**State-Wise Development Analysis In India**

**Problem Statement:**

* 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
* Data enrichment, validation should be done on XML data using Spark

**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
* Priority 3 - Data enrichment, validation should be done on XML data using Spark

**Data enrichment,Validation: (Should be implemented in SPARK)**

Develop Spark programs to handle the following:

1. If any of Project\_Objectives\_IHHL\_APL or Project\_Objectives\_IHHL\_APL is NULL or absent, consider it as ‘0’
2. If field Project\_Objectives\_IHHL\_TOTAL from input xml is NULL or absent, consult the lookup file for field to get the values of Project\_Objectives\_IHHL\_TOTAL based on District and State name.
3. If corresponding lookup entry for Project\_Objectives\_IHHL\_TOTAL is not found, consider that record to be invalid.

**Data analysis(Using Spark):**

Read the xml file StatewiseDistrictwisePhysicalProgress.xml and perform following analysis using SPARK.

1. Display the districts with percentage group <50 , >=50 & <=80 ,>80
2. Display the districts where percentage is within 50 to 80
3. Display the count of districts in each percentage group

**Data analysis(Using Pig/MR):**

Develop MapReduce/pig job to preprocess

1. Find out the districts who achieved 100 percent objective in BPL cards

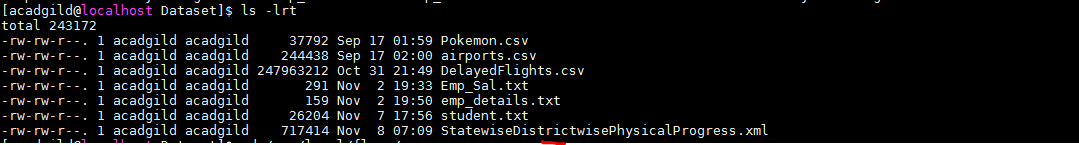
Export the results to mysql using sqoop

2. Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards.

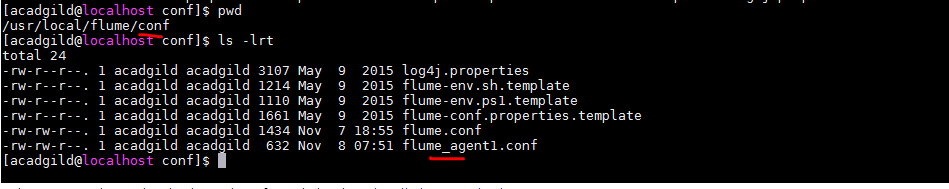
Export the results to MySQL using Sqoop.

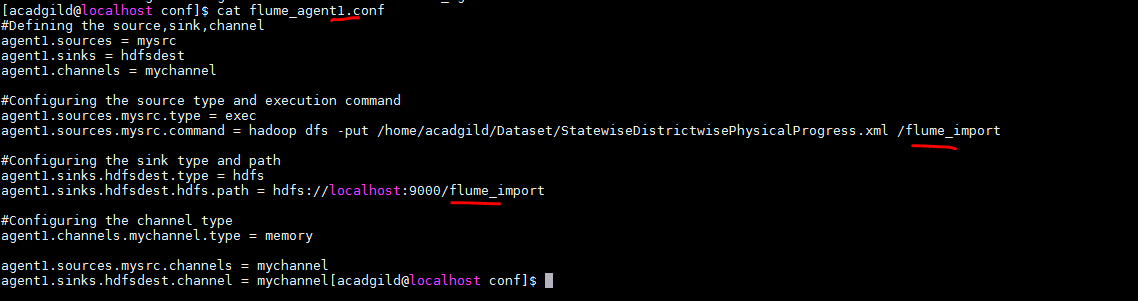
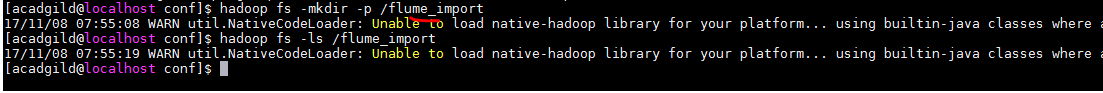
**Flume job:**

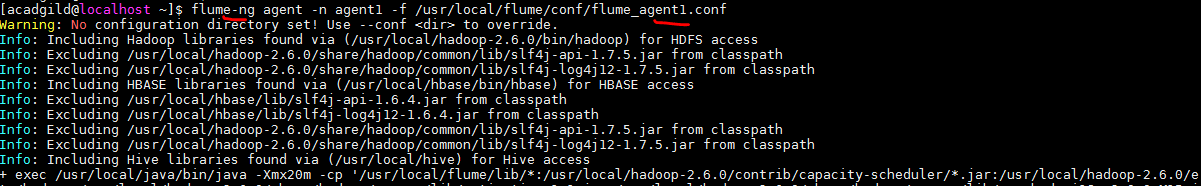
**Input Dataset in local:**

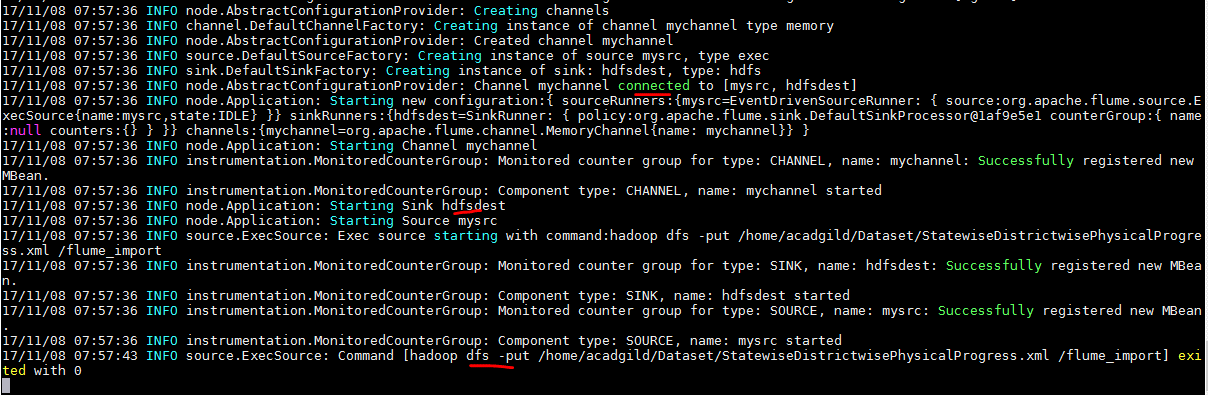
Input is in local and needs to be transferred to HDFS.

**Conf file – for flume agent:**

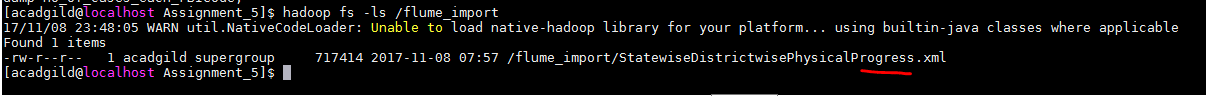
Below conf file is to define the properties of source(exec type),sink(target),channel.

Created a new directory in HDFS as a sink for flume(from where we can use the data for further analysis)

**Starting the flume agent – agent1:**

**Flume log:**

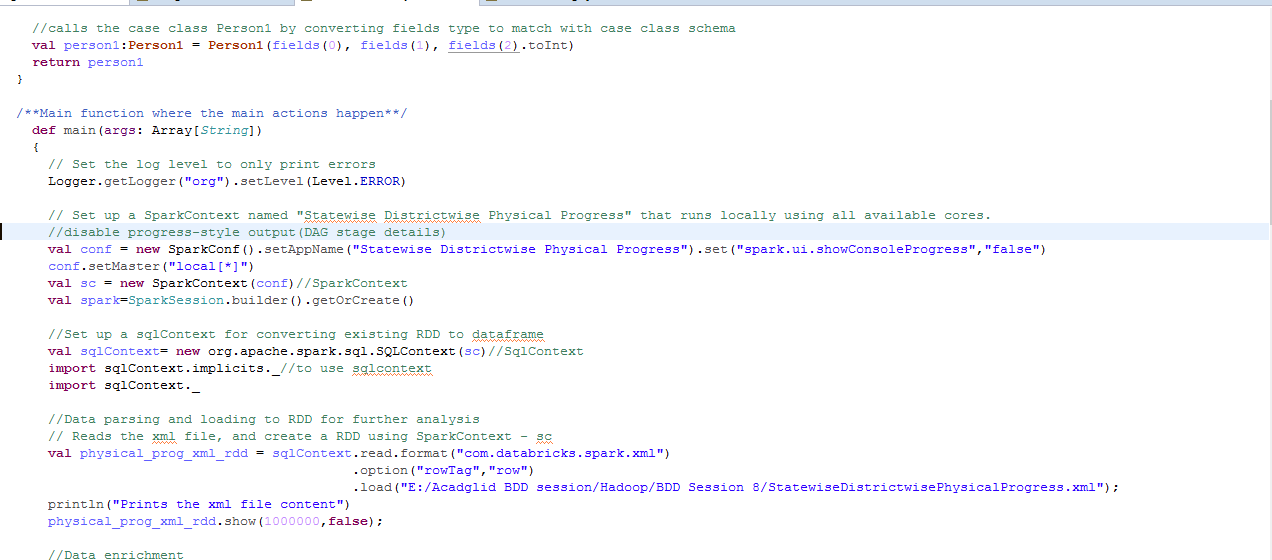
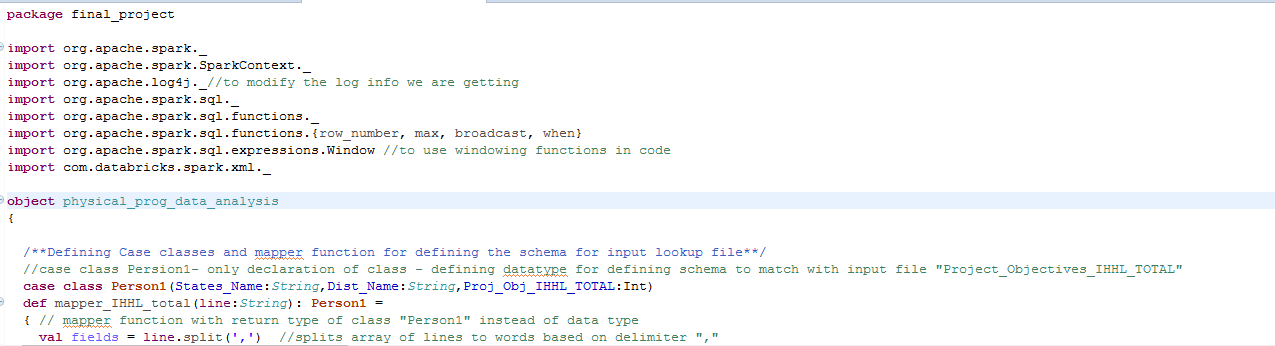
**HDFS sink after flume import:**

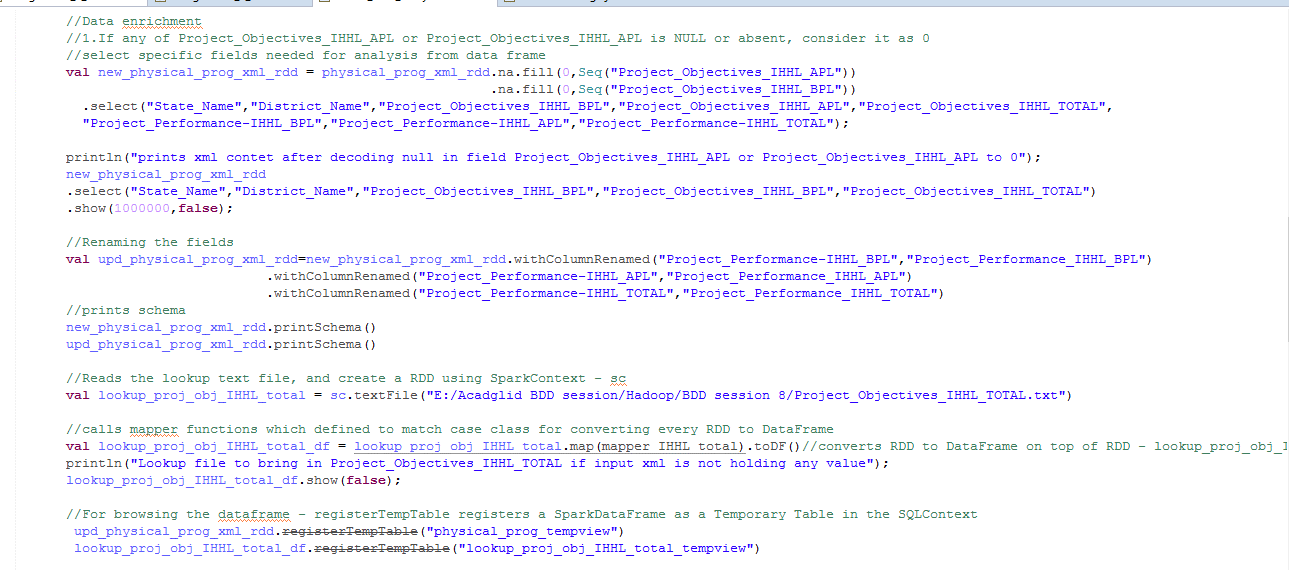
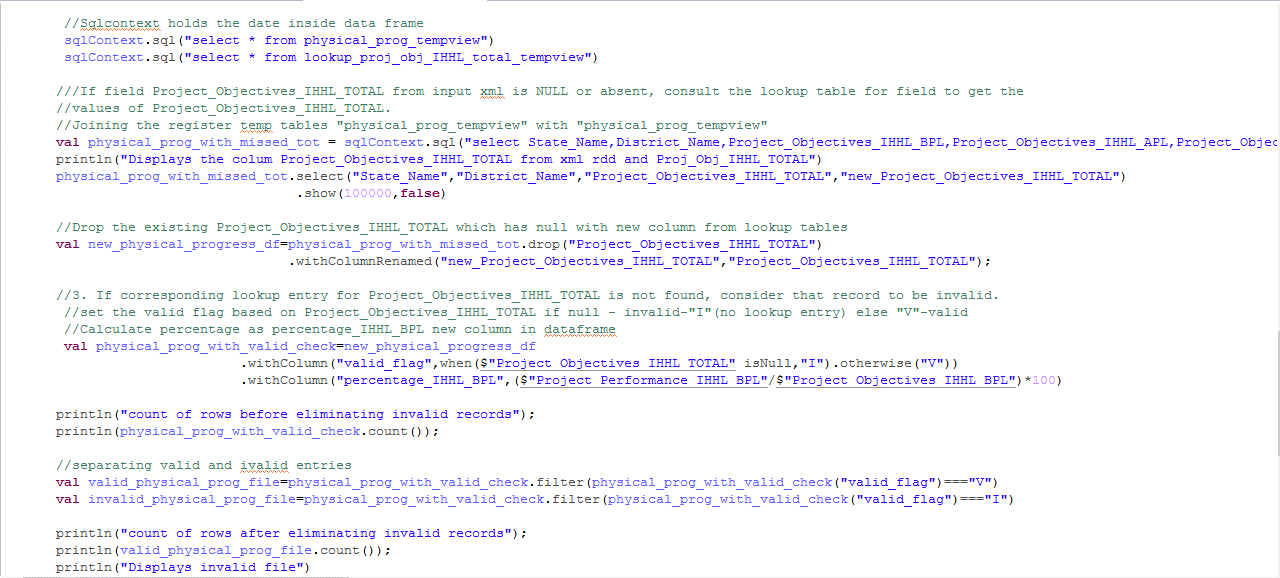
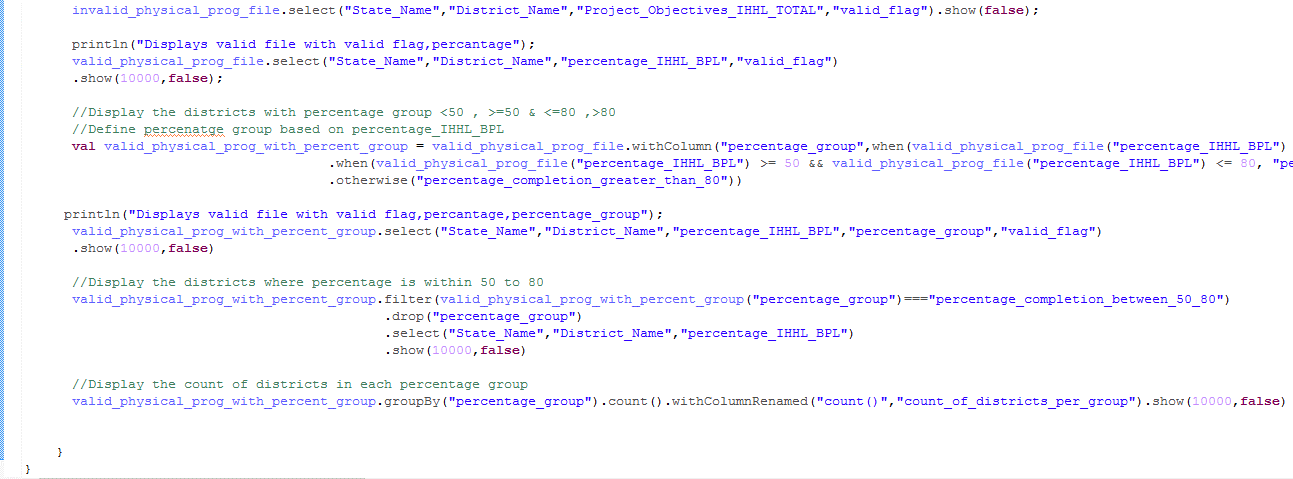
Input xml dataset is now transferred to HDFS by Flume agent.

**Data enrichment,Validation(using Spark):**

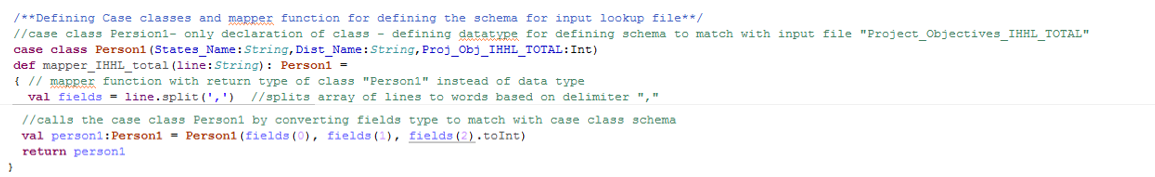
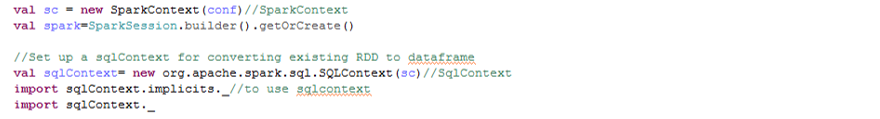
1. If any of Project\_Objectives\_IHHL\_APL or Project\_Objectives\_IHHL\_APL is NULL or absent, consider it as ‘0’
2. If field Project\_Objectives\_IHHL\_TOTAL from input xml is NULL or absent, consult the lookup file for field to get the values of Project\_Objectives\_IHHL\_TOTAL based on District and State name.
3. If corresponding lookup entry for Project\_Objectives\_IHHL\_TOTAL is not found, consider that record to be invalid.

**Code:**



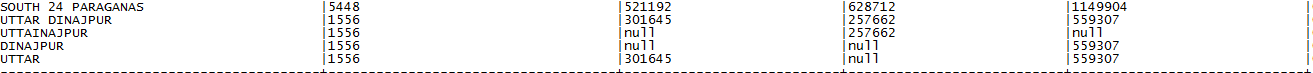
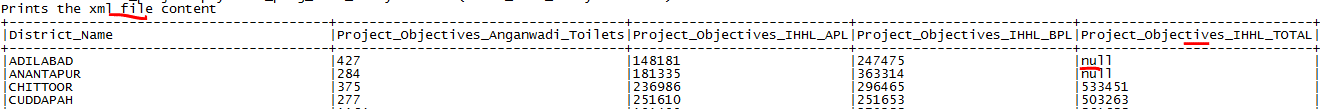
  

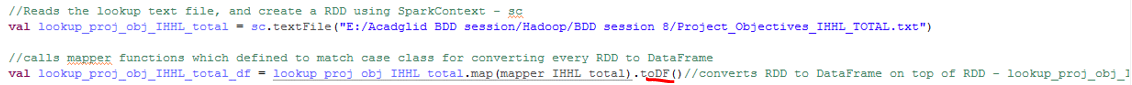
**Explanation:**

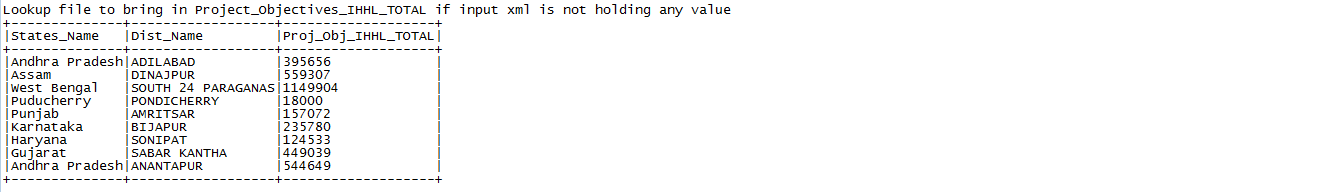
* Created a **case classe** called “Person1 for defining the schema and name of the fields in lookup input file
* Created a **mapper functions** for input file – which will convert the datatype of the fields to match with corresponding Person class schema and returns “Person” type
* Set the **log level to “Error”** to display only error messages - Logger.getLogger("org").setLevel(Level.ERROR) **and disabled progress-style** output(DAG stage details) - set("spark.ui.showConsoleProgress","false")
* Set up a **SparkContext** named – sc – for application " Statewise Districtwise Physical Progress" that runs locally using all available cores.
* Set up a **sqlContext** for converting existing RDD to **dataframe**
* Reads the xml file, and create a RDD using sqlContext and external package - com.databricks.spark.xml (to parse xml) based on “row” rowTag.

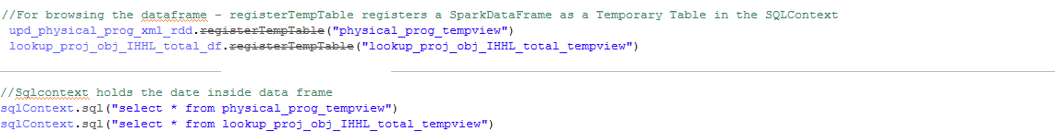


**Xml file content:**



* Reads the text lookup and create RDD using Spark Context.
* Calls mapper functions which is defined to match case class for converting every RDD to DataFrame.Calls mapper functions by passing mapper function with RDD and retunrs Dataframe as toDF() is used

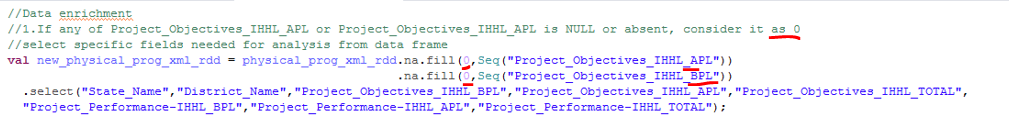
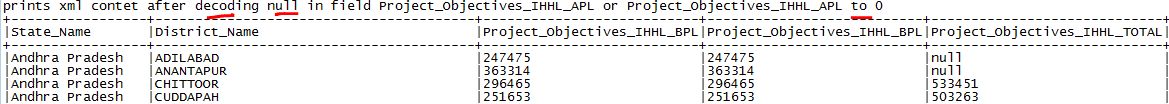


* For browsing the dataframe **- registerTempTable registers a SparkDataFrame** as a Temporary Table in the SQLContext and Sqlcontext holds the date of data frame
* Now as the data from RDD is converted to Dataframe,it can be queried with “SparkContext”.

**Data enrichment/validation - Spark:**

1. If any of Project\_Objectives\_IHHL\_APL or Project\_Objectives\_IHHL\_APL is NULL or absent, consider it as ‘0’

* Decoded “null” to “0” in both columns Project\_Objectives\_IHHL\_APL or Project\_Objectives\_IHHL\_APL with **.na.fill(0)** function.
* And selected specific columns needed from xml file for analysis with **.select**

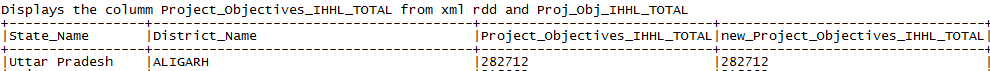
**Output:**

**2. If field Project\_Objectives\_IHHL\_TOTAL from input xml is NULL or absent, consult the lookup file for field to get the values of Project\_Objectives\_IHHL\_TOTAL based on District and State name.**

*val physical\_prog\_with\_missed\_tot = sqlContext.sql("select State\_Name,District\_Name,Project\_Objectives\_IHHL\_BPL,Project\_Objectives\_IHHL\_APL,Project\_Objectives\_IHHL\_TOTAL,Project\_Performance\_IHHL\_BPL,Project\_Performance\_IHHL\_APL,Project\_Performance\_IHHL\_TOTAL,****coalesce(CAST(nanvl(CAST(Project\_Objectives\_IHHL\_TOTAL AS INTEGER), CAST(Proj\_Obj\_IHHL\_TOTAL AS INTEGER)) AS INTEGER), CAST(Proj\_Obj\_IHHL\_TOTAL AS INTEGER)) as new\_Project\_Objectives\_IHHL\_TOTAL*** *from physical\_prog\_tempview p left outer join lookup\_proj\_obj\_IHHL\_total\_tempview l on District\_Name = Dist\_Name and State\_Name = States\_Name")*

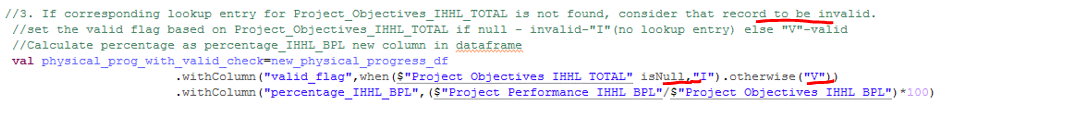
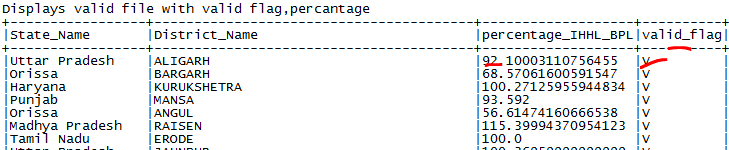
Joined the registered temp table “*physical\_prog\_tempview* " with " *lookup\_proj\_obj\_IHHL\_total\_tempview* " based on District and State name to get the value of “*Project\_Objectives\_IHHL\_TOTAL*” from lookup if that value is null in input xml file with SqlContext and ***coalesce& nanvl***

**Output:**

Project\_Objectives\_IHHL\_TOTAL is null hence taken from lookup file.  

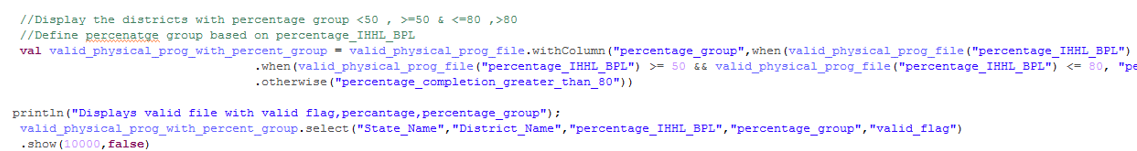
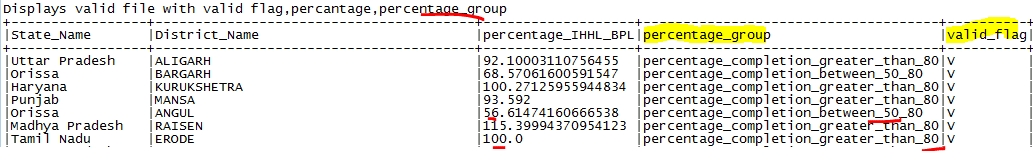
**3. If corresponding lookup entry for Project\_Objectives\_IHHL\_TOTAL is not found, consider that record to be invalid.**

* Using .withColumn defined a new column based on value in Project\_Objectives\_IHHL\_TOTAL field. If null then “I”(invalid) else valid record.
* And calculated percentage in new column – (performance/objective)\*100

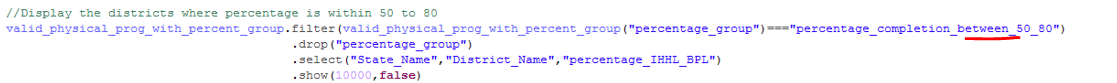
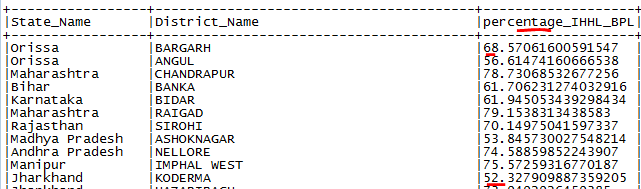
**Output:**

**Spark Problem statement – Data analysis:**

**1. Display the districts with percentage group <50 , >=50 & <=80 ,>80**

For the problem statement, **created and defined a new column "percenategroup" in existing dataframe** -as < 50 , 50-80, 80 > - based on calculated percenatage values in lookup file using withColumn – which **is to create or add columns dynamically to the new data frame from existing DF****Output:**

**2. Display the districts where percentage is within 50 to 80**

On the valid records filter the percentage\_group which is between\_50\_and\_80.**Output:**

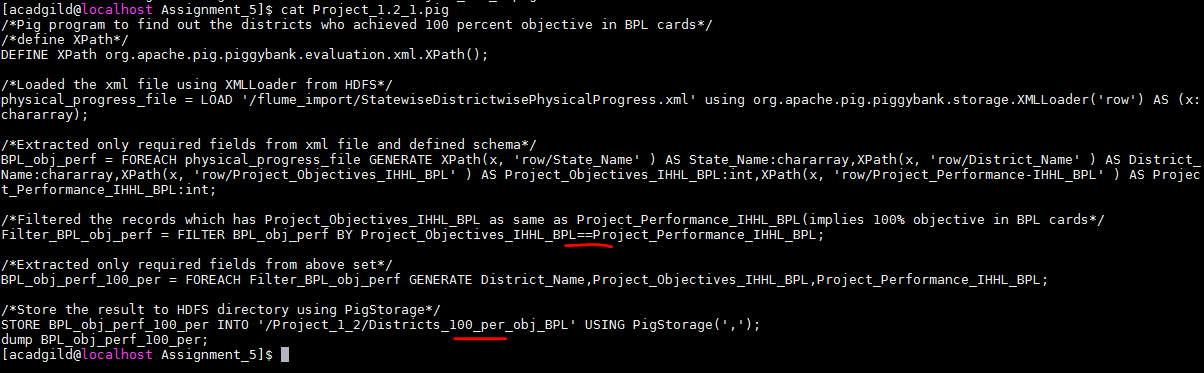
**3. Display the count of districts in each percentage group**

Group by percentage\_group and get count of records in a separate columns which is count of districts in each percentage group.**Output:**

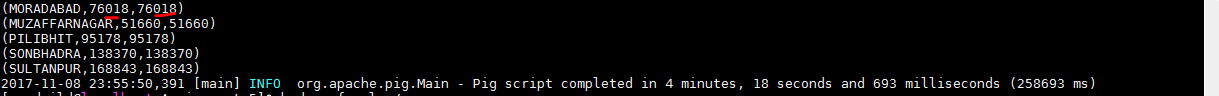
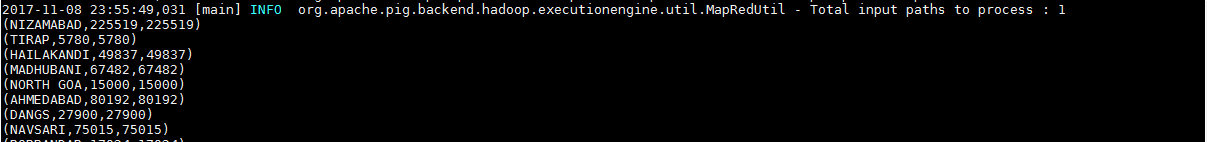
**Pig/MR Problem statement – Data analysis:**

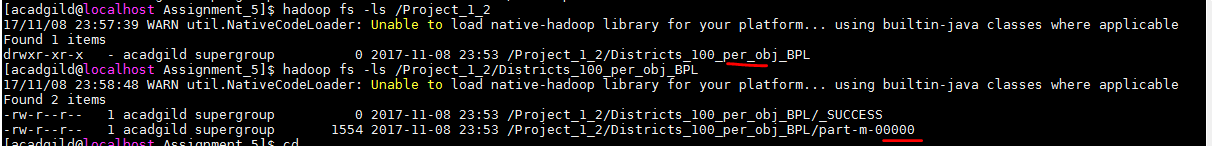
**1. Find out the districts who achieved 100 percent objective in BPL cards**

**Export the results to mysql using sqoop**

**Pig program:**

**Program invocation:**

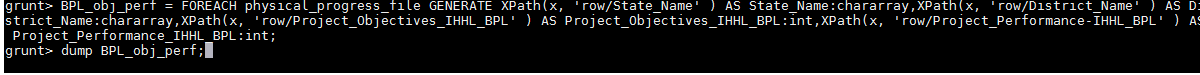
**Output:**

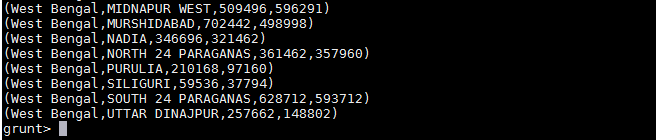


**Commands:**

**i) Loaded the xml file using XMLLoader(as input file is xml file) from HDFS**

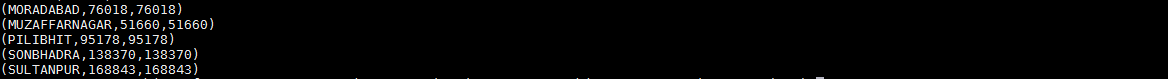


**ii) Extracted only required fields from xml file using XPath and defines a schema**



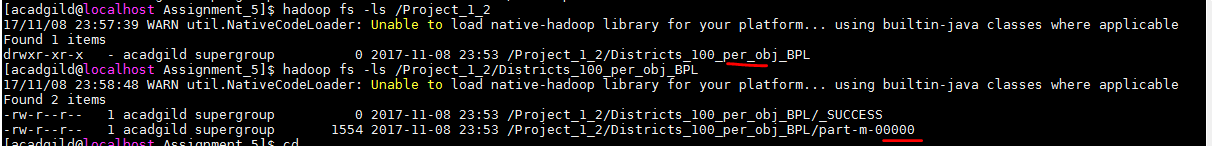
**iii) Filtered the records which has Project\_Objectives\_IHHL\_BPL as same as Project\_Performance\_IHHL\_BPL(implies 100% objective in BPL cards – implies who achieved 100 percent objective in BPL cards)**

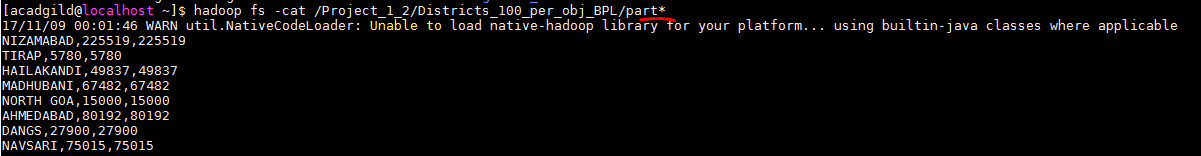
**iv) Extracted only required fields from above set**

**Final output:**

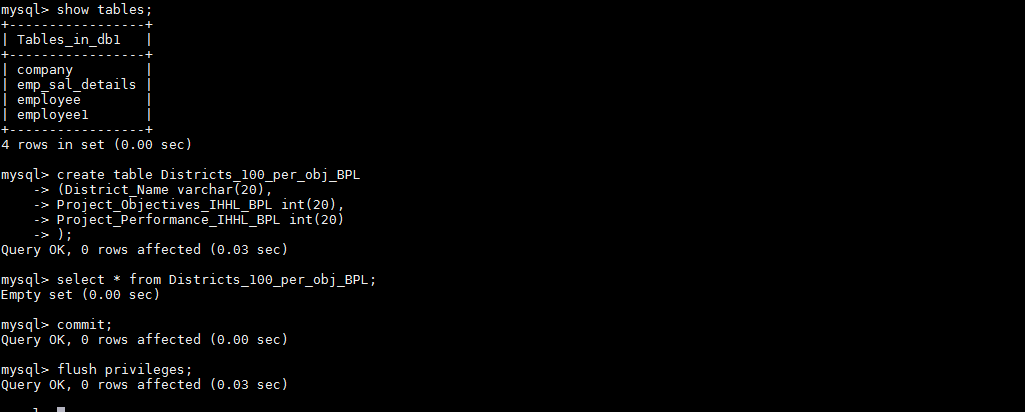
**v) Store the result to HDFS directory using PigStorage for further transferring it to MySql.**

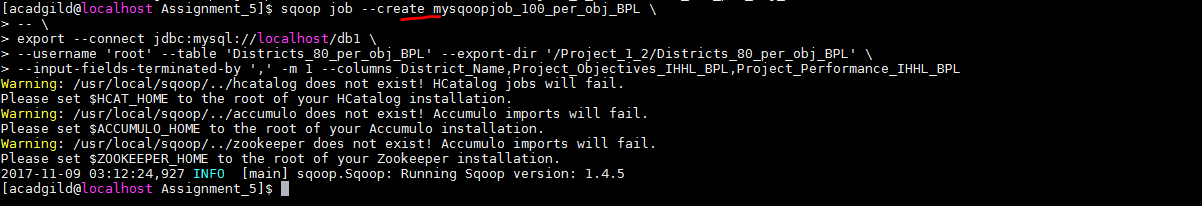






**Sqoop export from HDFS to Mysql:**

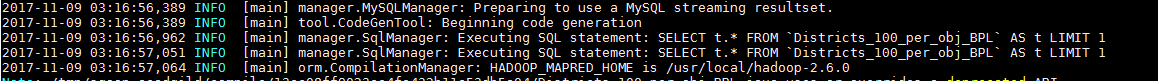
**For transferring data from HDFS to Mysql table should exist.Created a new table “Districts\_100\_per\_obj\_BPL” with columns as from above result inside ‘db1’ database.**

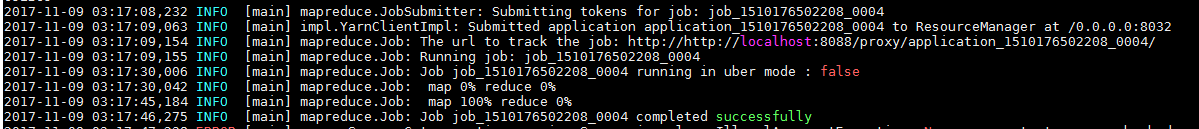
**Sqoop job:**

**Sqoop job execution:**



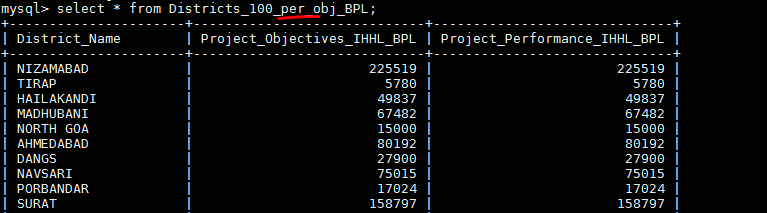
**Sqoop log:**

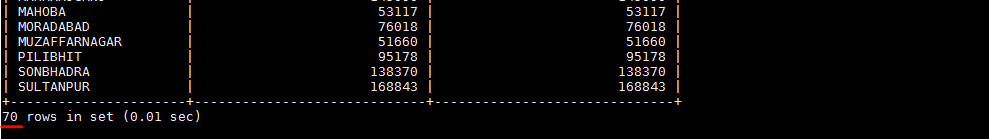




**Mysql table:**

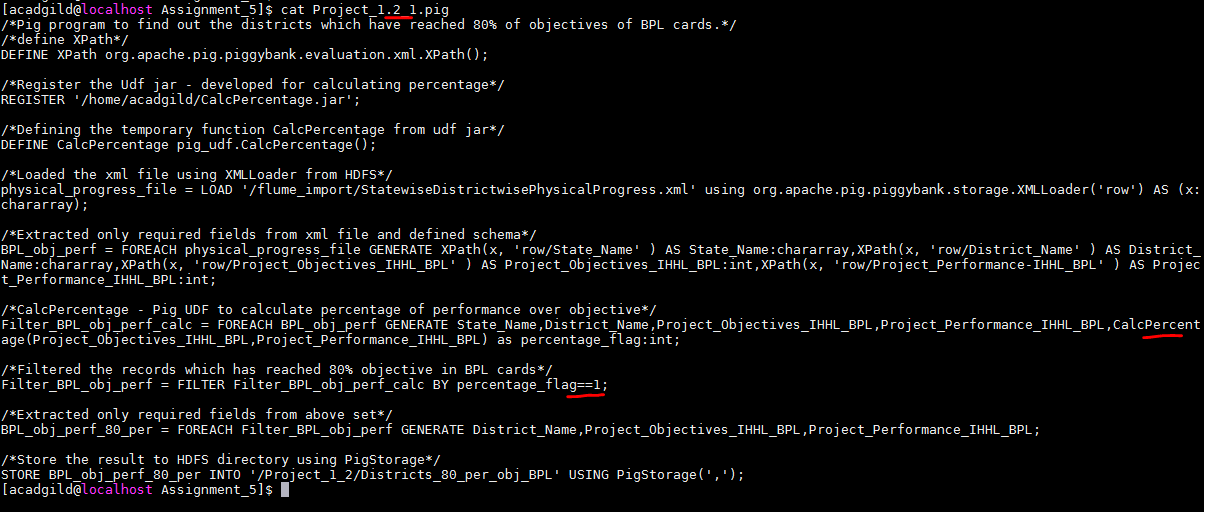
Data got loaded from HDFS to Mysql table.



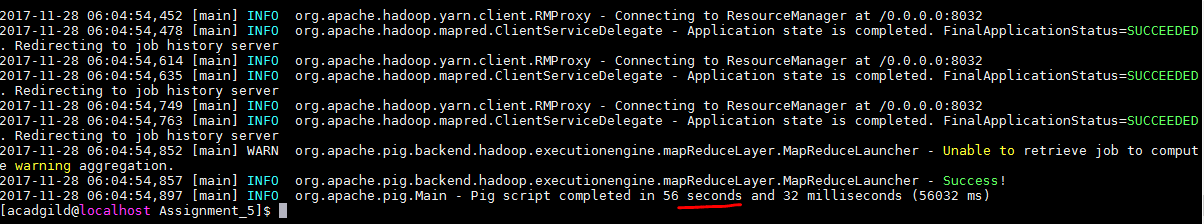


**2.Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards.**

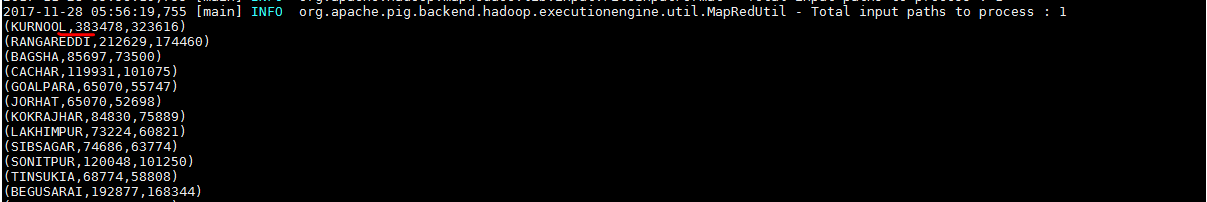
**Export the results to MySQL using Sqoop.**

**Pig program:**

**Program invocation:**



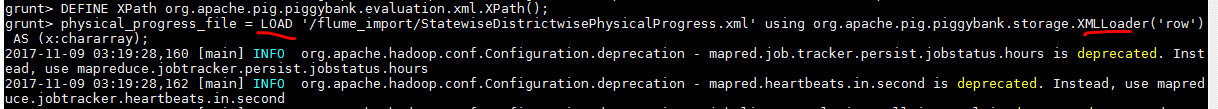
**Output:**



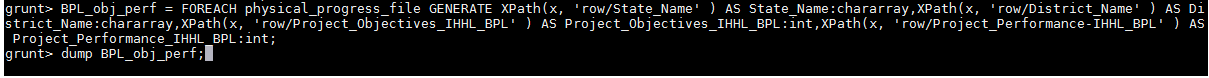
**Commands:**

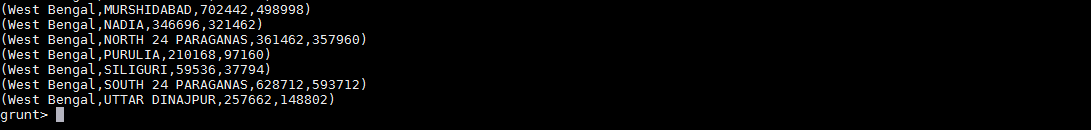
i) define XPath

ii) Loaded the xml file using XMLLoader from HDFS



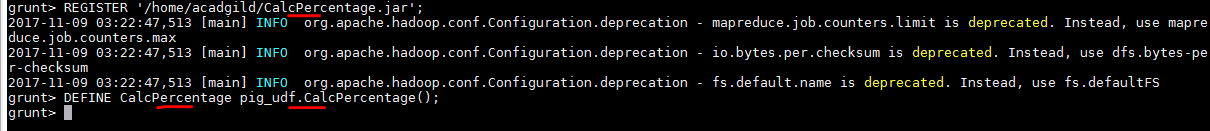
iii) Extracted only required fields from xml file and defined schema





iv) Register the Udf jar - developed for calculating percentage

v) Defining the temporary function CalcPercentage from udf jar

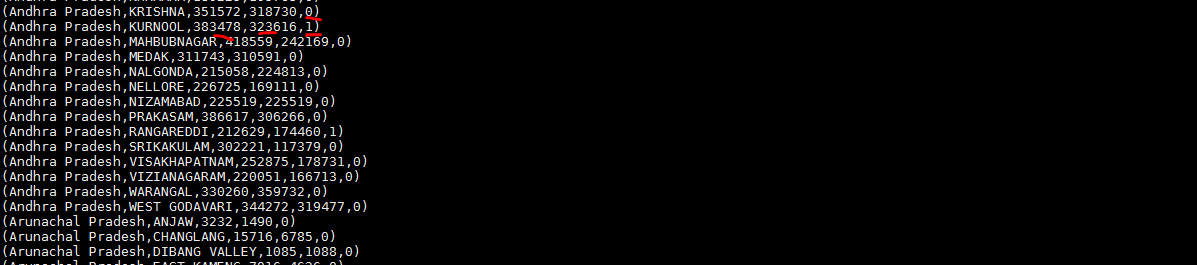


vi) Perform **CalcPercentage - Pig UDF to calculate percentage of performance over objective**

**and to set the percentage\_flag to “1” if percentage calculated 80%**

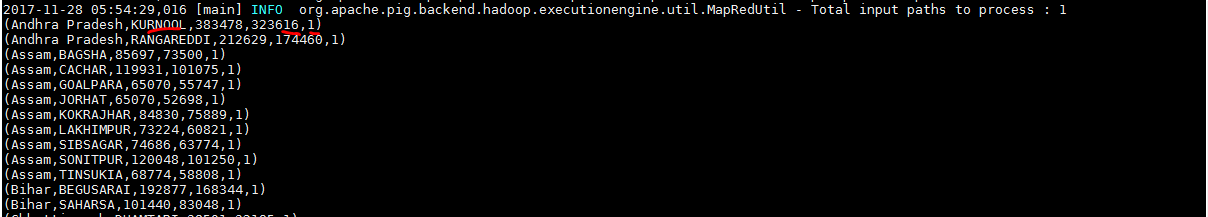
to “1” . Call UDF by passing 2 arguments - objective and performance in the temporary function created above. UDF/function will return “percentage\_flag” – “1” or “0”





vii) Filtered the records which has reached 80% objective in BPL cards – by filtering the records having “percentage\_flag” as “1”.

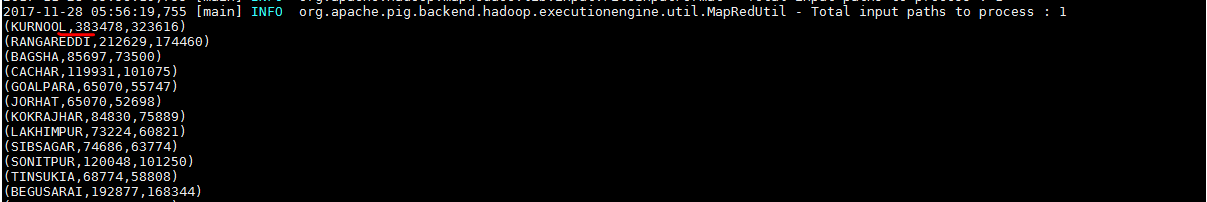




viii) Extracted only required fields from above set

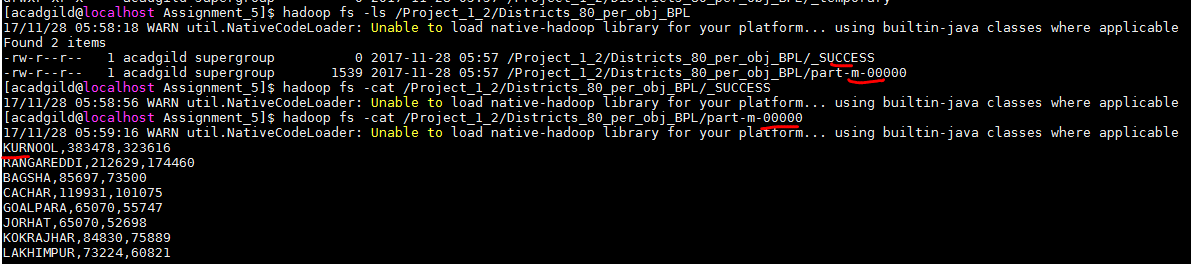


**Final output:**



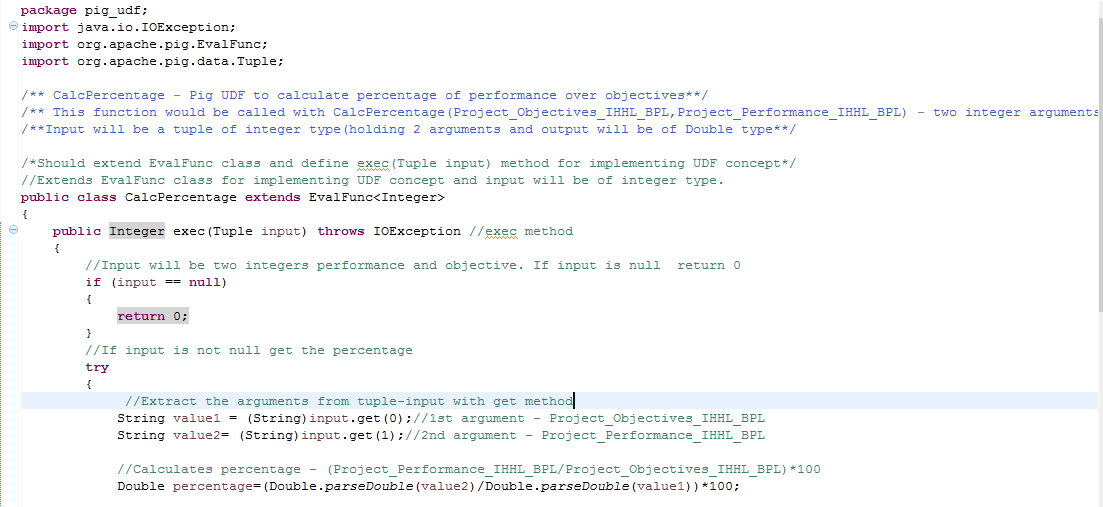
ix) Store the result to HDFS directory using PigStorage

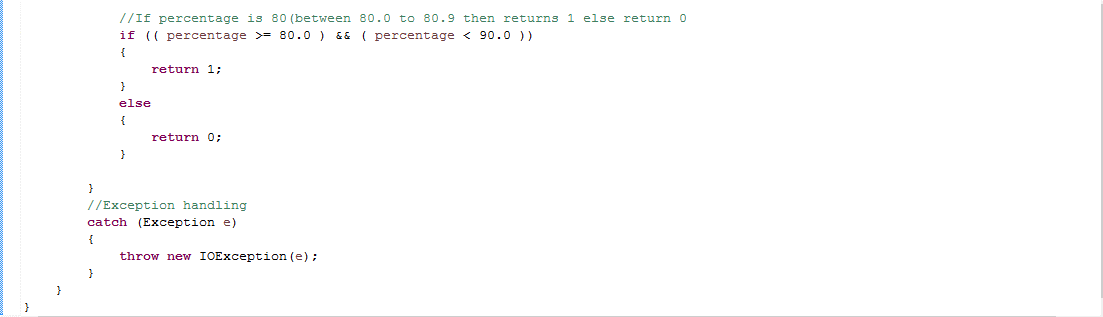




**Pig – UDF: to calculate percentage of objectives of BPL cards:**

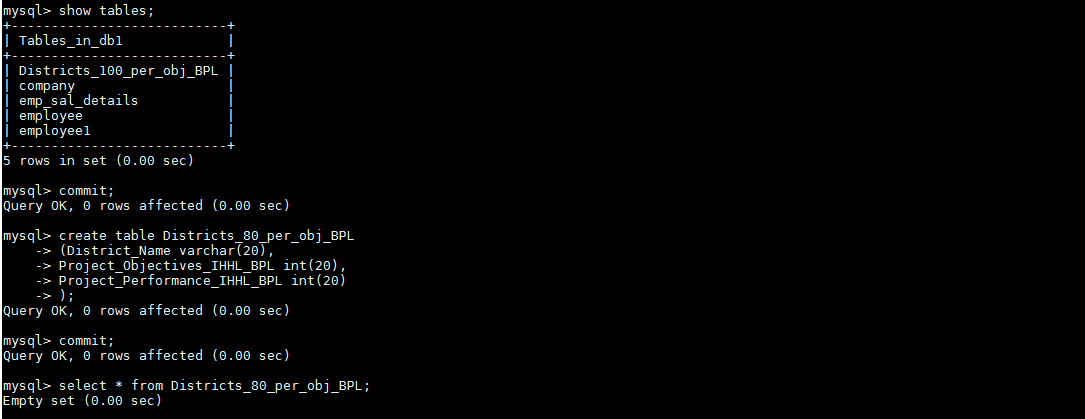
Designed to calculate the percentage of two inputs and return percentage\_flag which indicates whether that district/records has reached 80% of objective.



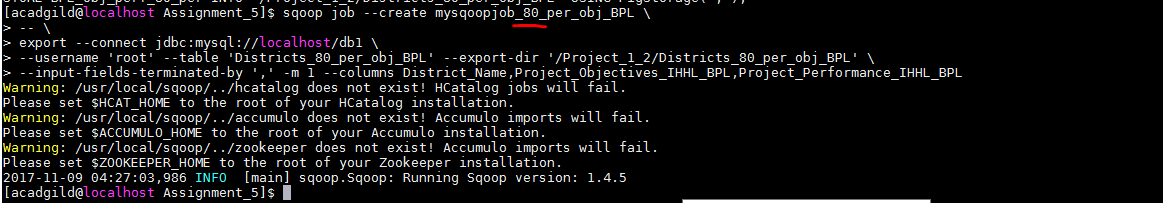


**Sqoop export from HDFS to Mysql:**

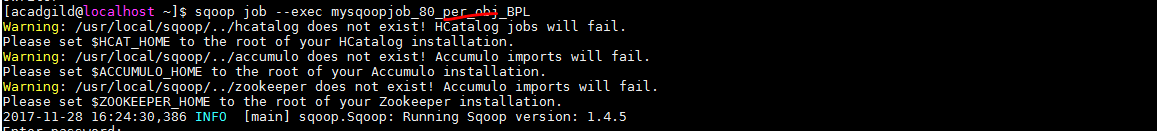
For transferring data from HDFS to Mysql table should exist.Created a new table “Districts\_80\_per\_obj\_BPL” with columns as from above result inside ‘db1’ database.



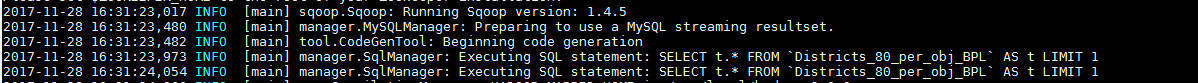
**Sqoop job:**

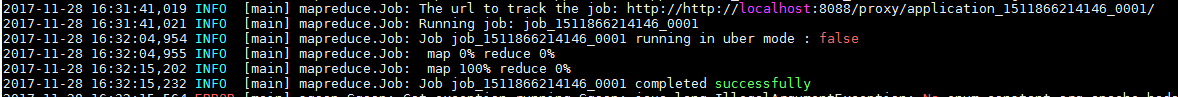


**Sqoop job execution:**



**Job log:**





**Mysql table:**

Data got loaded from HDFS to Mysql table.

