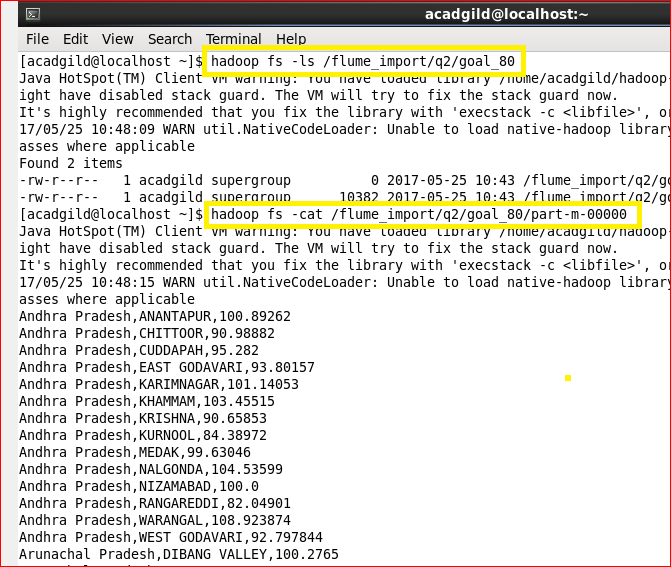
Step 5: filtered relation for null values.

Step 6: stored the result on HDFS file.

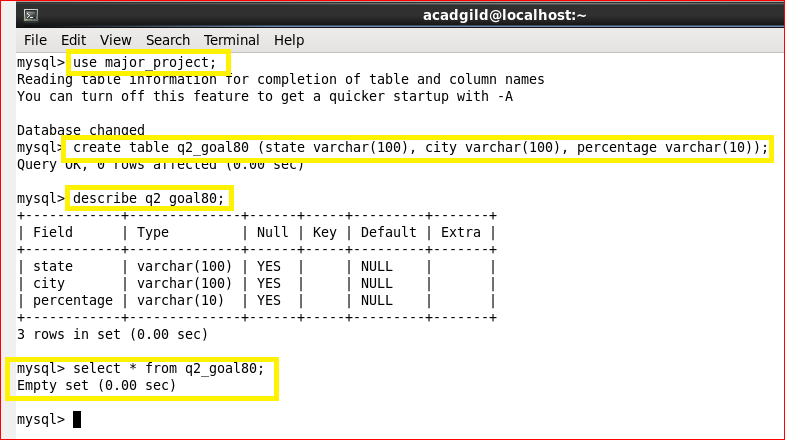
Running pig script:



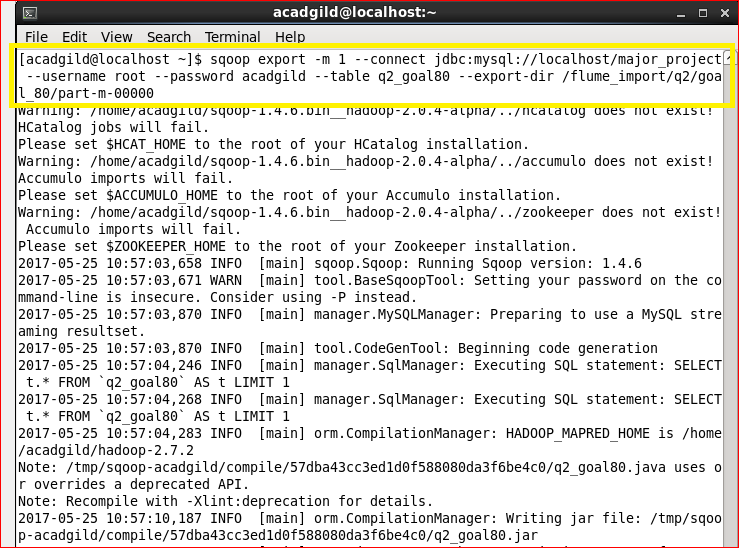


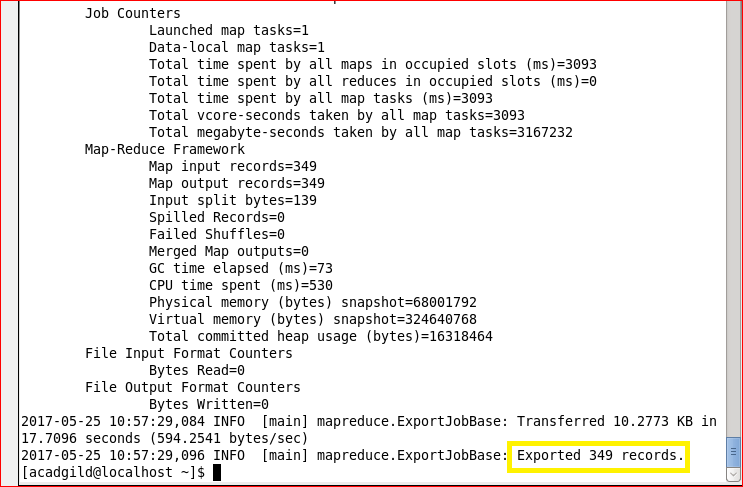


Creating table in MySQL:



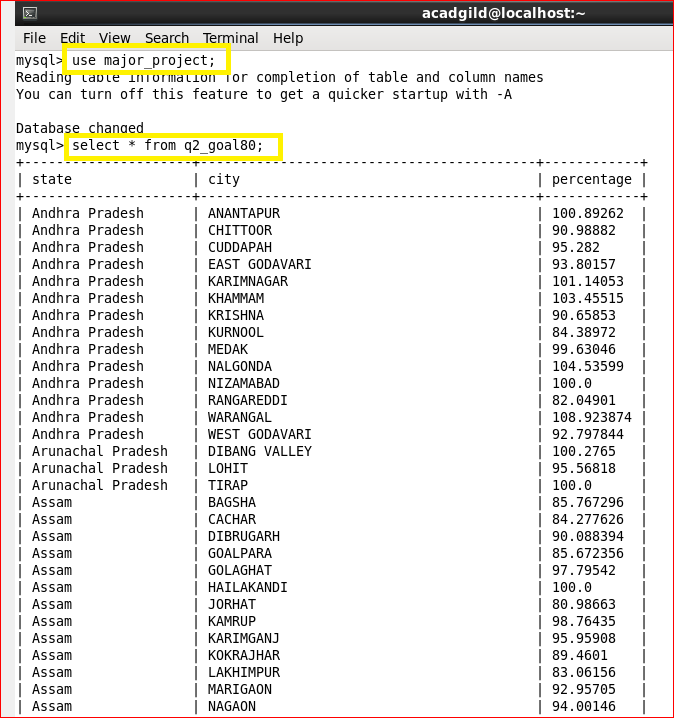
Exporting data from HDFS into MySQL using sqoop:

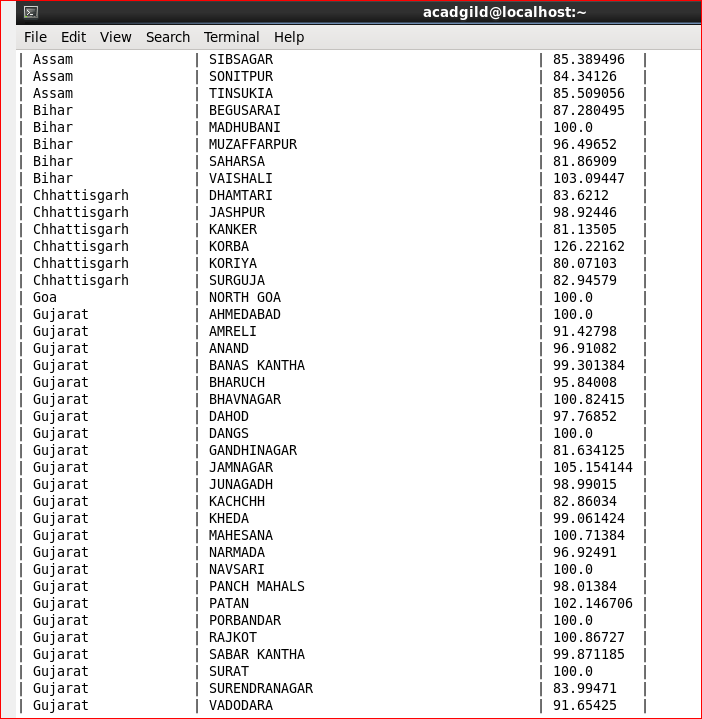


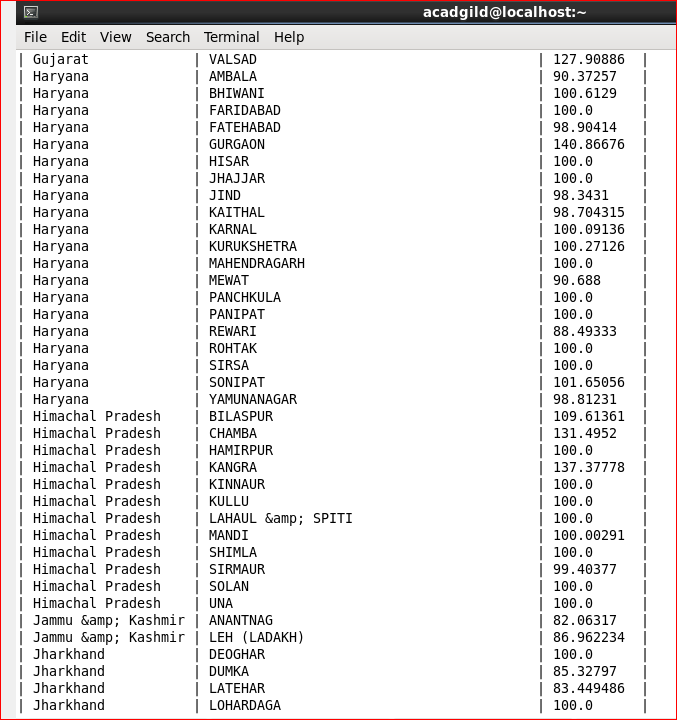


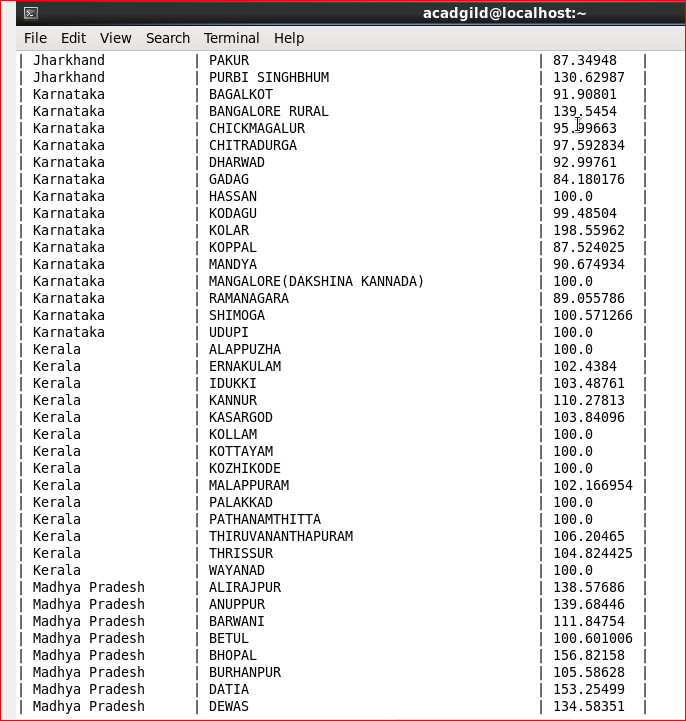
**Output:**

**Total 349 record.**

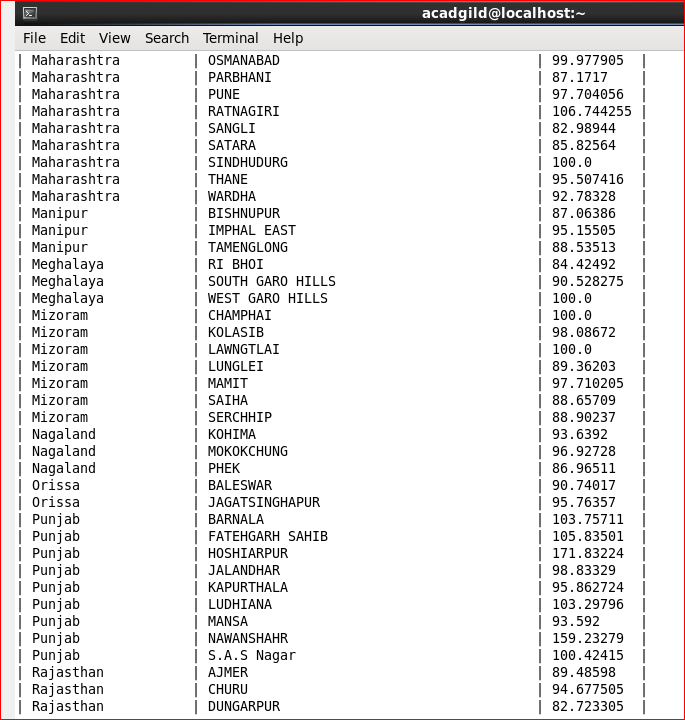


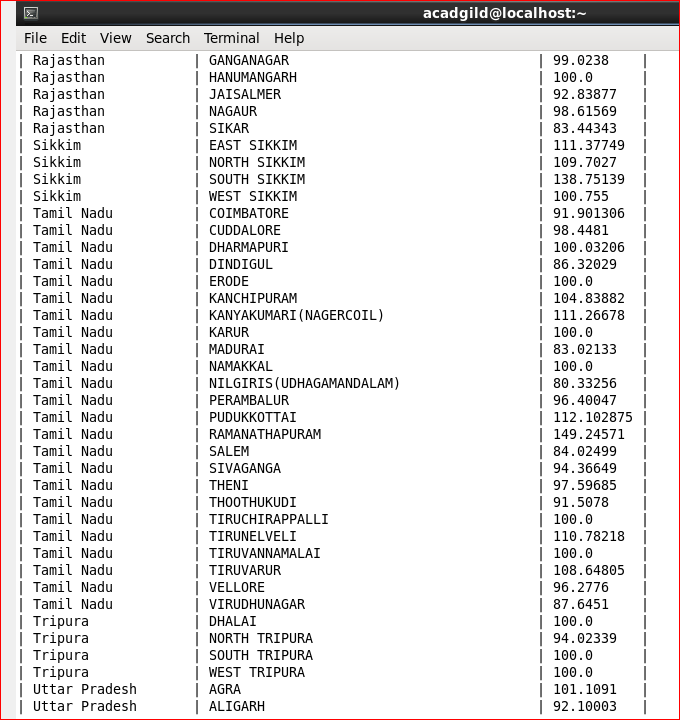


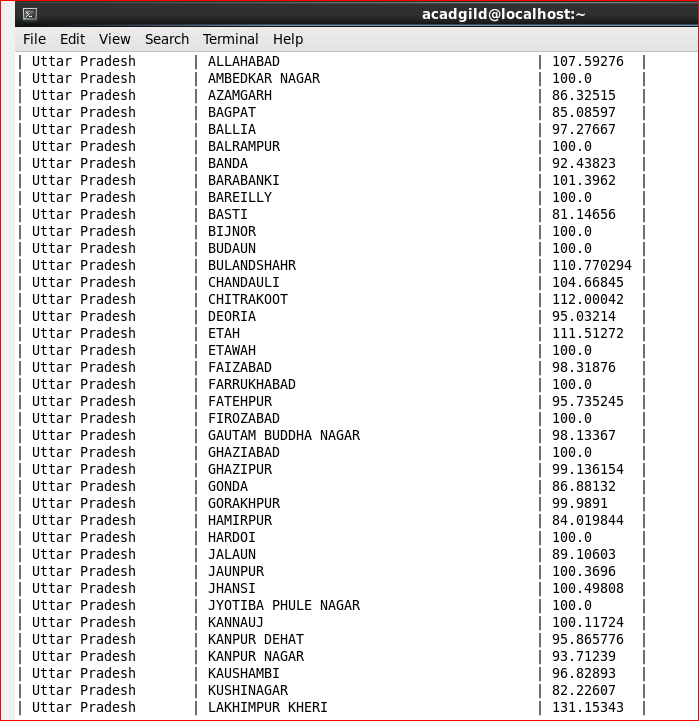




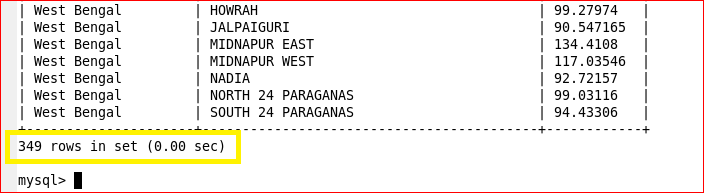






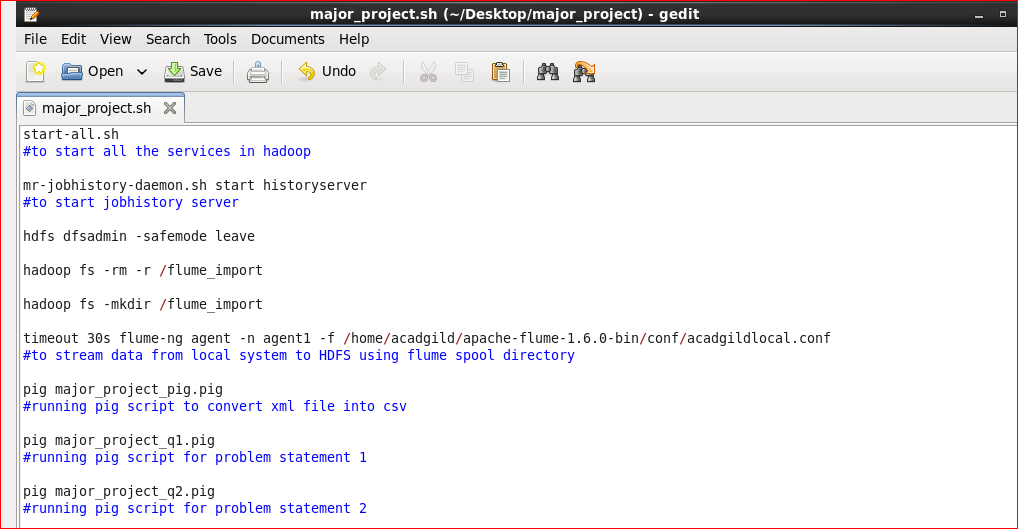






**Final solution:**

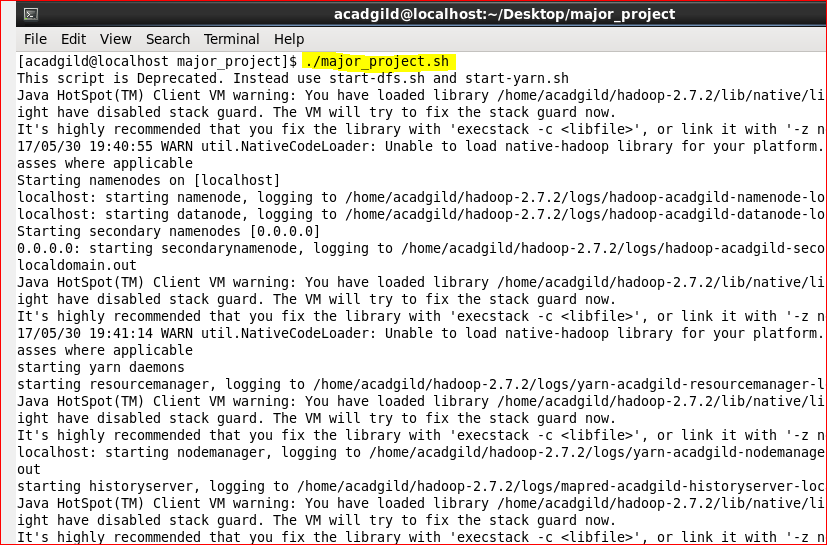
We can write shell script for whole project and this shell script can be run from command line by giving one simple command ./major\_project.sh







**Running shell script:**

****

The outputs after running this shell script are same as outputs attached above.