**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 Big Data and Hadoop Development

A C A D G I L D

Page 4 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 Big Data and Hadoop Development A C A D G I L D Page 5

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.

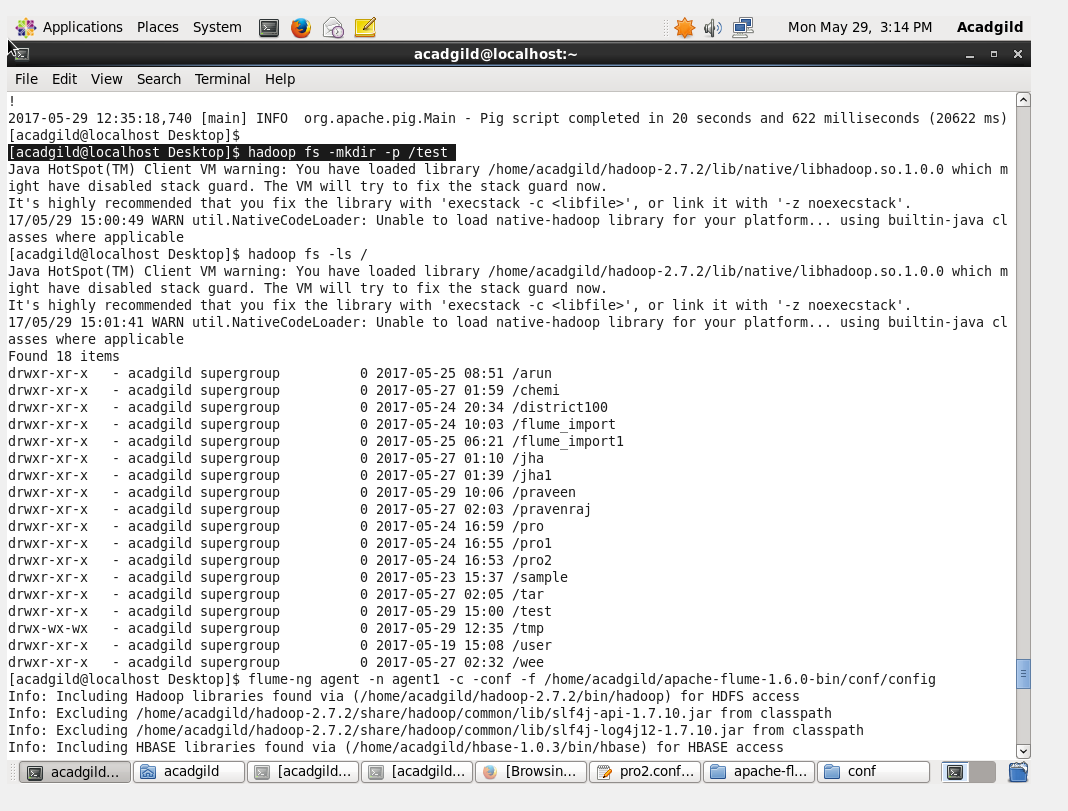
**Problem statement :**

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

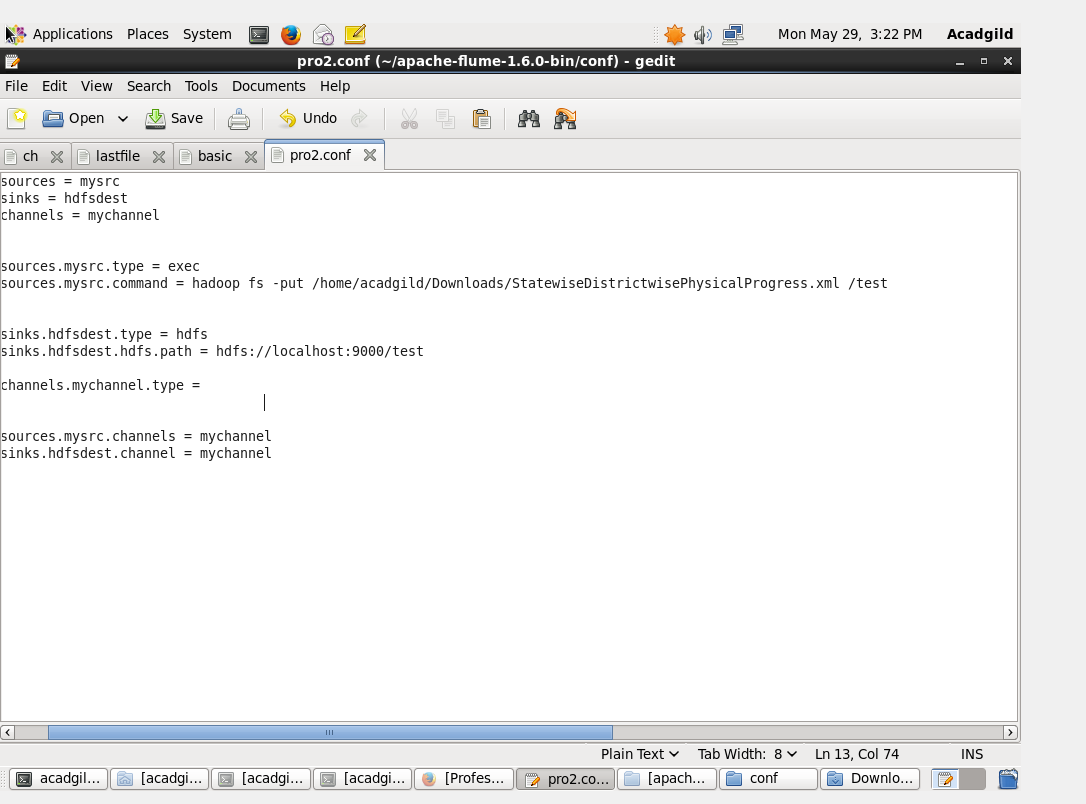
**2. Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards. Export the results to MySQL using Sqoop.**

**First, creating a directory named as flume\_import. In the below screenshot we can see that directory has been created.**

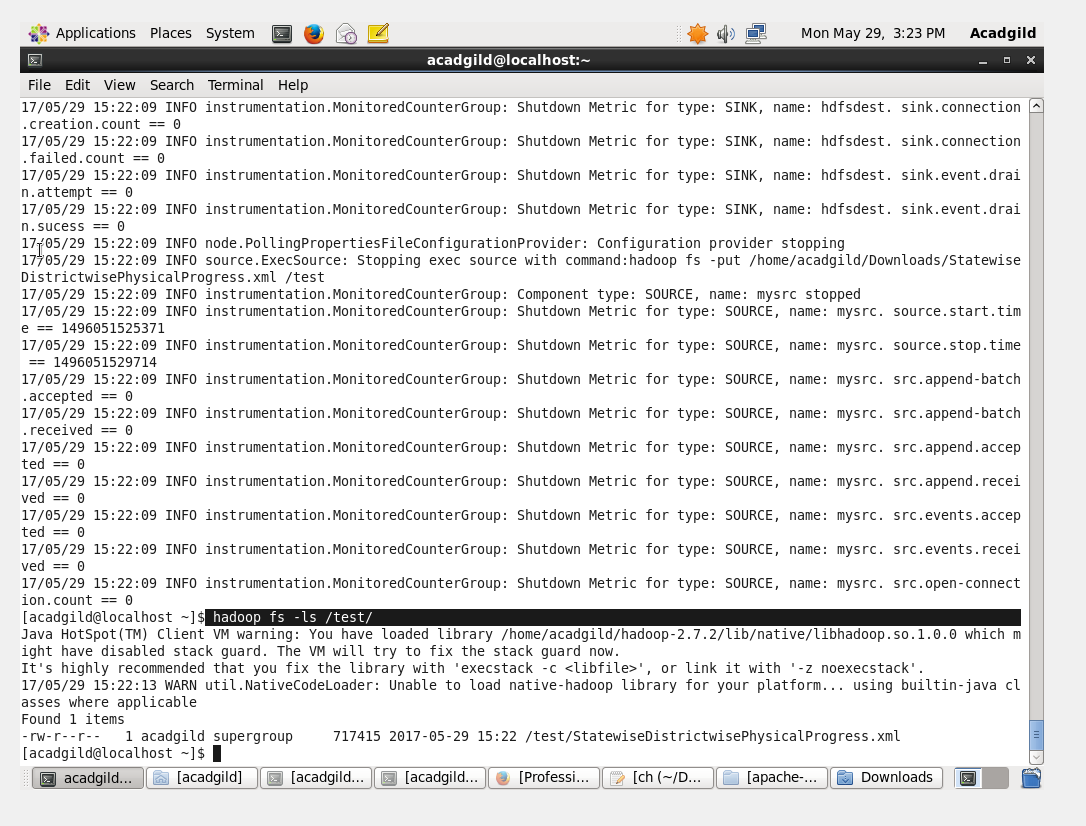
**Flum**

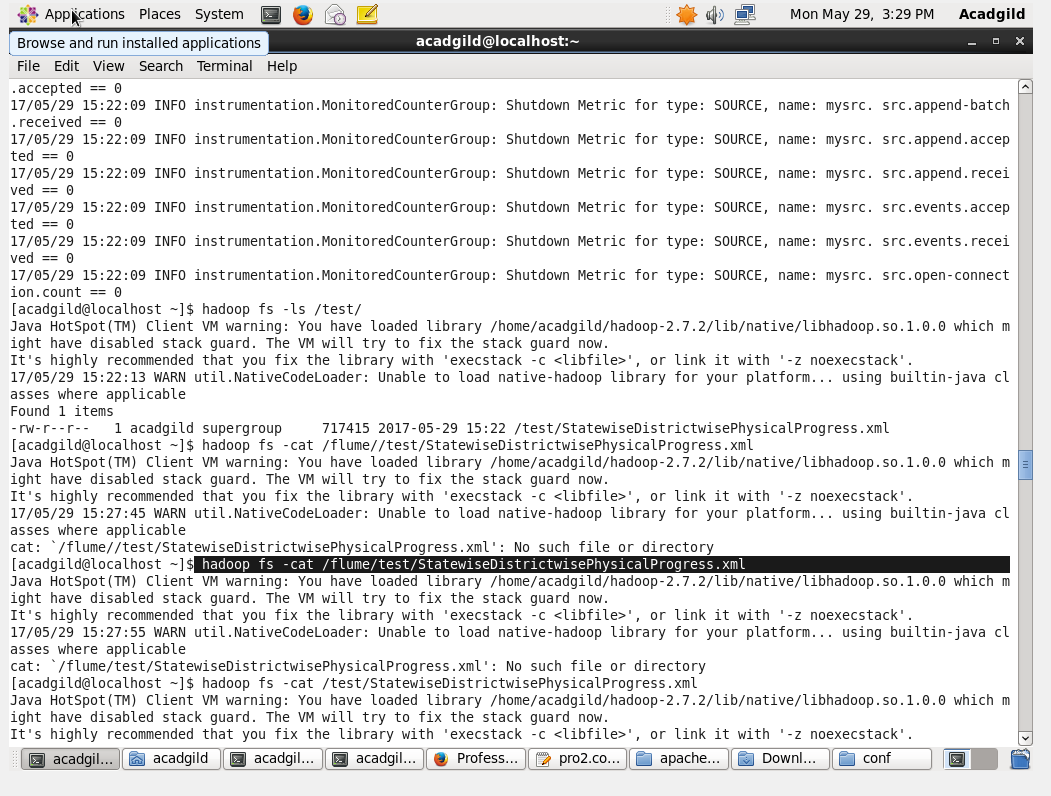


**Copying dataset from local file system to HDFS using flume by using the below conf file.**



**Listing the created directory and performing basic cat operation.**

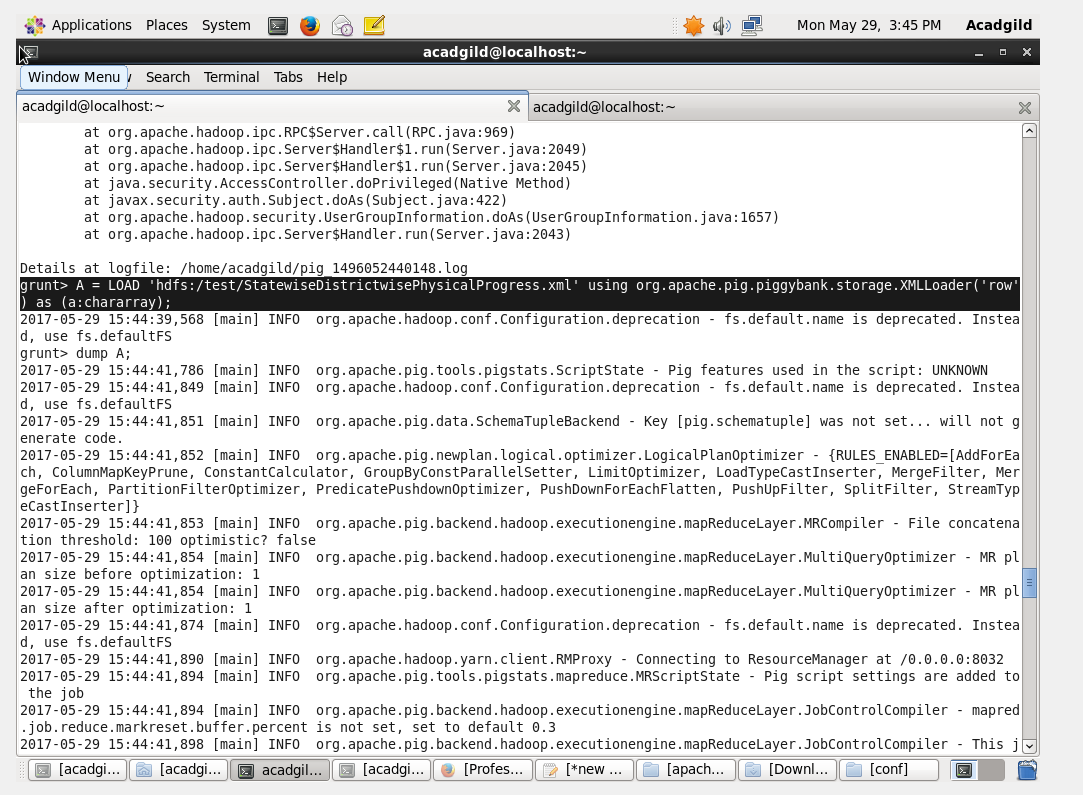






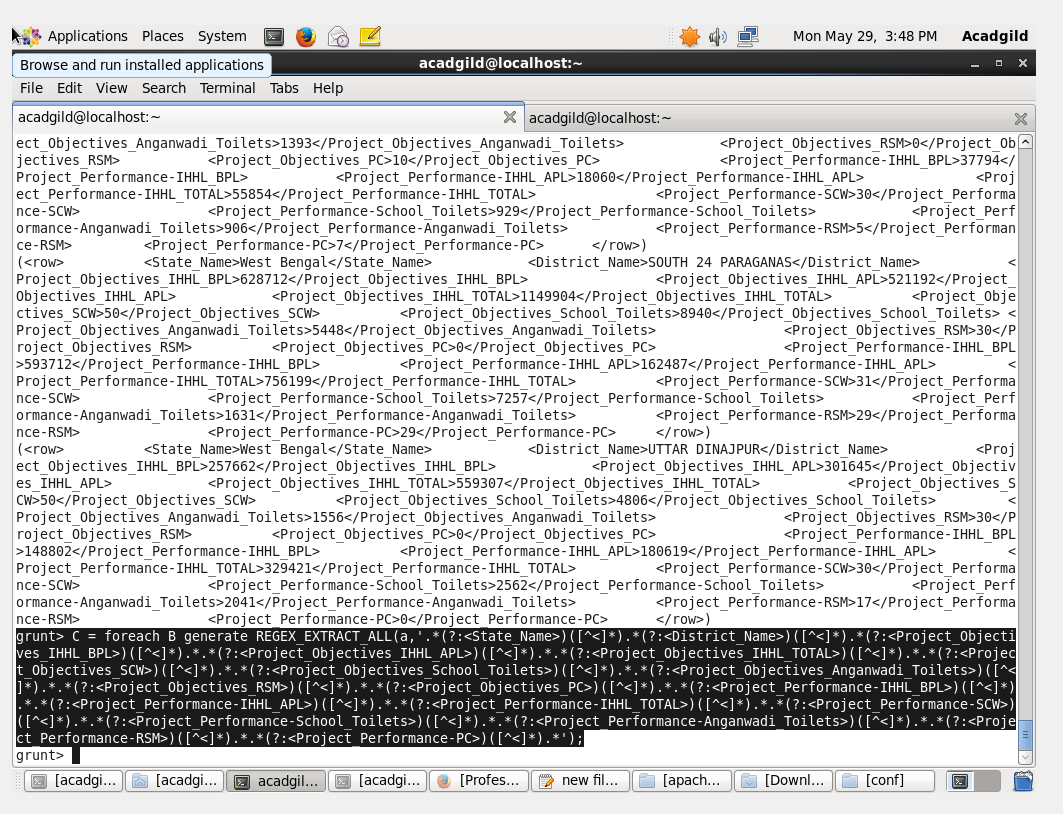
**Here data is loaded from /flume project location which contains a XML file and in order to parse the xml file so that Pig can understand it.**

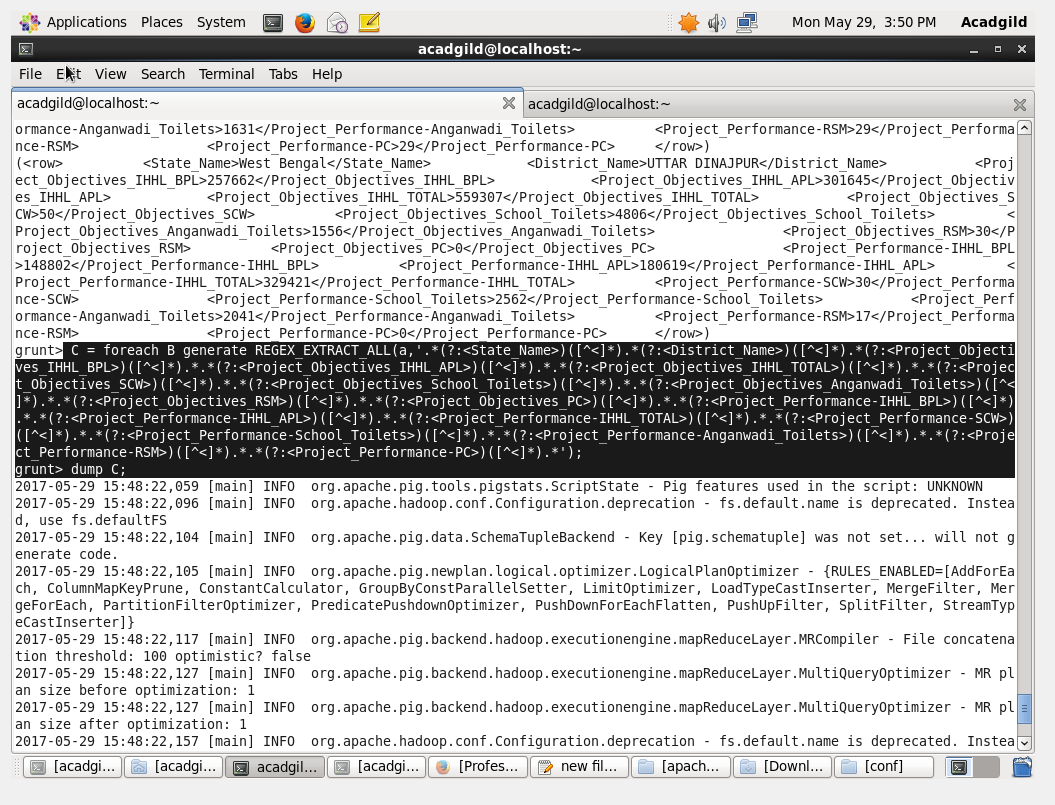
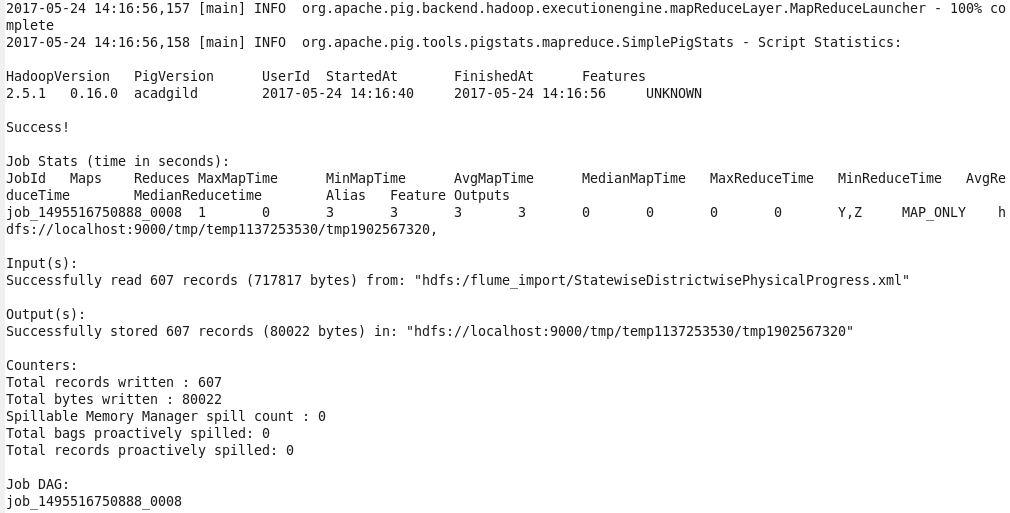
**XMLLoader() function which is present in PiggyBank of Apache is used.**



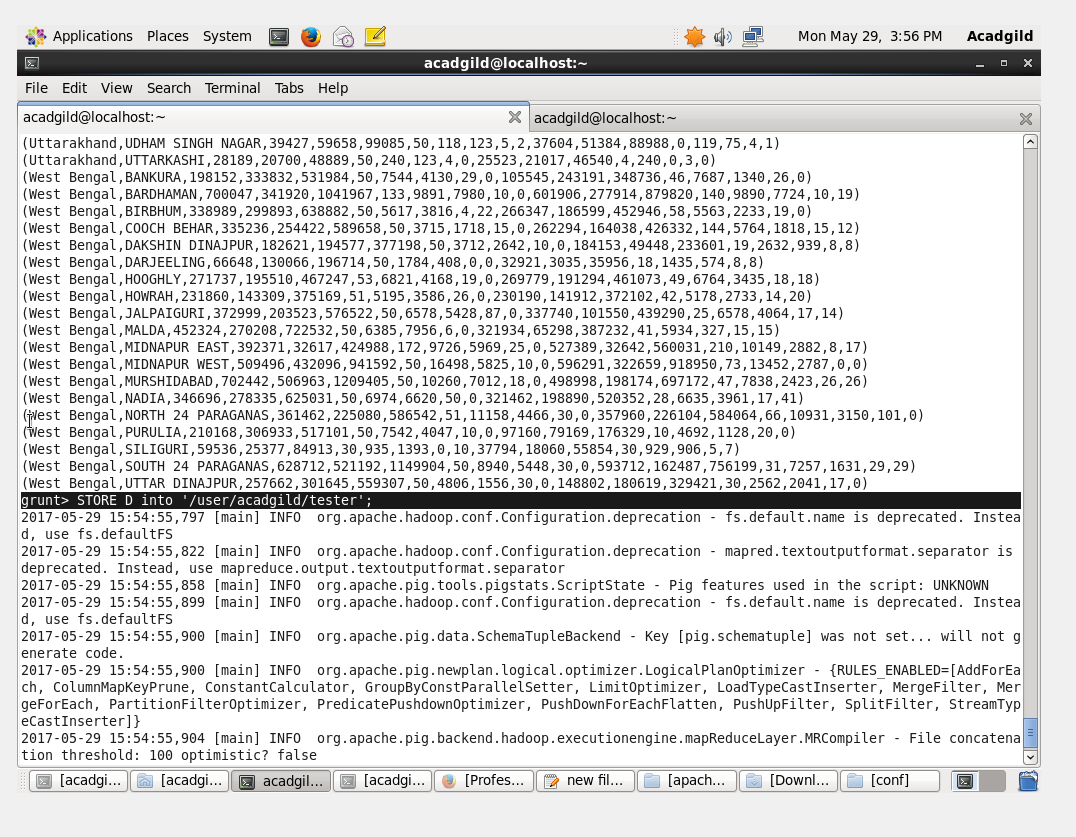


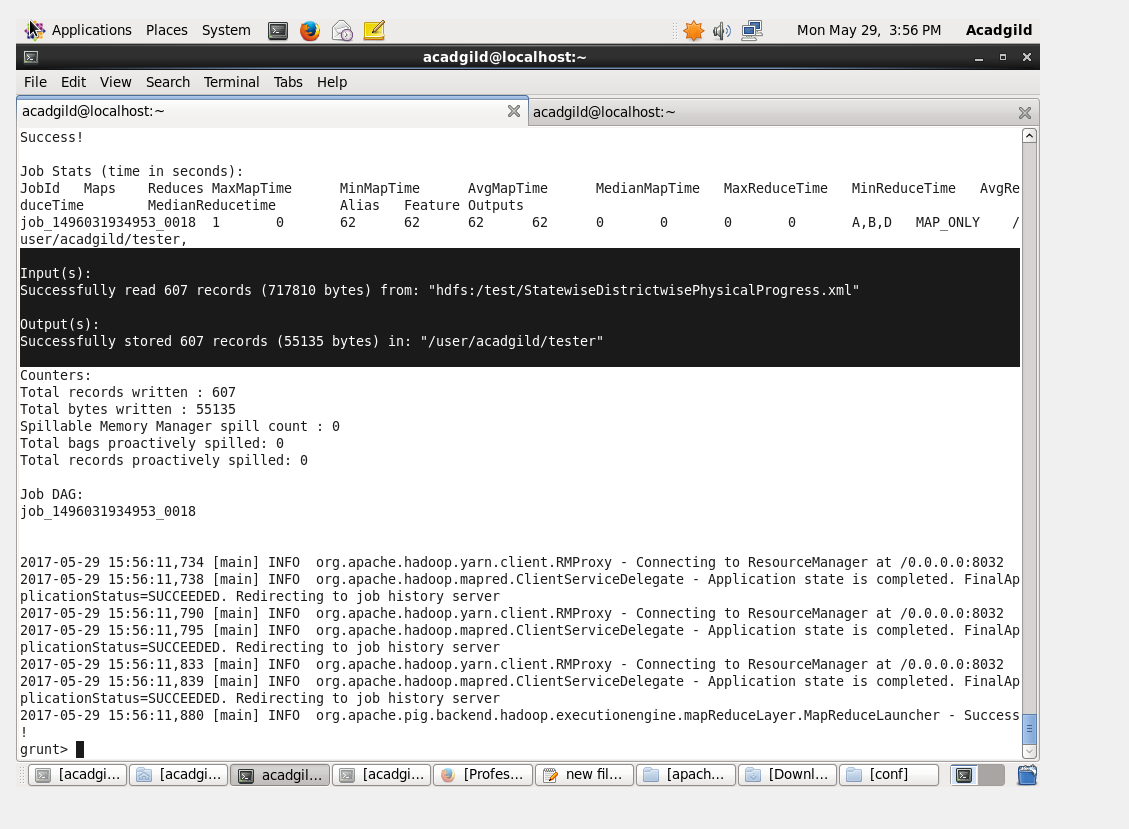
Using regex-function for capturing the values within tag and FLATTEN is used flattening the TUPLE.



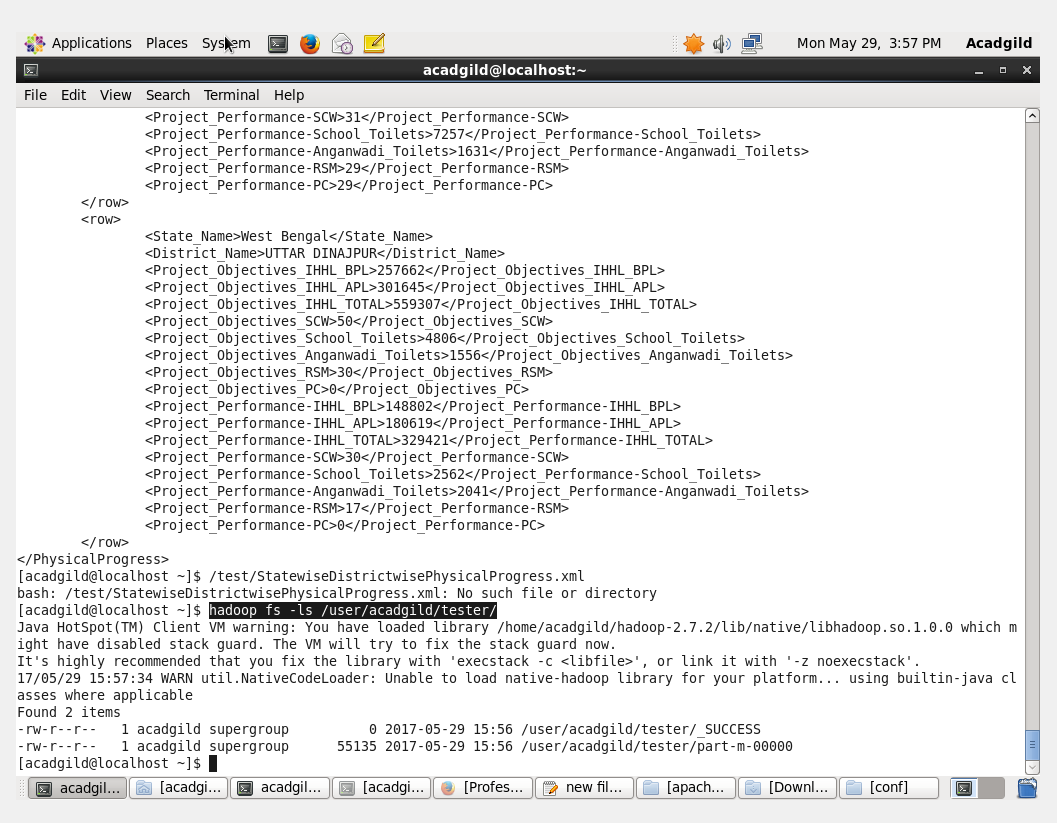


**Storing the above results into HDFS for further analysis.**

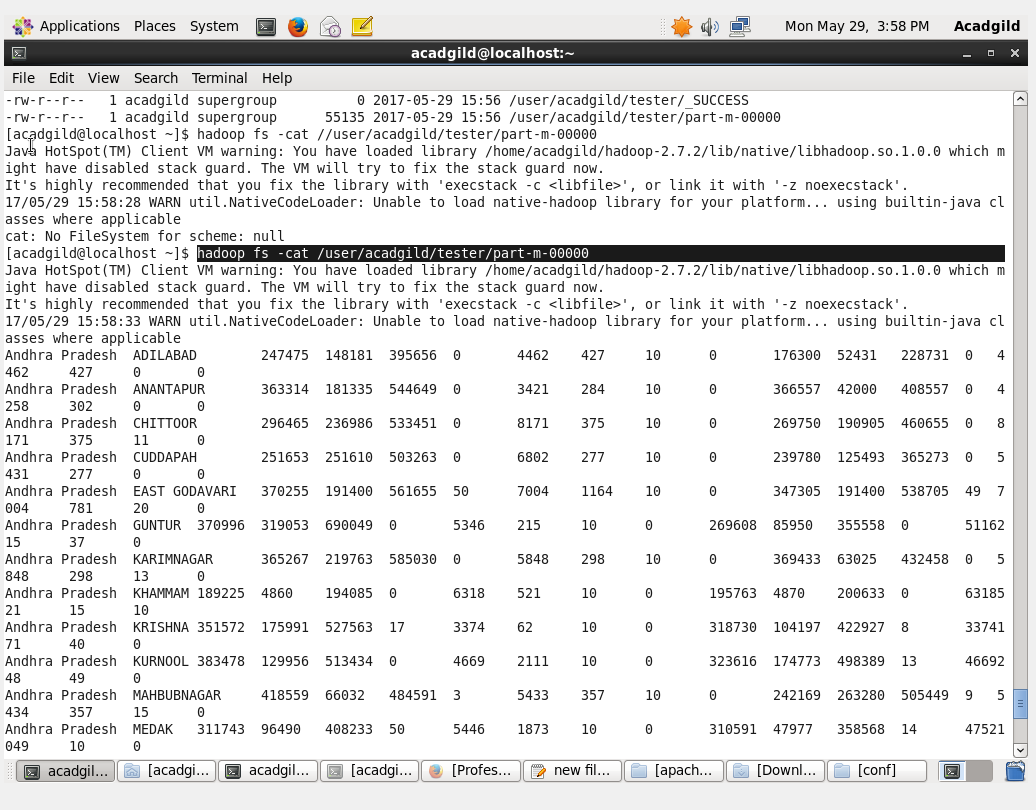


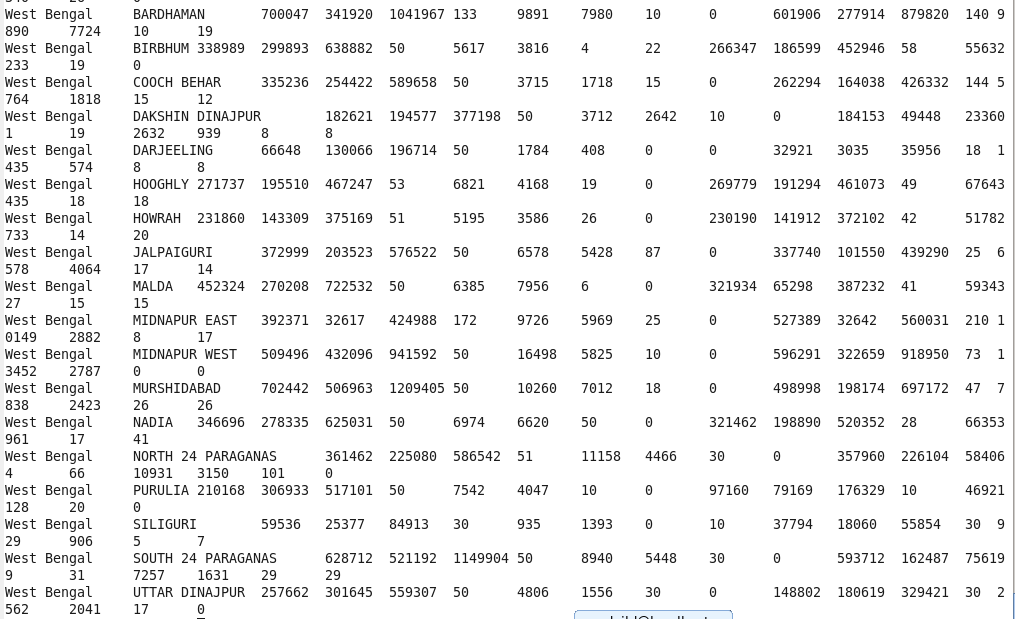


Here, we can see that data is successfully stored.



Using cat command we can see the stored data.





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

Step1: loading the dataset

Step2: filtering the loaded data so as to get the districts who achieved 100 percent objective in BPL cards.

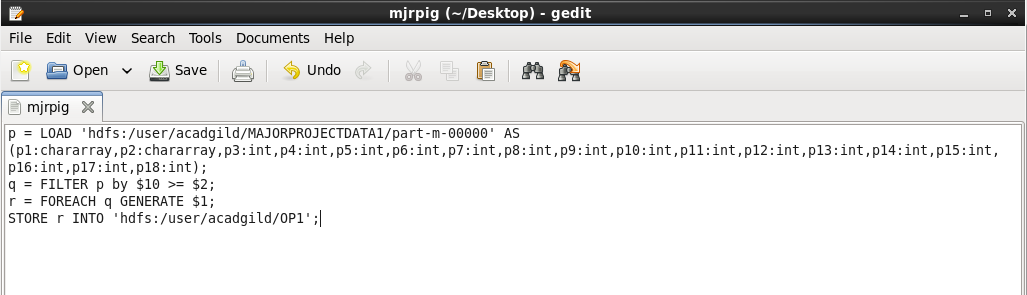
$10 is the project objective in BPL cards column

$2 is the project performance in BPL column

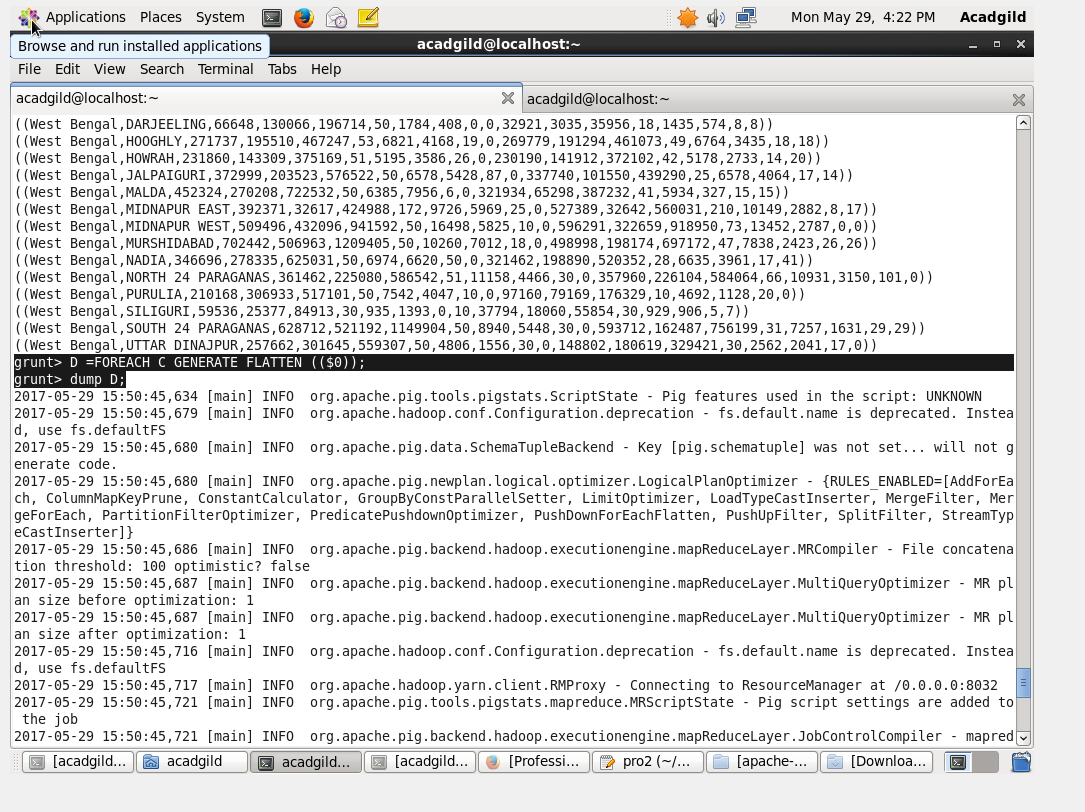
Comparing whether $10 i.e, project objectives in BPL have achieved 100 percent or not.

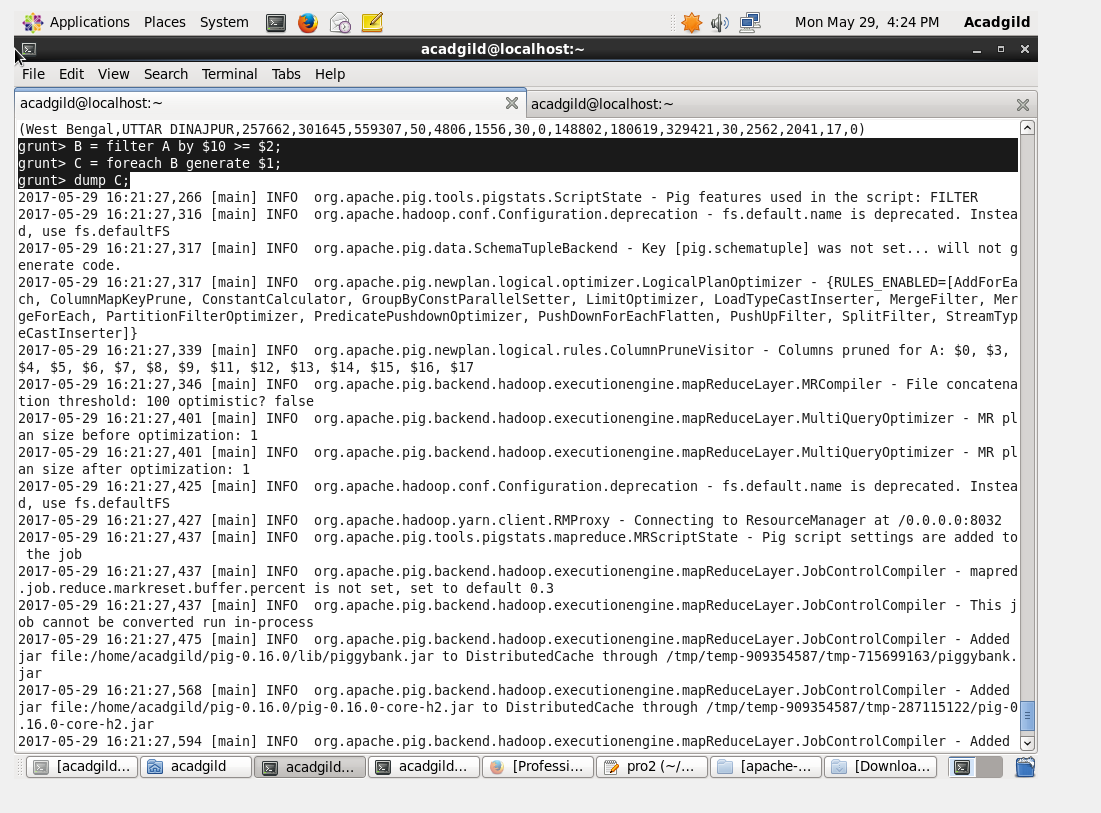
Step3: If achieved then generating $1 which is the district column

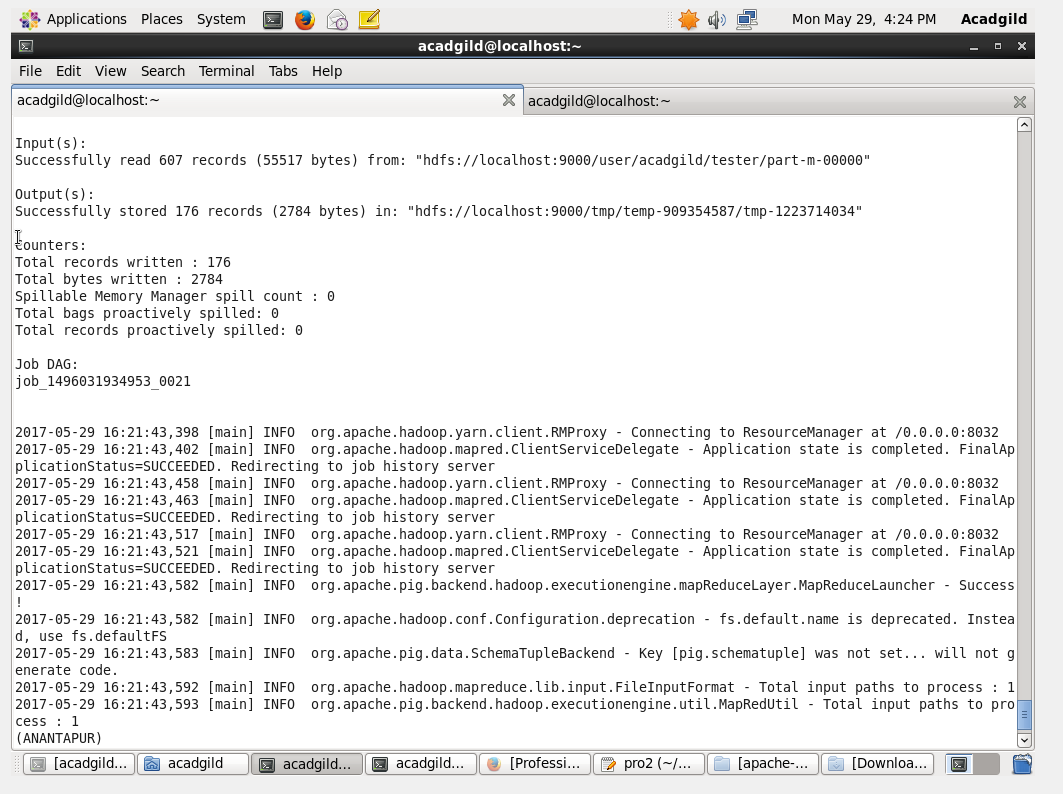
Step4: finally storing the results as OP1.

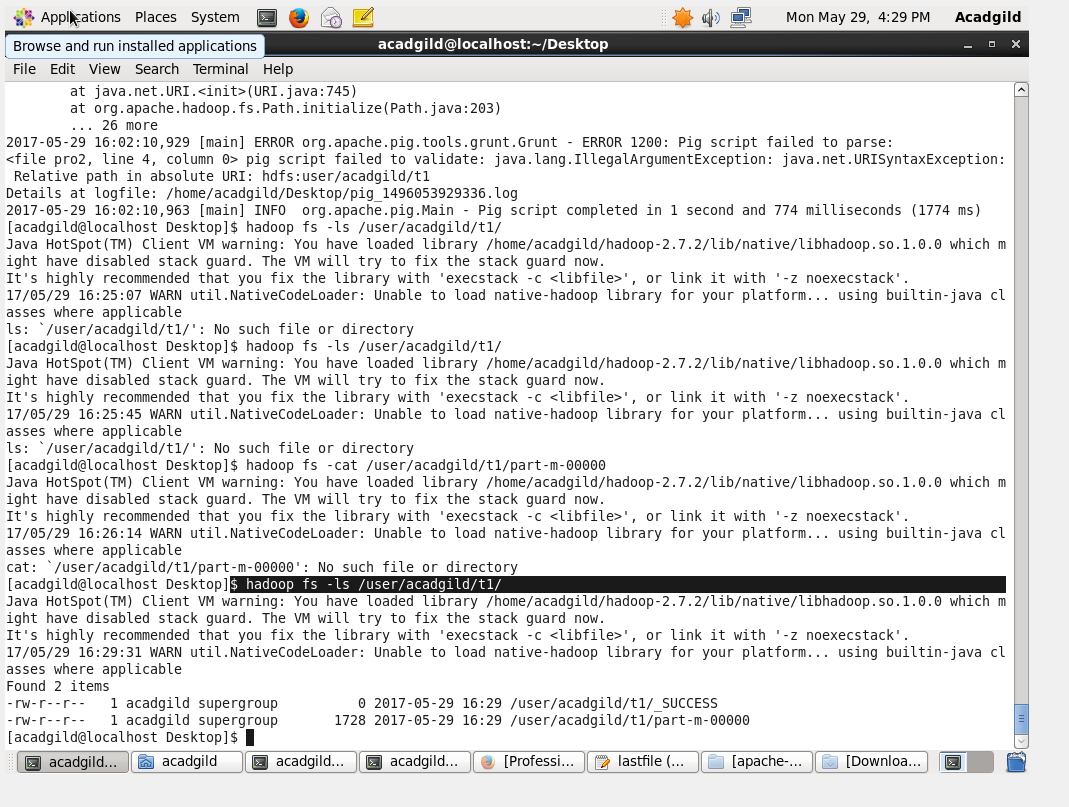


Running the above pigscript :



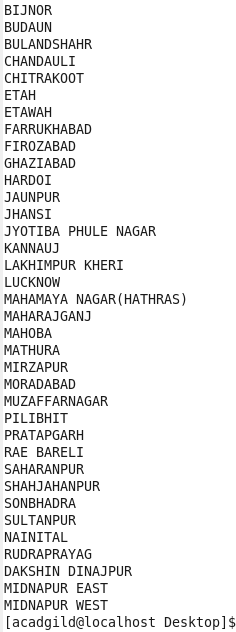
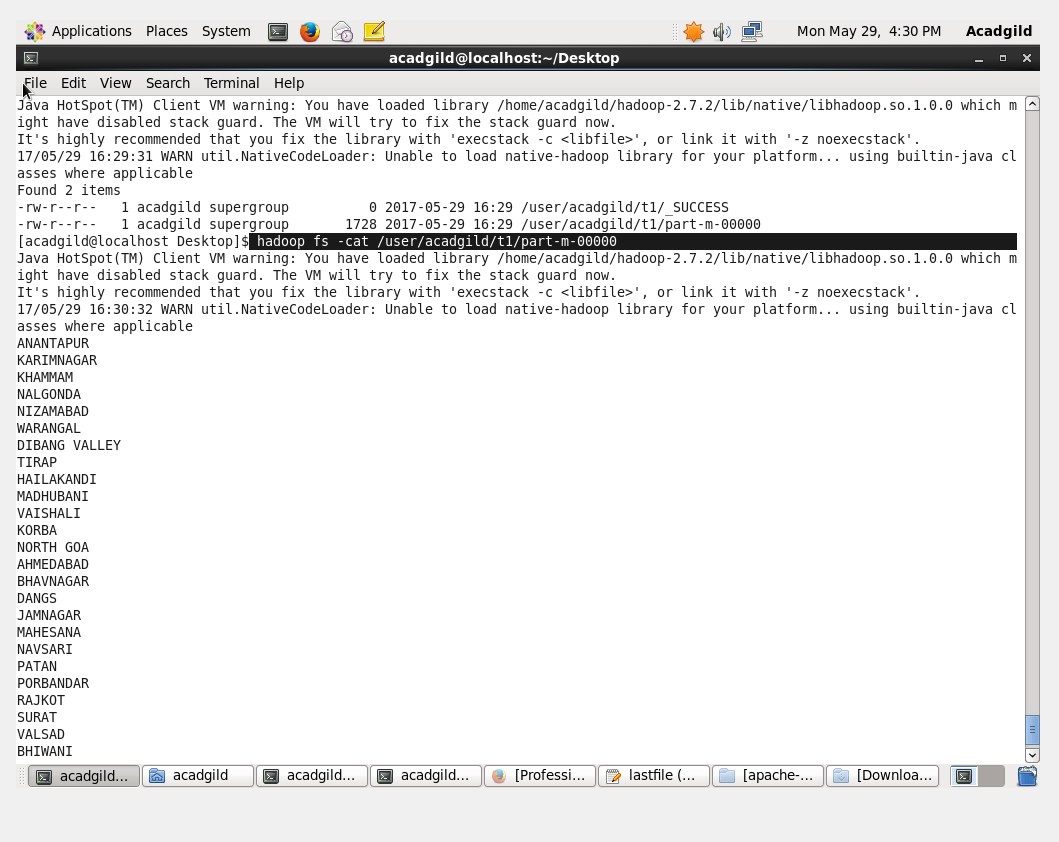




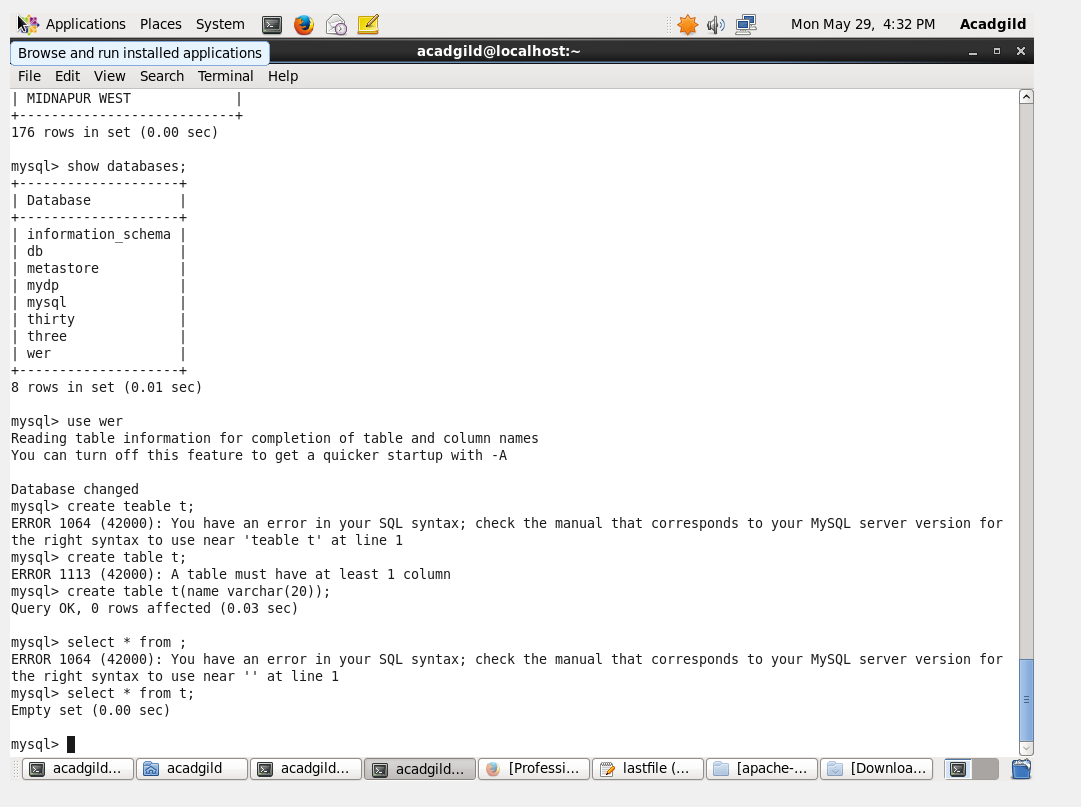


Here, we can see that job ran successfully.

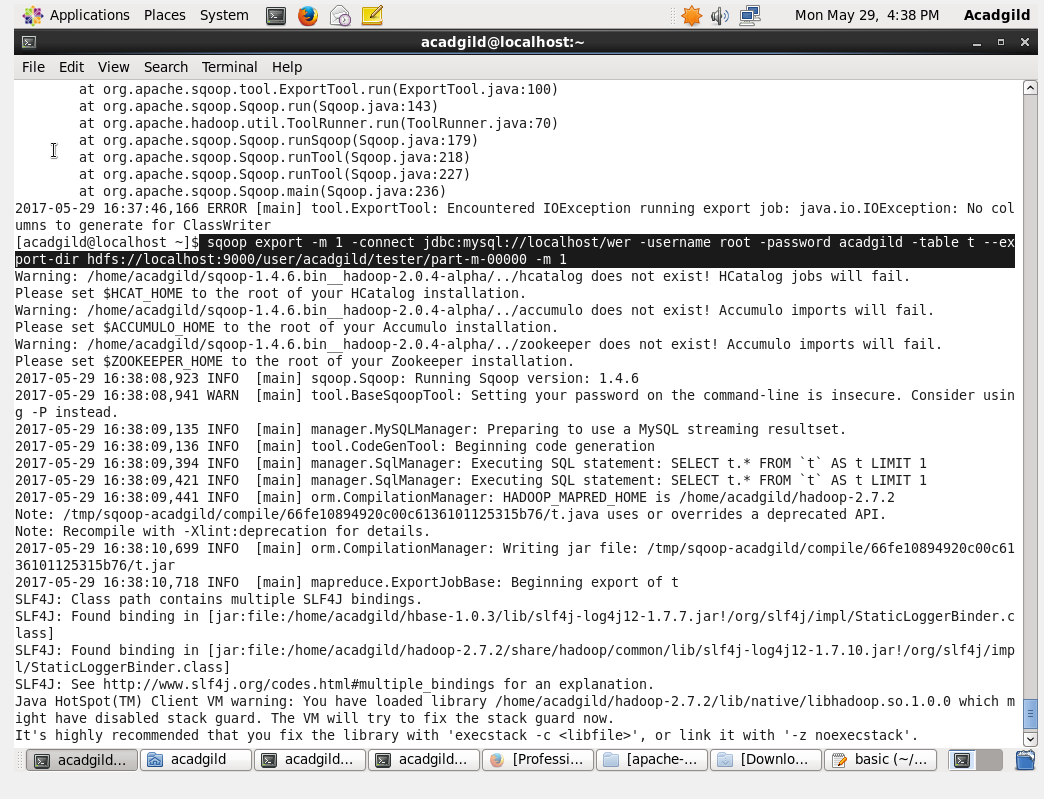
Then, using cat command displaying the results.

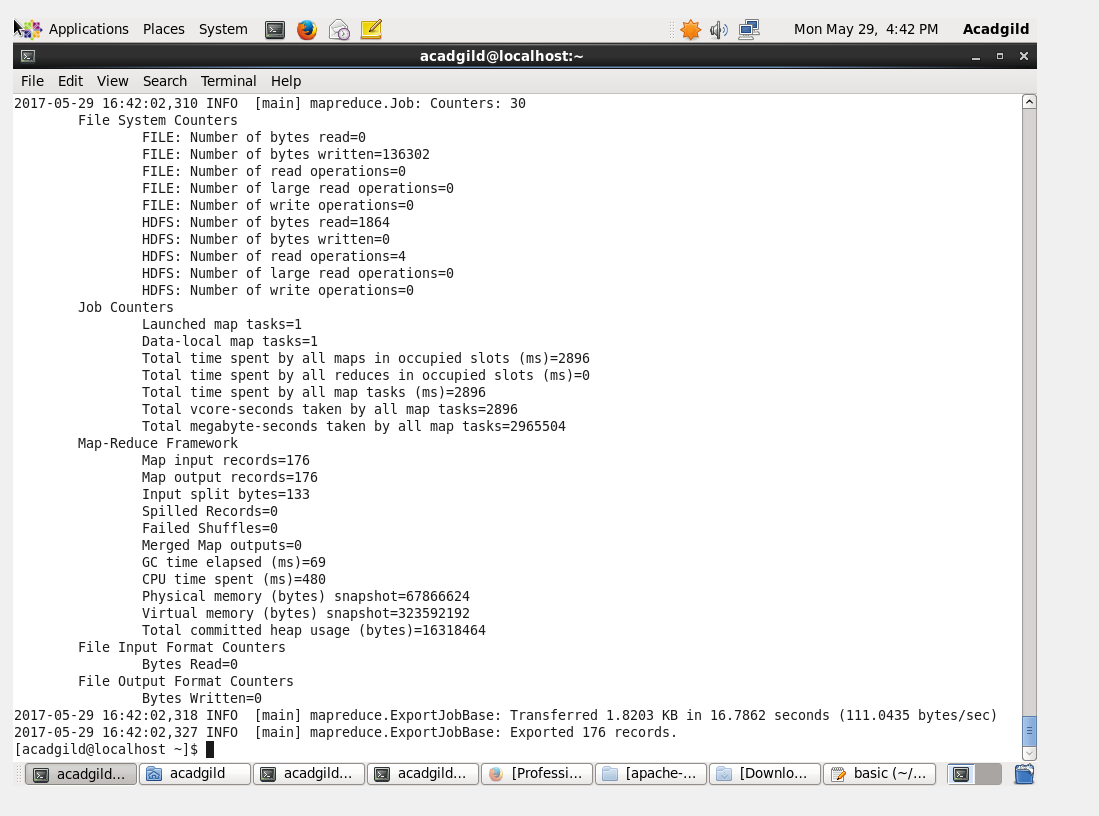


To export the result from pig using sqoop creating a table in mysql and inserting no values in the table so as to perform the export.



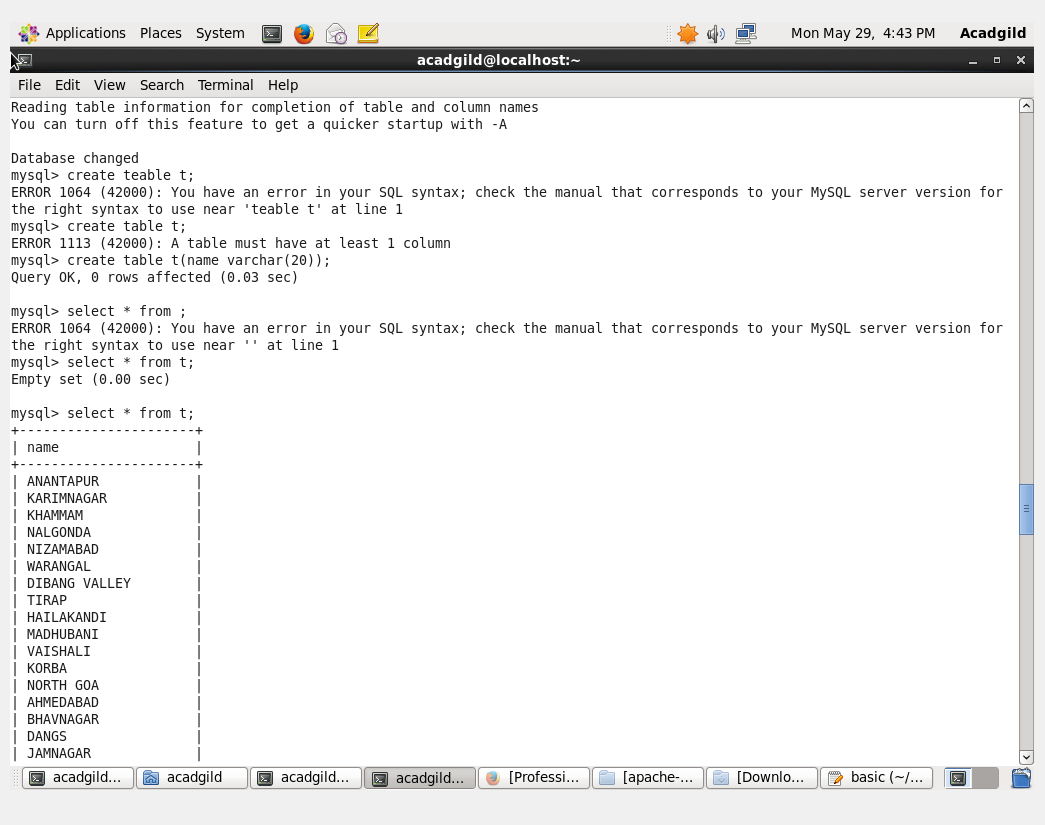
Below is the export command to export the records from pig to mysql

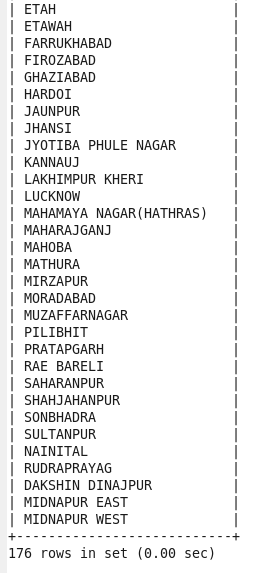




Here, we can see that the data is successfully exported.

Now, if we perform the select operation on the empty created table it will show the exported data.

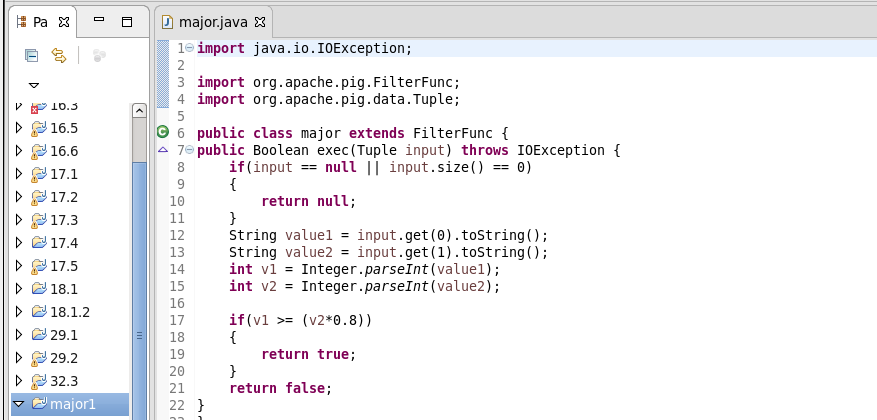




1. **Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards. Export the results to MySQL using Sqoop.**

**Creating UDF:**

Adding the necessary Imports



Firstly, registering the jar

Then, loading the data and the datatype and name of fields are defined.

**Filtering Districts that acheived 80%:**

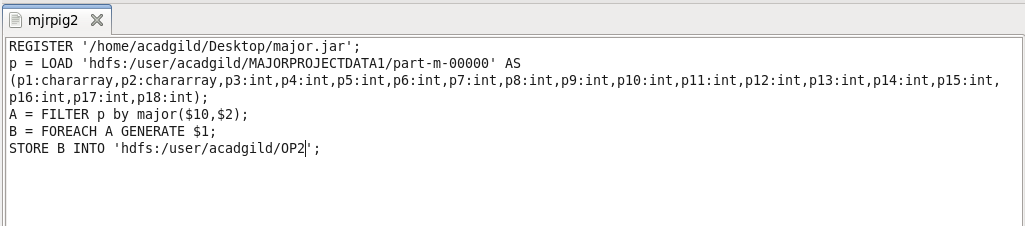
Used a udf function “major” which filter the districts which have acheived less than 80%

**Generating District List:**

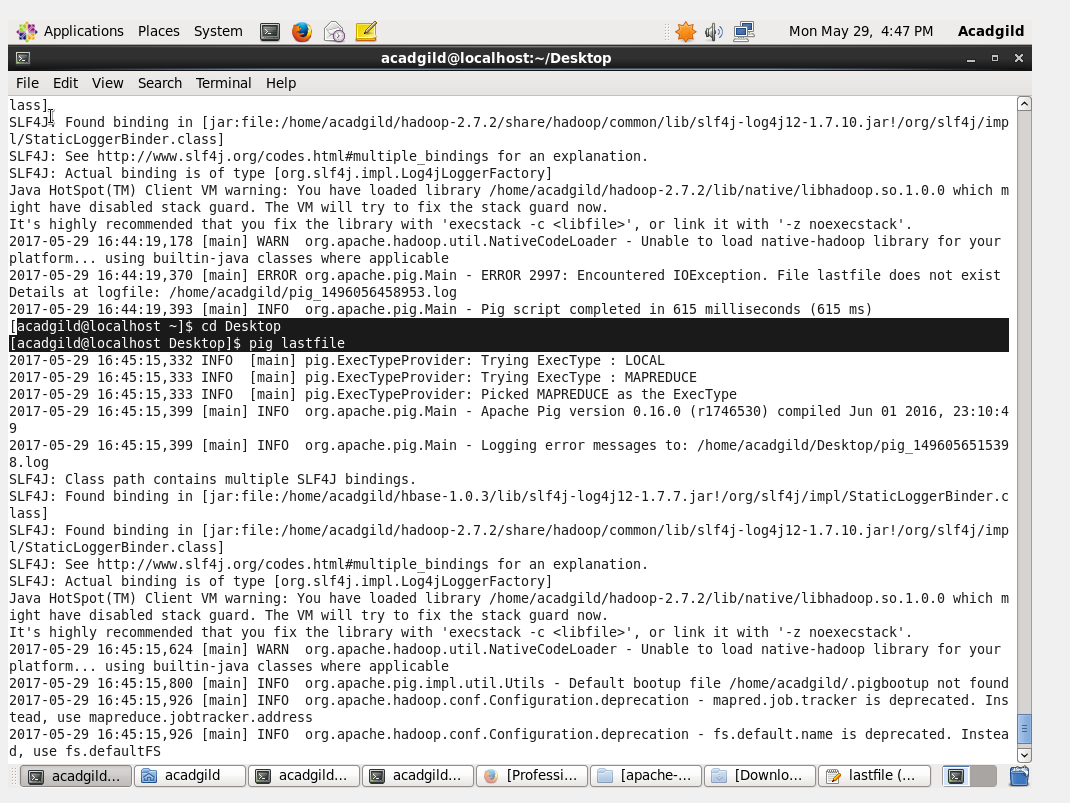
$1 is the district column. Here we are generating $1 so as to get the districts name which have reached 80% of objectives of BPL cards.

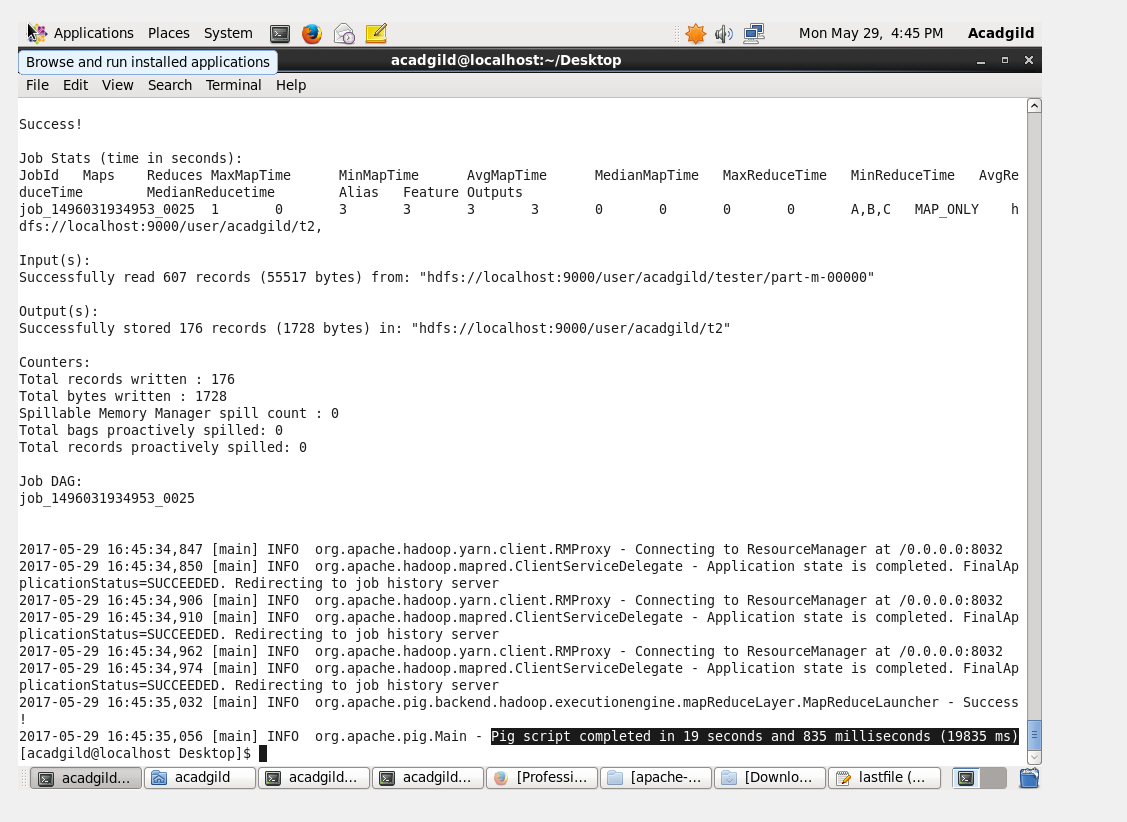
**Finally, Storing the result**

Data is stored into “OP2” in Hdfs from which it will be exported to mysql.

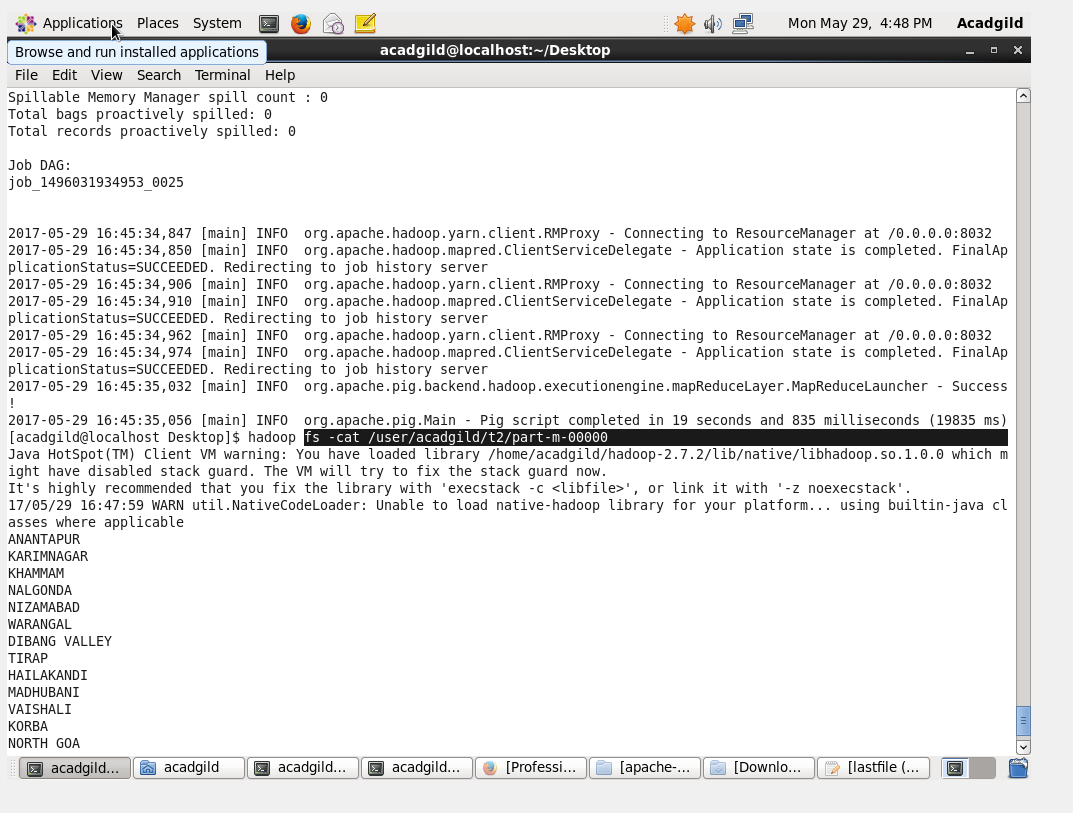


**Running above pigscript:**

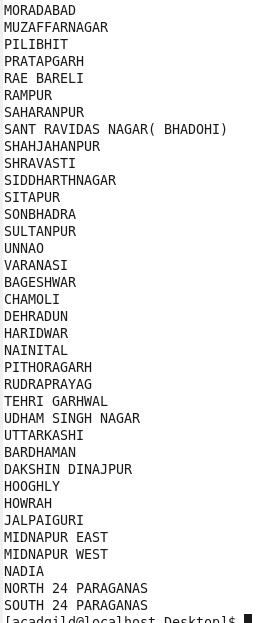




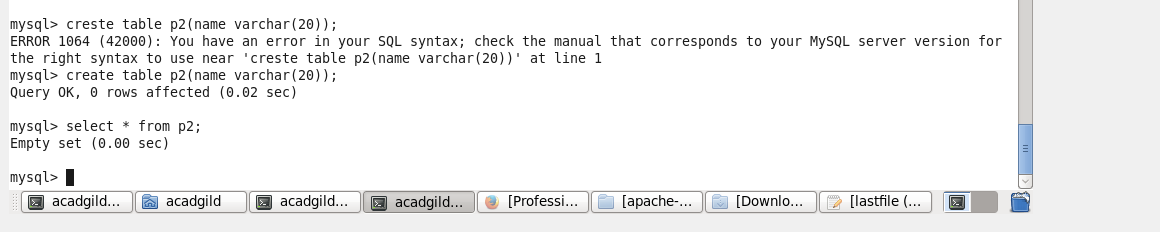
**Listing it and then performing cat operation on it.**



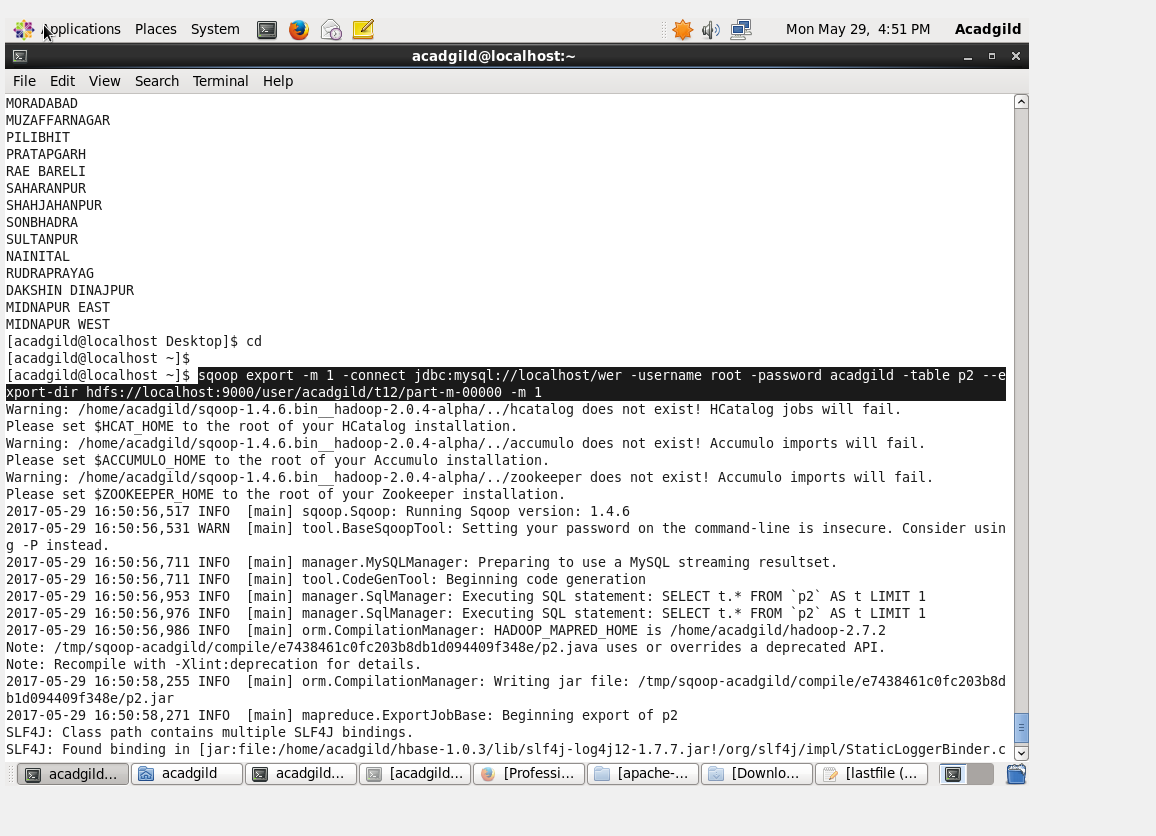
**Cat command will display the stored output of the pig commands**



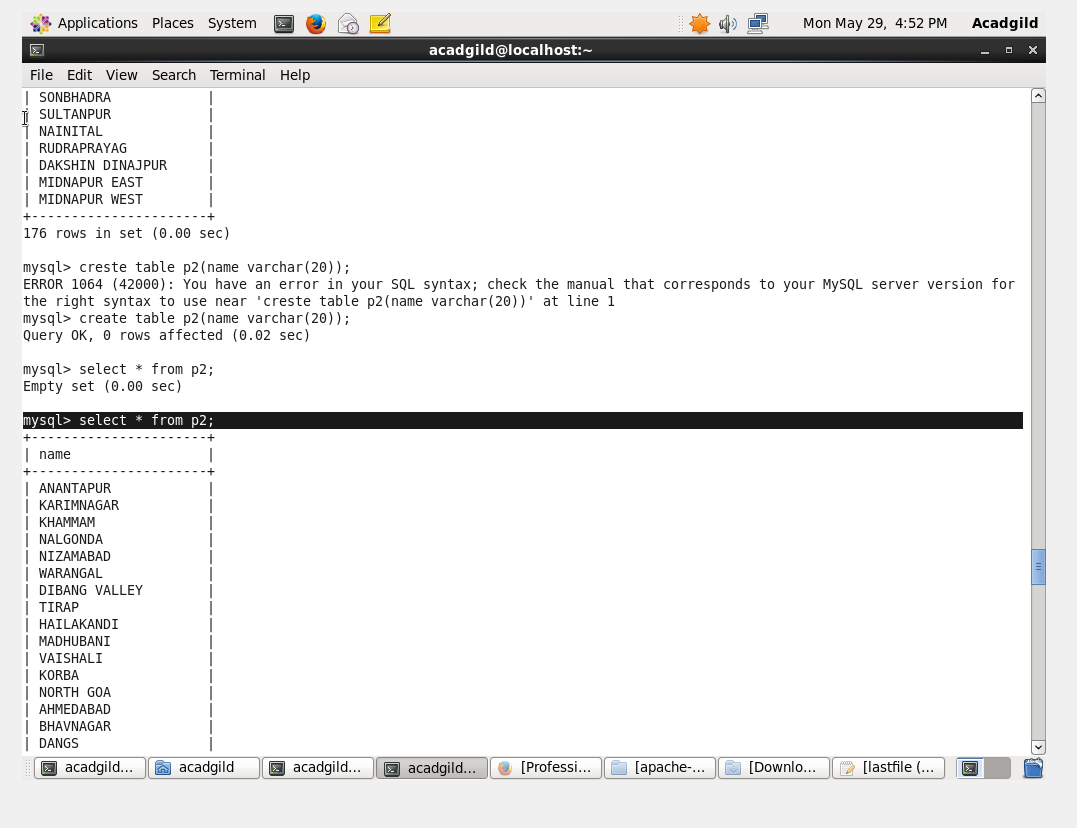
**To export the result from pig to mysql using sqoop creating a table in mysql and inserting no values in the table so as to perform the export.**

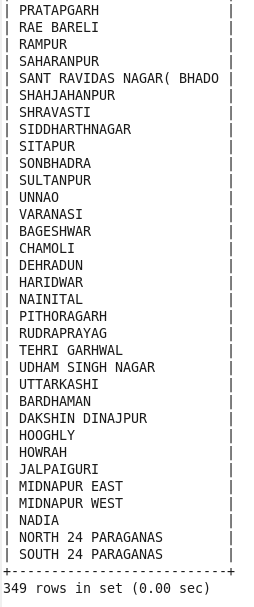


**Below is the export command to perform export operation**



**Using select command displaying the results.**





**I have not attached the full output. Only the first and last part of the outputs are attached.**