**Project 2:-**

**Problem Statement:-Develop MR program to parse logs and convert request string into structured format (/a/b/c/d => a b c d)**

**IP**

**Category-1**

**Category-2**

**page**

**status\_code**

**Solution:-**

Created a MAP only job to parse the logs and convert request string into structured format:-

**WeblogMapper class:-**

**package** com.eng.mr;

**import** java.io.IOException;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapreduce.Mapper;

**import** java.util.regex.\*;

**public** **class** WeblogMapper **extends** Mapper<Object, Text, Text, Text>

{

**private** Pattern p =

Pattern.*compile*("^([\\d.]+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(.+?)\" (\\d{3}) (\\d+) \"([^\"]+)\" \"([^\"]+)\"");

**private** Text outputKey = **new** Text();

**private** Text outputValue = **new** Text();

@Override

**protected** **void** map(Object key, Text value, Context context) **throws** IOException, InterruptedException

{

String entry = value.toString();

Matcher m = p.matcher(entry);

**if** (!m.matches())

{

**return**;

}

outputKey.set(m.group(1)); //ip

StringBuilder b = **new** StringBuilder();

b.append(',');

String rh []=m.group(5).split(" ");

String category []=rh[1].split("\\/");

**if** (category.length==4)

{

b.append(category[1]);//category 1 53.49.121.225

b.append(',');

b.append(category[2]);//category 2

b.append(',');

b.append(category[3]);//page

}

**else** **if** (category.length==3)

{

b.append(category[1]);//category 1 53.49.121.225

b.append(',');

b.append(" ");//page

b.append(',');

b.append(category[2]);//category 2

}

**else** **if** (category.length==2)

{

b.append(" ");//category 1 53.49.121.225

b.append(',');

b.append(" ");//category 2

b.append(',');

b.append(category[1]);//page

}

**if** (category.length==5)

{

b.append(category[1]);//category 1 53.49.121.225

b.append(',');

b.append(category[2]);//category 2

b.append(',');

b.append(category[4]);//page

}

**else** **if** (category.length<=1)

{

b.append(" ");//Category 1

b.append(',');

b.append(" ");//Category 2

b.append(',');

b.append(" ");//page

}

b.append(',');

b.append(m.group(6)); //code

outputValue.set(b.toString());

context.write(outputKey, outputValue);

}

}

**WeblogDriver class:-**

package com.eng.mr;

import java.io.IOException;

import org.apache.hadoop.fs.Path;

//import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WeblogDriver {

public static void main(String[] args) throws IOException,

InterruptedException, ClassNotFoundException {

Job job = new Job();

job.setJobName("Weblog");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

String input=args[0];

String outpath = args[1];

FileInputFormat.setInputPaths(job, new Path(input));

FileOutputFormat.setOutputPath(job, new Path(outpath));

job.setJarByClass(WeblogDriver.class);

job.setMapperClass(WeblogMapper.class);

job.setNumReduceTasks(0);

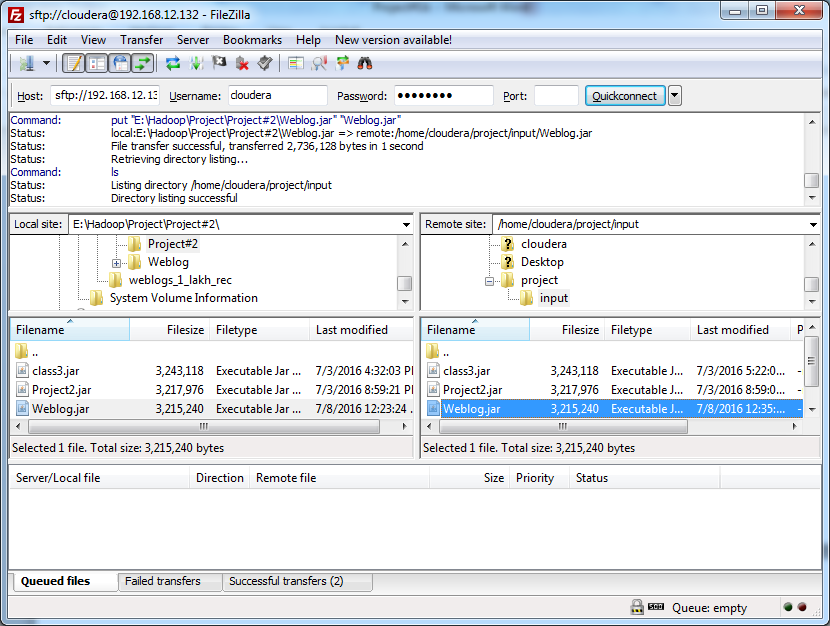
//job.setReducerClass(WordCountReducer.class);

job.waitForCompletion(true);

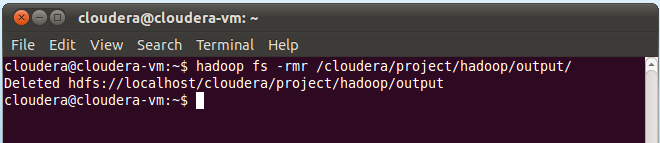
}

}

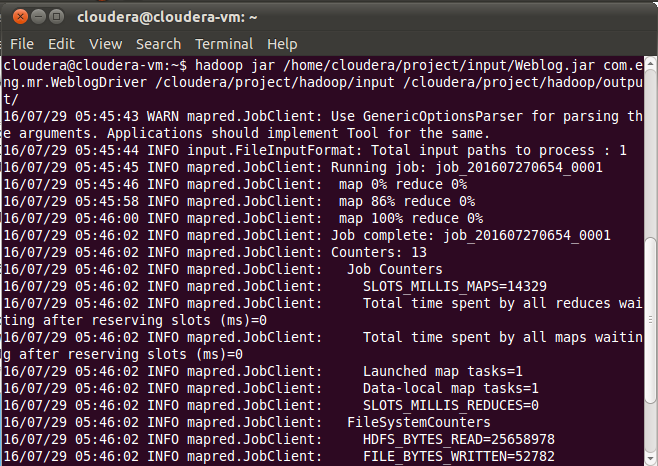
**Transfer of jar file to the VM:-**



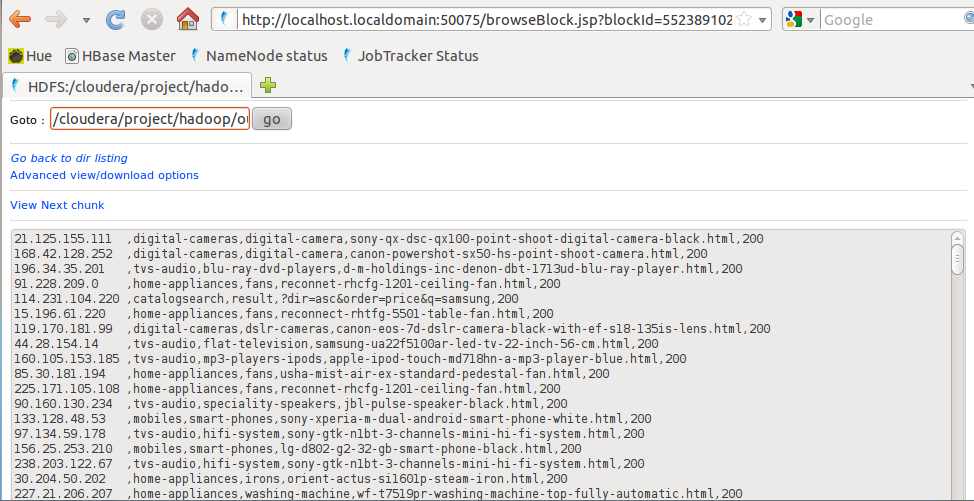
**Deleting old output file:-**



**Running MR Program:-**



**Output file:-**



*Pig is used to solve the project 3-5.*

*Pig provides an engine for executing data flows in parallel on Hadoop. It includes a*

*language, Pig Latin, for expressing these data flows*

**3. Count of page views by individual user**

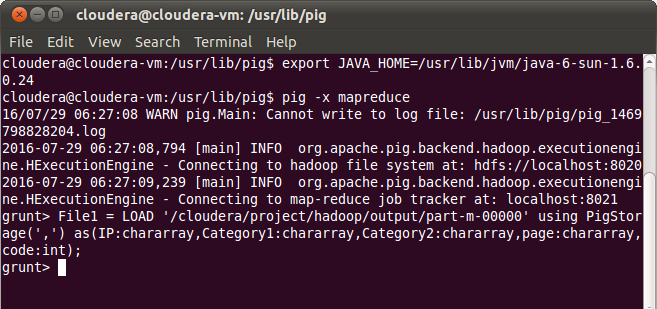
**IP, count(\*)**

**Loading the output from project2 into pig:-**

**Command**:-

File1 = LOAD '/cloudera/project/hadoop/output/part-m-00000' using PigStorage(',') as(IP:chararray,Category1:chararray,Category2:chararray,page:chararray,code:int);

**Screenshot:-**



**Group the records as per the IP:-**

**Command:-**

GPRD = group File1 by IP;

**Count the grouped record in previous step:-**

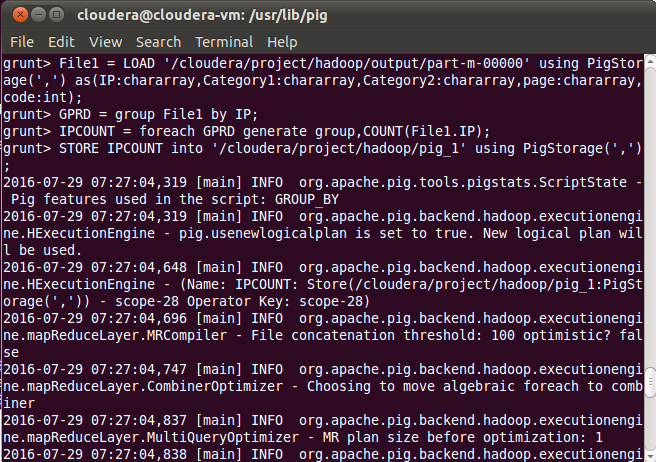
IPCOUNT = foreach GPRD generate group,COUNT(File1.IP)

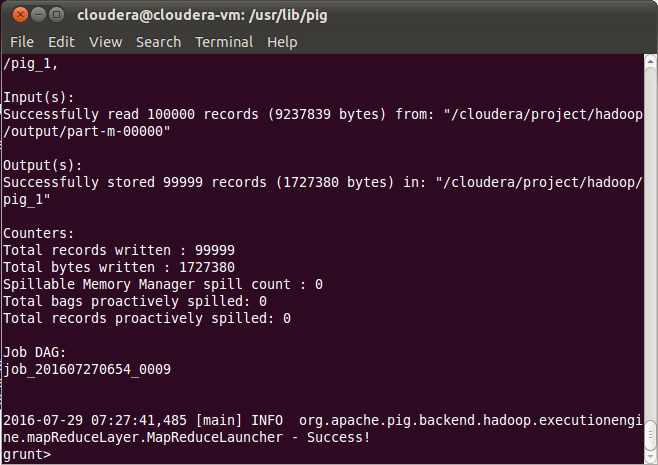
**Store the count results into HDFS:-**

**Command:-**

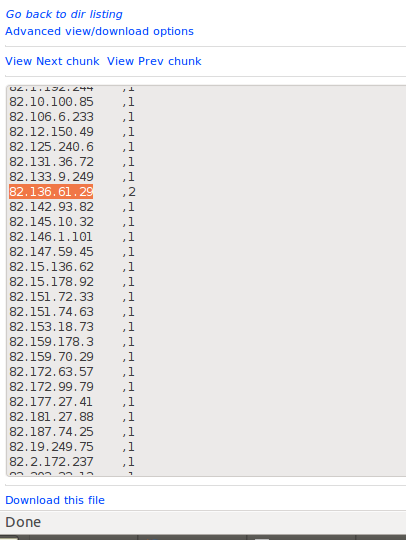
**STORE IPCOUNT into ‘/cloudera/project/hadoop/pig\_1’ using PigStorage(‘,’);**

**Screenshot:-**





**O\P:- IP and their count:-**



**4. Top / Bottom 5: catagery-1/ catagery-2 / page /users (Exclude status code other than 200, also exclude record related to css/js/image)**

**Top 5 and bottom 5 records**

**Category, total\_number\_views**

**page, total\_number\_views**

**IP, total\_number\_of\_views**

**TOP 5 records by Category1:-**

**Filter the records in File1 having status code as 200:-**

**Command:-**

File1\_status\_code\_200 = FILTER File1 by code == 200;

**Group the filtered records by category1**

**Command:-**

GRPD\_Category1 = group File1\_status\_code\_200 by Category1;

**Generate count for each category1:-**

**Command:-**

CNT\_FOR\_Category1 = foreach GRPD\_Category1 generate group,COUNT(File1\_status\_code\_200.Category1) as counting;

**Sort the results from above step in descending order .**

**Command:-**

SORTED\_CATEGORY1\_DEC = ORDER CNT\_FOR\_Category1 by counting DESC;

**Limit the sorted output to 5 in order to obtain top 5 category1:-**

**Command:-**

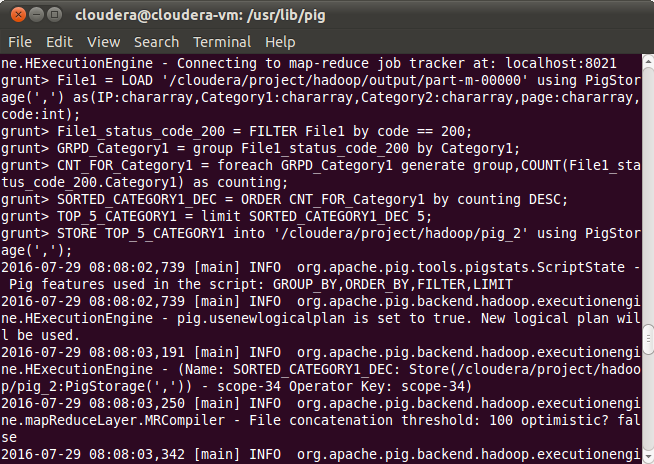
TOP\_5\_CATEGORY1 = limit SORTED\_CATEGORY1\_DEC 5;

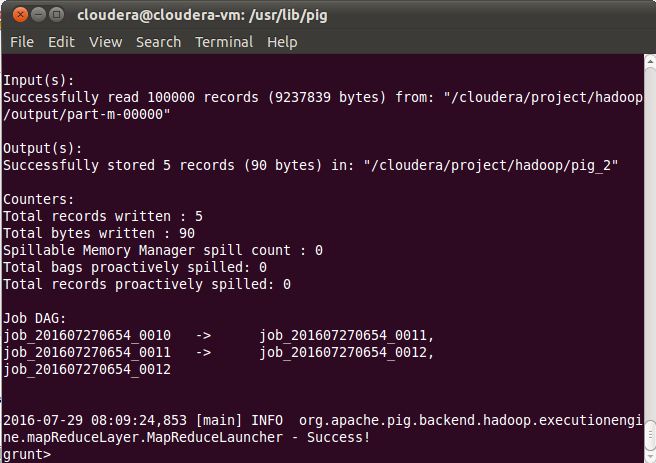
**Store the output to HDFS:-**

**Command:-**

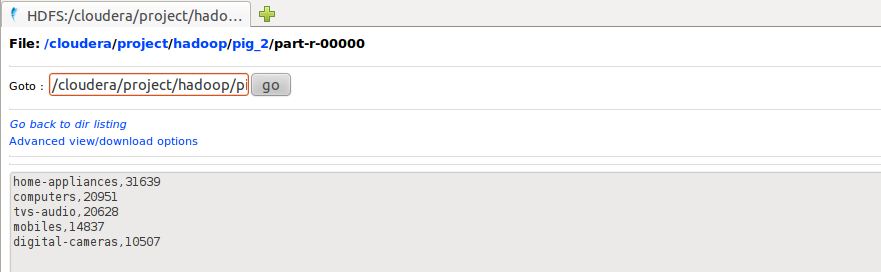
STORE TOP\_5\_CATEGORY1 into ‘/cloudera/project/hadoop/pig\_2’ using PigStorage(‘,’);

**Screenshot:-**





**O\P:- Top 5 records by category1:-**



**BOTTOM 5 records by Category1:-**

**Sort the CNT\_FOR\_Category1 in ascending order:-**

**Command:-**

SORTED\_CATEGORY1\_ASC = ORDER CNT\_FOR\_Category1 by counting ASC;

**Limit the sorted output to 5 in order to obtain top 5 category1:-**

**Command:-**

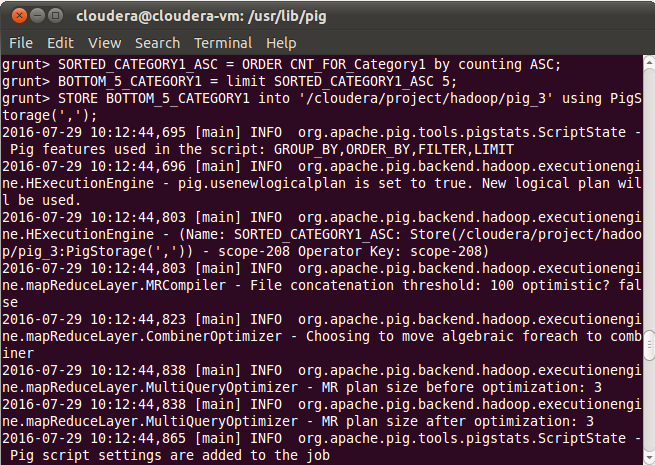
BOTTOM\_5\_CATEGORY1 = limit SORTED\_CATEGORY1\_ASC 5;

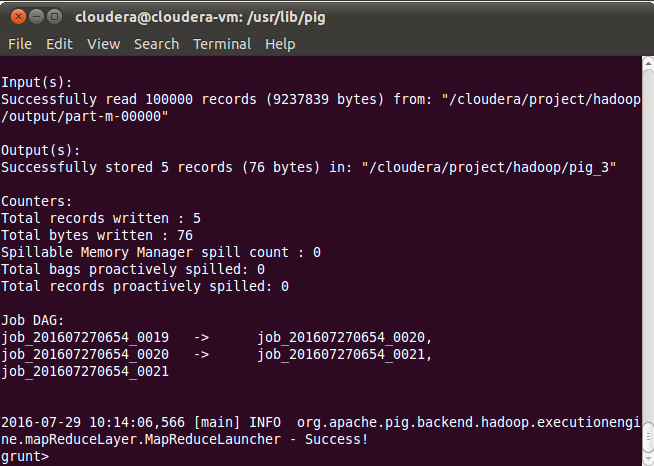
**Store the output to HDFS:-**

**Command:-**

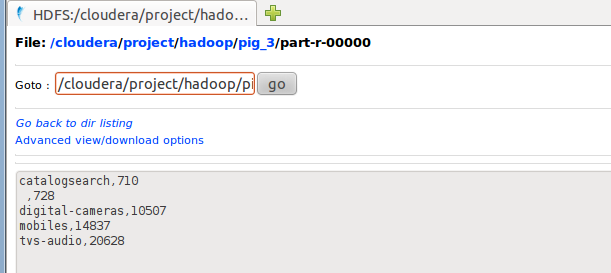
STORE BOTTOM\_5\_CATEGORY1 into ‘/cloudera/project/hadoop/pig\_3’ using PigStorage(‘,’);

**Screenshot:-**





**O\P:- Bottom 5 records by category1:-**



**Category2**

**Top 5 records by Category2**

**Filter the records in File1 having status code as 200:-**

**Command:-**

File1\_status\_code\_200 = FILTER File1 by code == 200;

**Group the filtered records by category2**

**Command:-**

GRPD\_Category2 = group File1\_status\_code\_200 by Category2;

**Generate count for each category1:-**

**Command:-**

CNT\_FOR\_Category2 = foreach GRPD\_Category2 generate group,COUNT(File1\_status\_code\_200.Category2) as counting;

**Sort the results from above step in descending order.**

**Command:-**

SORTED\_CATEGORY2\_DEC = ORDER CNT\_FOR\_Category2 by counting DESC;

**Limit the sorted output to 5 in order to obtain top 5 category1:-**

**Command:-**

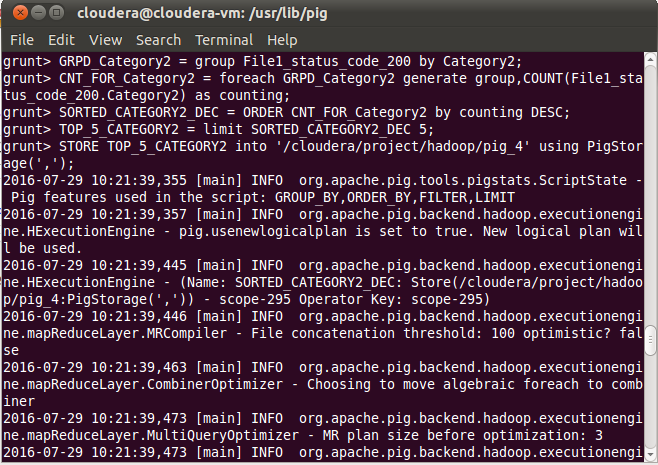
TOP\_5\_CATEGORY2 = limit SORTED\_CATEGORY2\_DEC 5;

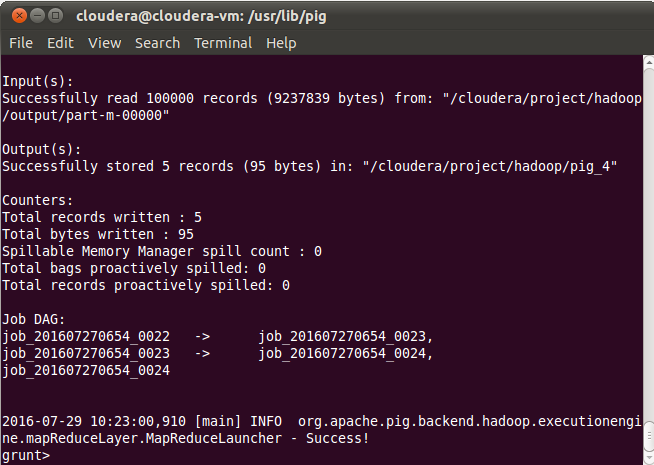
**Store the output to HDFS:-**

**Command:-**

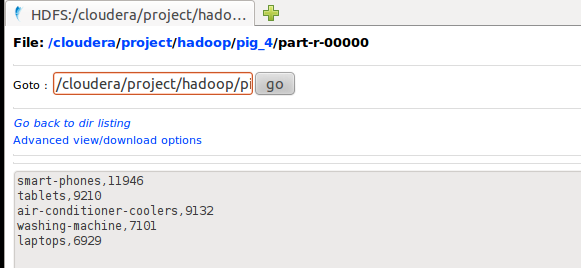
STORE TOP\_5\_CATEGORY2 into ‘/cloudera/project/hadoop/pig\_4’ using PigStorage(‘,’);

**Screenshot:-**





**O\P:-Top 5 records by category2:-**



**BOTTOM 5 records by Category2:-**

**Sort the CNT\_FOR\_Category2 in ascending order:-**

**Command:-**

SORTED\_CATEGORY2\_ASC = ORDER CNT\_FOR\_Category2 by counting ASC;

**Limit the sorted output to 5 in order to obtain top 5 category1:-**

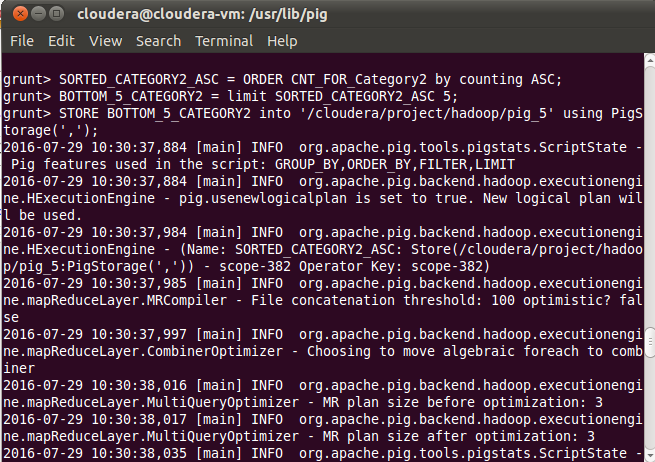
**Command:-**

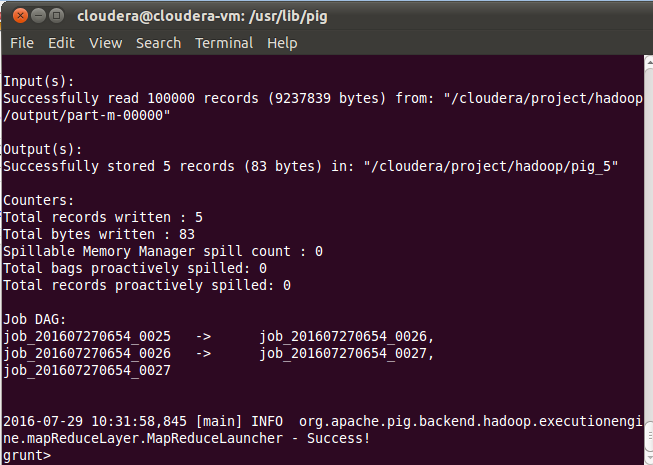
BOTTOM\_5\_CATEGORY2 = limit SORTED\_CATEGORY2\_ASC 5;

**Store the output to HDFS:-**

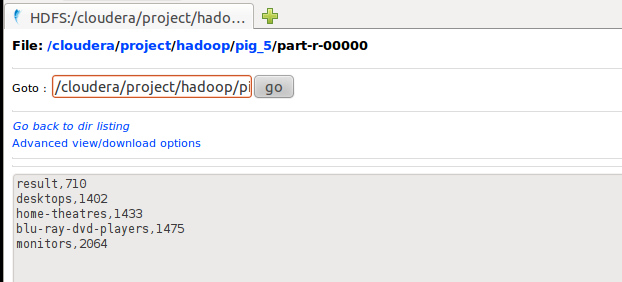
**Command:-**

STORE BOTTOM\_5\_CATEGORY2 into ‘/cloudera/project/hadoop/pig\_5’ using PigStorage(‘,’);





**O\P:-Bottom 5 records by category2**



**TOP 5 records by Page:-**

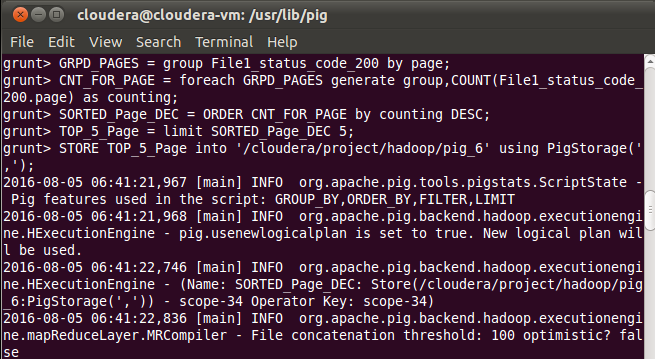
GRPD\_PAGES = group File1\_Sstatus\_code\_200 by page;

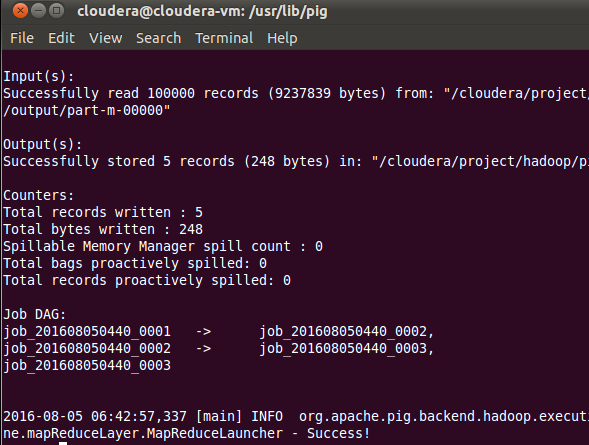
CNT\_FOR\_PAGE = foreach GRPD\_PAGES generate group,COUNT(File1\_status\_code\_200.page) as counting;

SORTED\_PAGE\_DEC = ORDER CNT\_FOR\_PAGE by counting DESC;

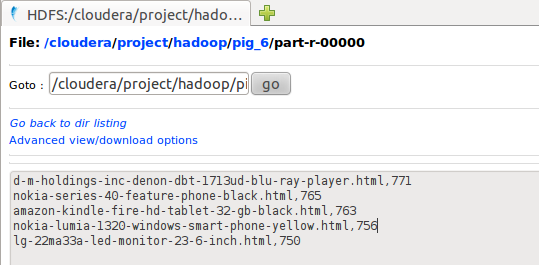
TOP\_5\_Page = limit SORTED\_PAGE\_DEC 5;

STORE TOP\_5\_Page into ‘/cloudera/project/hadoop/pig\_6’ using PigStorage(‘,’);





**Output:-TOP 5 Page count:-**

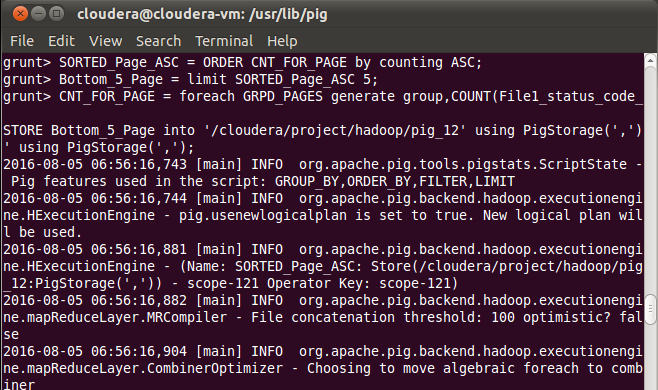


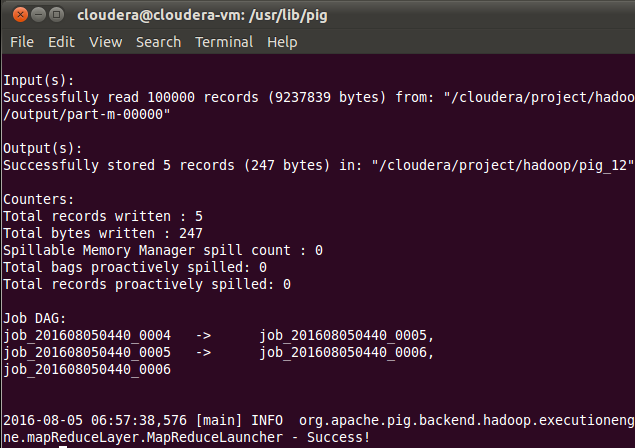
Bottom 5 page:-

SORTED\_PAGE\_ASC = ORDER CNT\_FOR\_PAGE by counting ASC;

Bottom\_5\_Page = limit SORTED\_PAGE\_ASC 5;

STORE Bottom\_5\_Page into ‘/cloudera/project/hadoop/pig\_12’ using PigStorage(‘,’);





**Output:-Bottom 5 pages:-**



**5. Total page views / Category wise pageviews / Unique pageviews**

**page,total\_number\_of\_views**

**category, total\_views**

**page, total\_number\_of\_unique\_views**

**Category1 wise view:-**

**Group the original output file without any filter by category1.**

**Command:-**

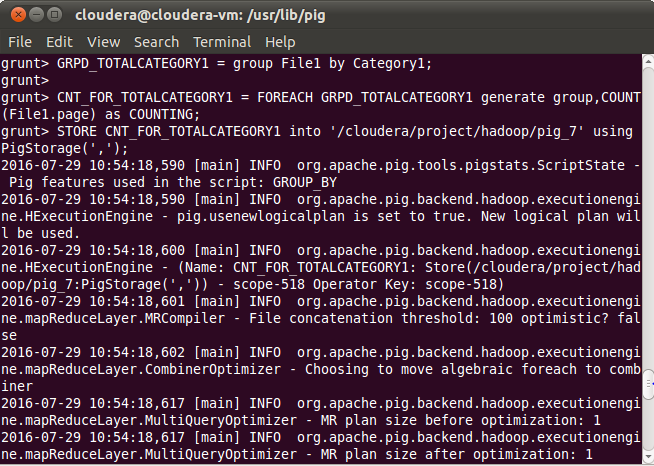
GRPD\_TOTALCATEGORY1 = group File1 by Category1;

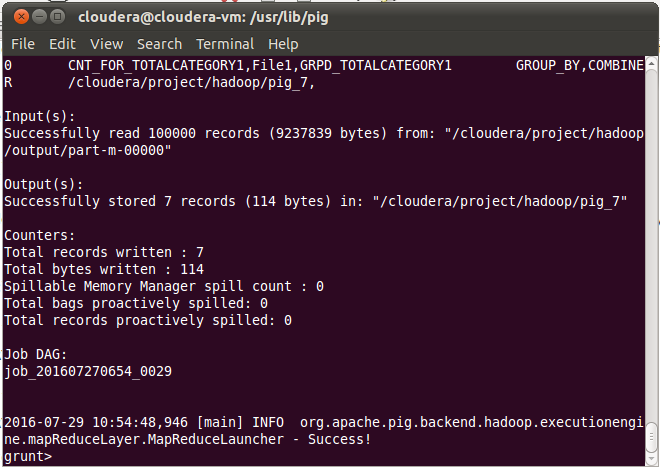
**Count the records grouped by category1.**

**Command:-**

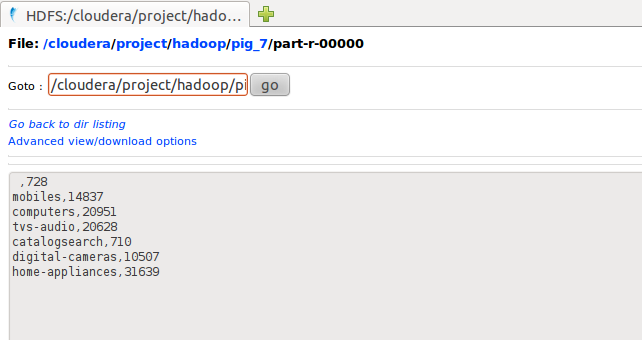
CNT\_FOR\_TOTALCATEGORY1 = FOREACH GRPD\_TOTALCATEGORY1 generate group,COUNT(File1.Category1) as counting;

STORE CNT\_TOTALCATEGORY1 into '/cloudera/project/hadoop/pig\_7' using PigStorage(',');





**O\P:-Category1 wise records:-**



**Category2 wise view:-**

**Group the original output file without any filter by category2.**

**Command:-**

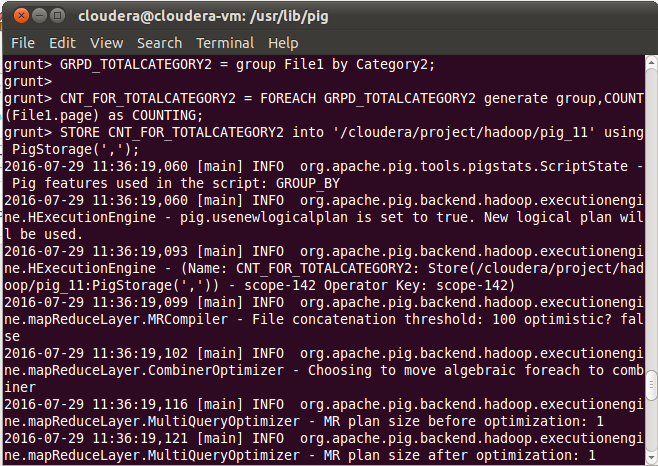
GRPD\_TOTALCATEGORY2 = group File1 by Category2;

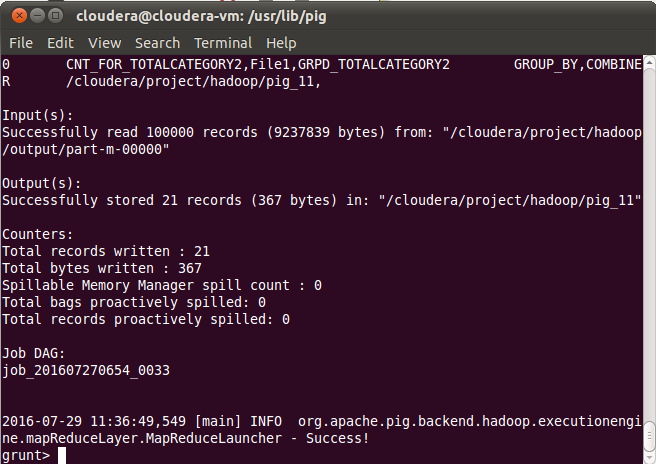
**Count the records grouped by category1.**

**Command:-**

CNT\_FOR\_TOTALCATEGORY2 = FOREACH GRPD\_TOTALCATEGORY2 generate group,COUNT(File1.Category2) as counting;

STORE CNT\_TOTALCATEGORY2 into '/cloudera/project/hadoop/pig\_11' using PigStorage(',');





**O\P:-Category2 wise records**

**Category1 wise view:-**

**Group the original output file without any filter by category1.**

**Command:-**

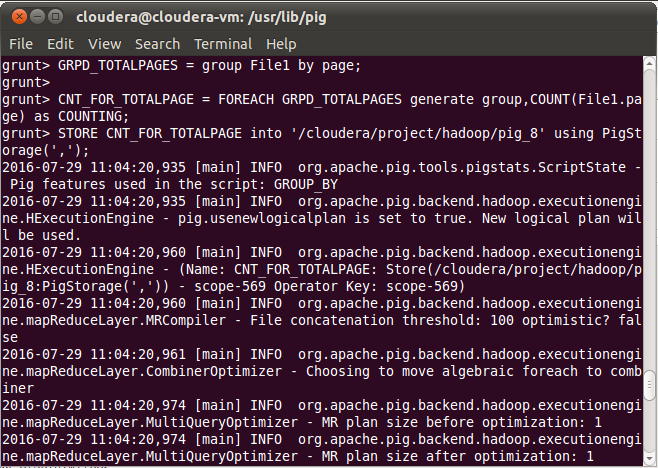
GRPD\_TOTALCATEGORY1 = group File1 by page;

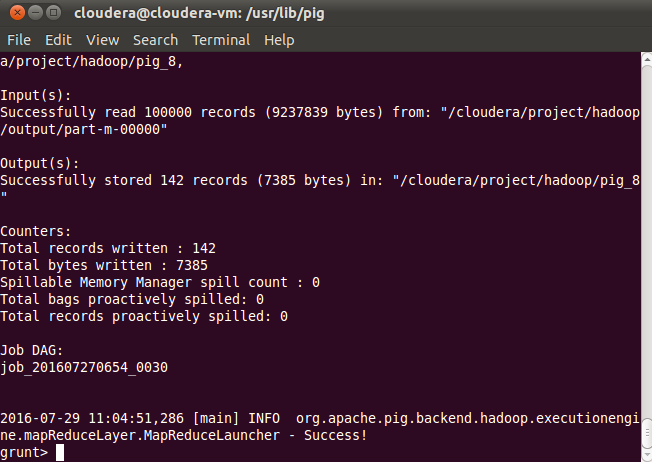
**Count the records grouped by category1.**

**Command:-**

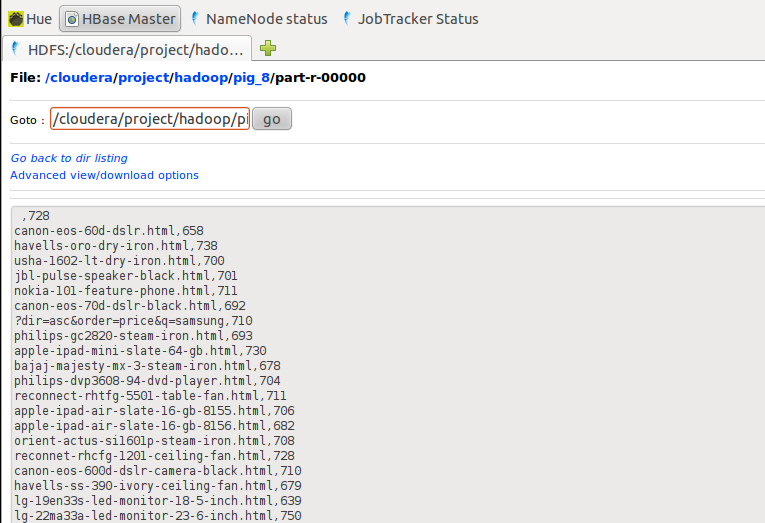
CNT\_FOR\_TOTALPAGE = FOREACH GRPD\_TOTALPAGE generate group,COUNT(File1.page) as counting;

STORE CNT\_FOR\_TOTALPAGE into '/cloudera/project/hadoop/pig\_8' using PigStorage(',');





**O\P:-Page wise records:-**



**Unique pageviews:-**

**Group the File1 by page.**

**Command:-**

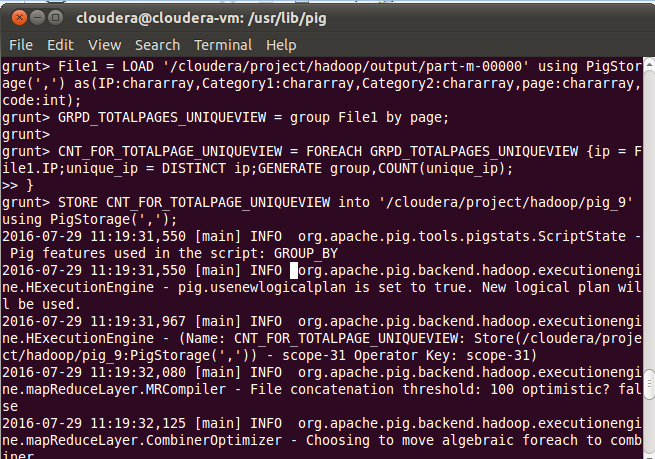
GRPD\_TOTALPAGES\_UNIQUEVIEW = group File1 by page;

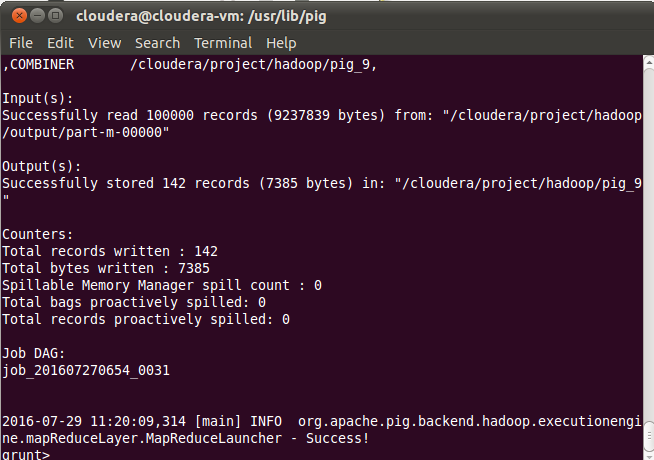
**For each group generate unique page count**

**Command:-**

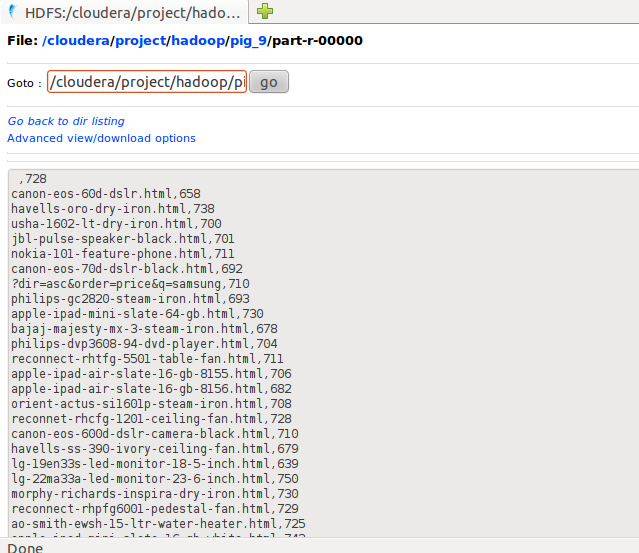
CNT\_FOR\_TOTALPAGE\_UNIQUEVIEW = FOREACH GRPD\_TOTALPAGES\_UNIQUEVIEW {ip = File1.IP;unique\_ip = DISTINCT ip;GENERATE group,COUNT(unique\_ip);}

STORE CNT\_FOR\_TOTALPAGE\_UNIQUEVIEW into '/cloudera/project/hadoop/pig\_9' using PigStorage(',');





**Output (Unique Page view):-**



6. Count of status code = 200 / 404 / 400 / 500

status\_code, count

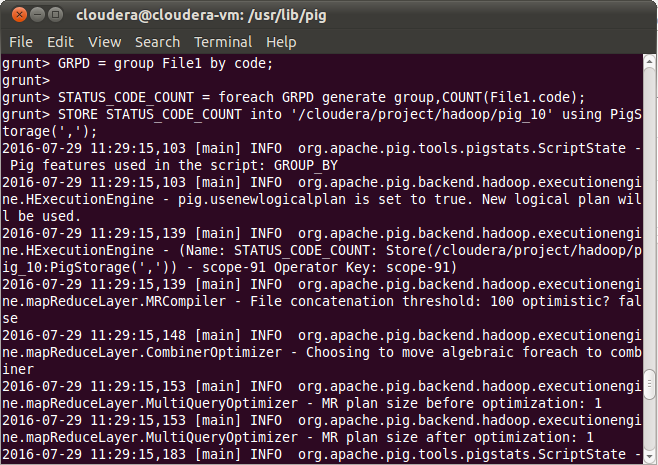
**Count of records with Status 200**

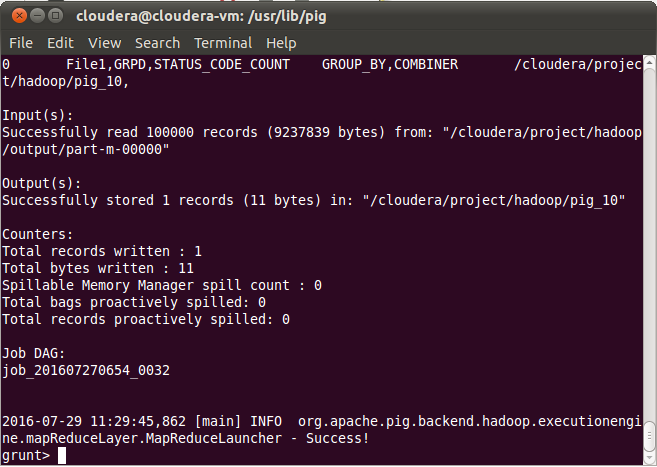
**Command:-**

GRPD = group File1 by code;

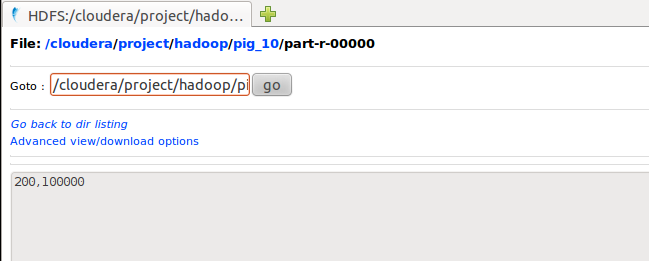
STATUS\_CODE\_COUNT = foreach GRPD generate group,COUNT(File1.code);

STORE STATUS\_CODE\_COUNT into ‘/cloudera/project/hadoop/pig\_10’ using PigStorage(‘,’);





**OUTPUT:-Count of records with code 200:-**

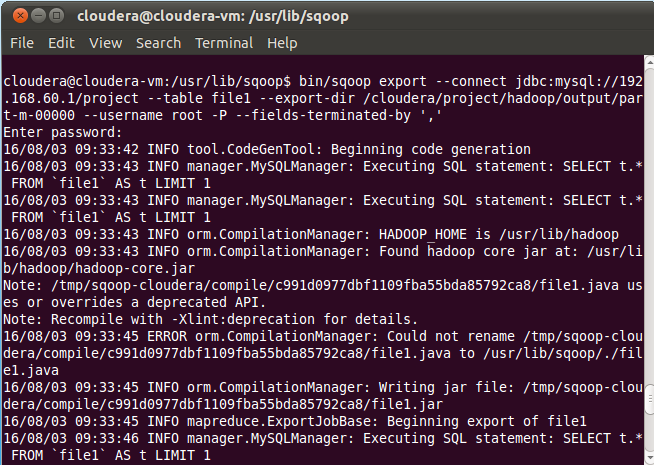


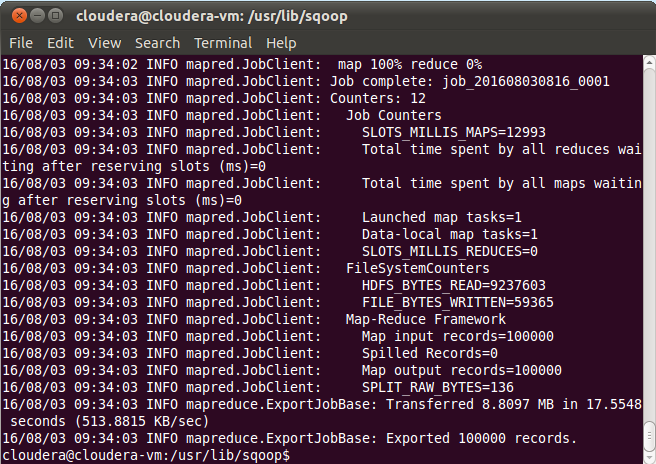
**7. Load results into tables in MySql Database using Sqoop.**

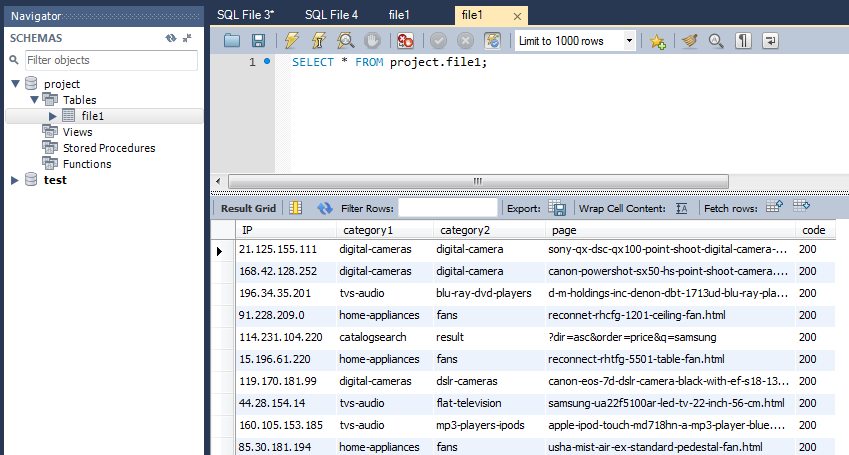
Loading data through SQOOP:-

**Loading parsed weblogs:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table file1 –export-dir /cloudera/project/hadoop/output/part-m-00000 –username root –P –fields-terminated-by ‘,’

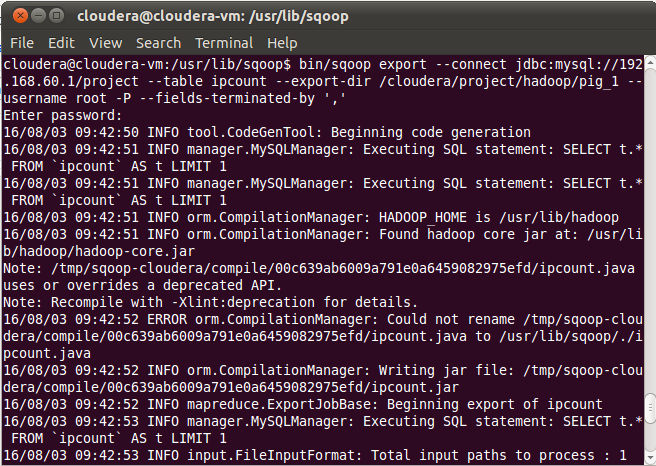


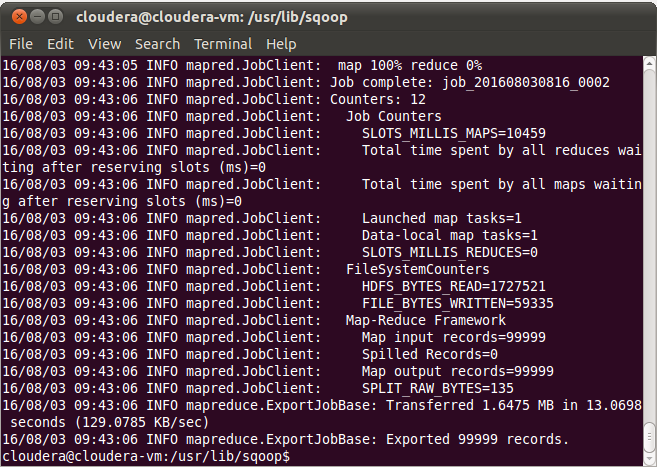


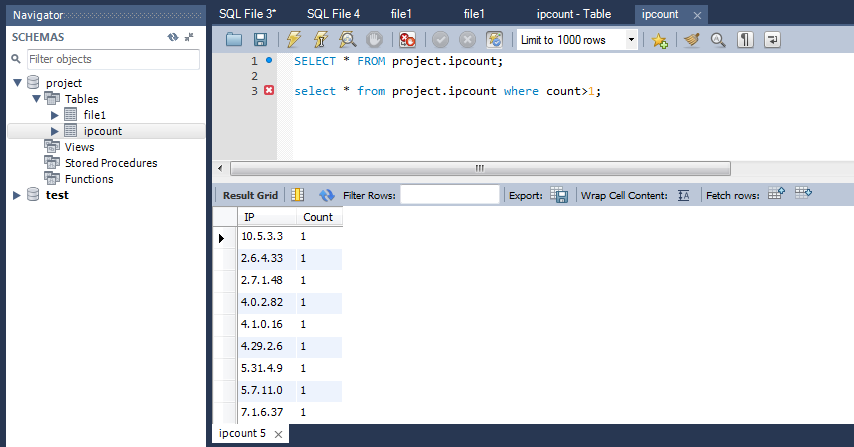


**IP count:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table ipcount –export-dir /cloudera/project/hadoop/pig\_1 –username root –P –fields-terminated-by ‘,’

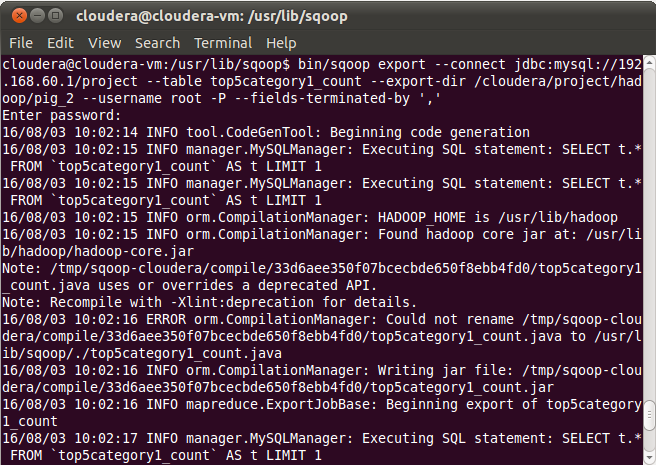


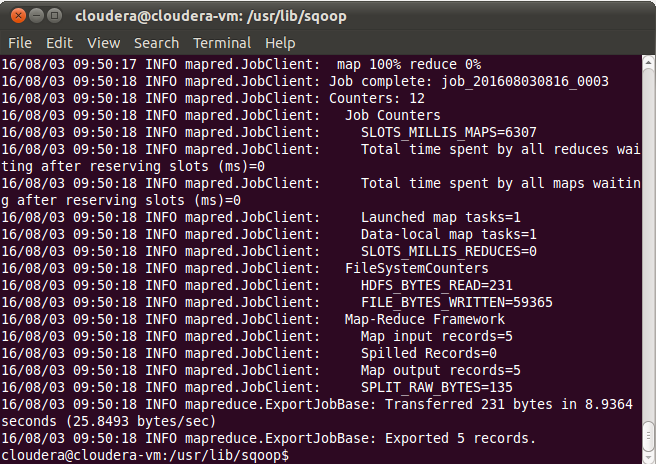


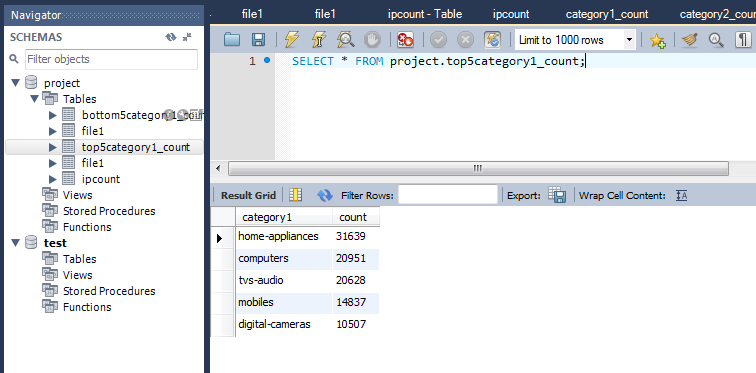


**Category1 TOP5 count**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table top5category1\_count –export-dir /cloudera/project/hadoop/pig\_2 –username root –P –fields-terminated-by ‘,’

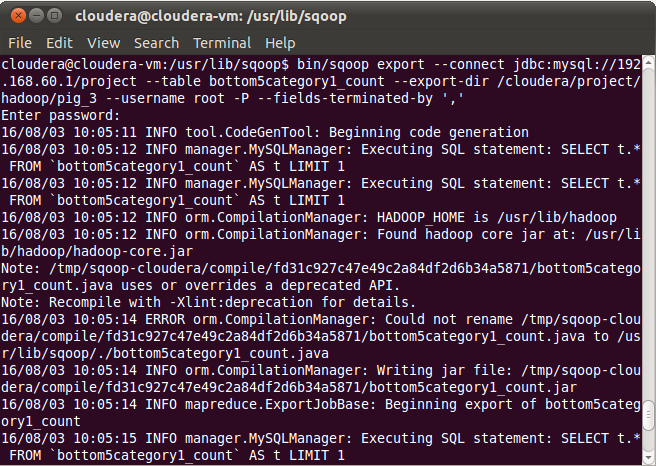


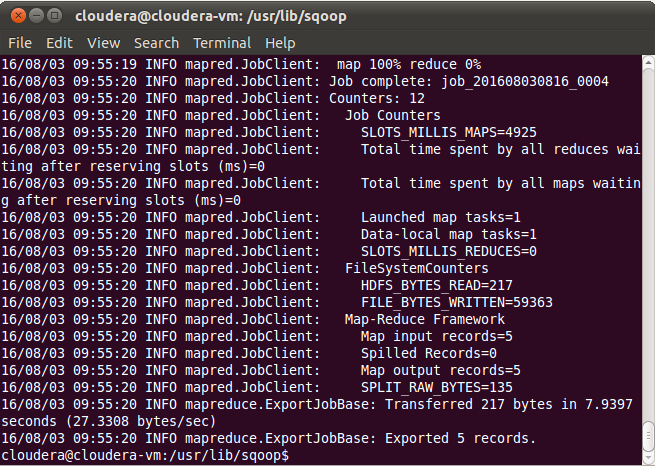


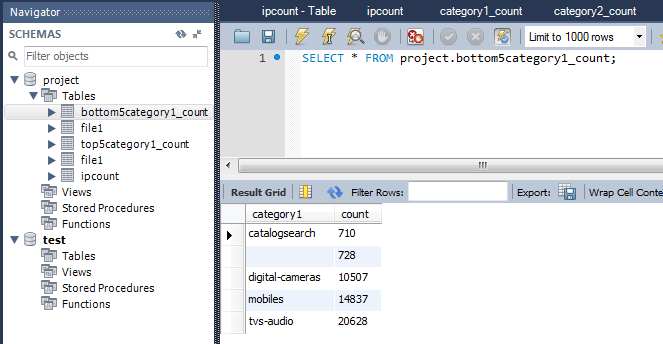


**Category1 bottom5 count:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table bottom5category1\_count –export-dir /cloudera/project/hadoop/pig\_3 –username root –P –fields-terminated-by ‘,’

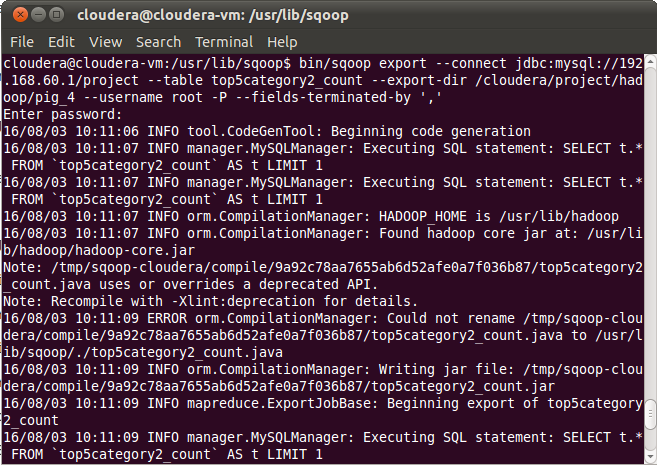


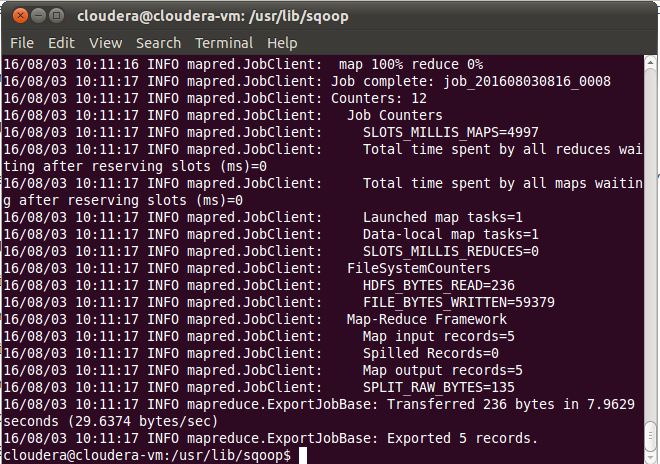


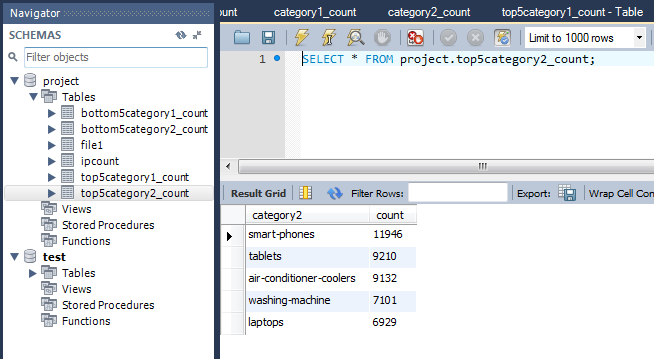


**Top 5 category 2:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table top5category2\_count –export-dir /cloudera/project/hadoop/pig\_4 –username root –P –fields-terminated-by ‘,’

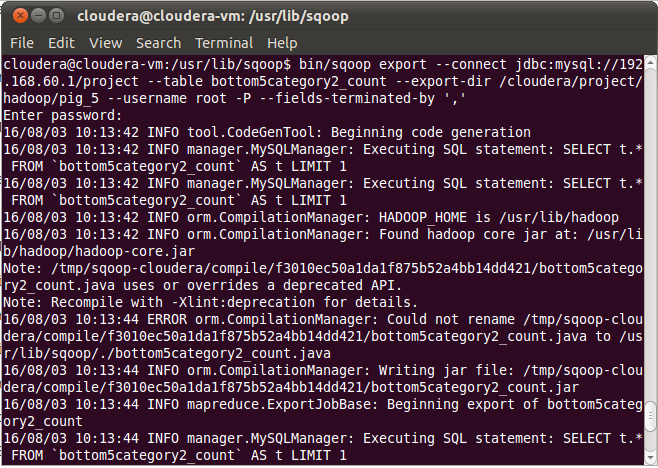


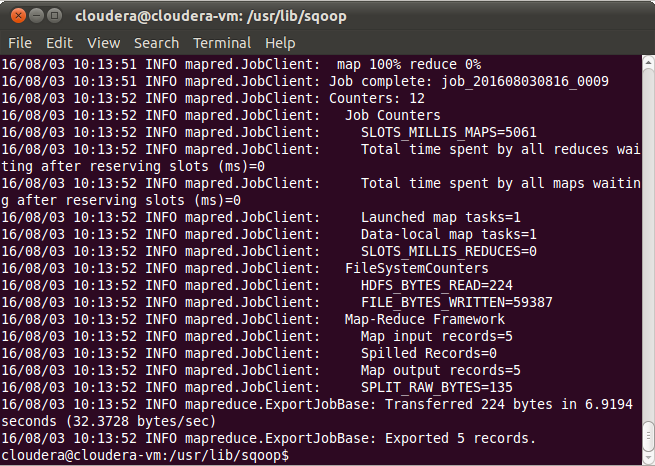


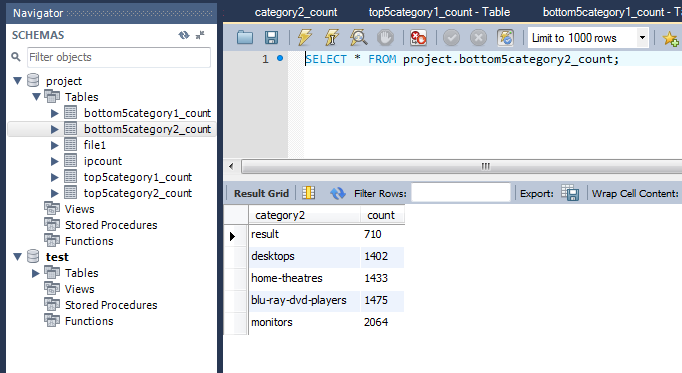


**Bottom 5 category 2**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table bottom5ctegory2\_count –export-dir /cloudera/project/hadoop/pig\_5 –username root –P –fields-terminated-by ‘,’

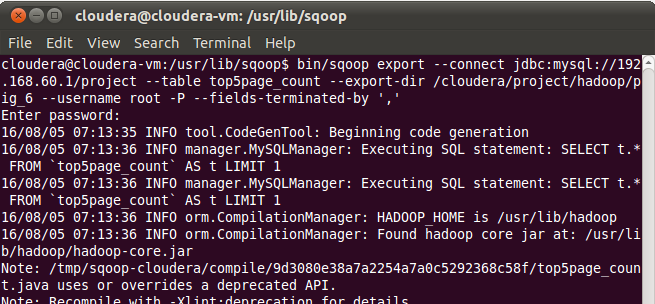


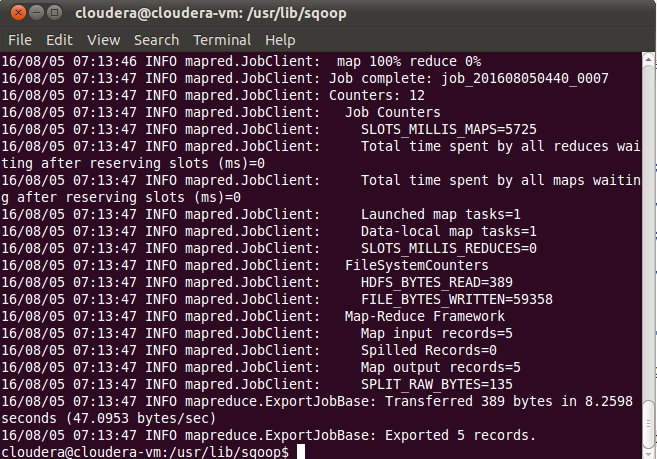


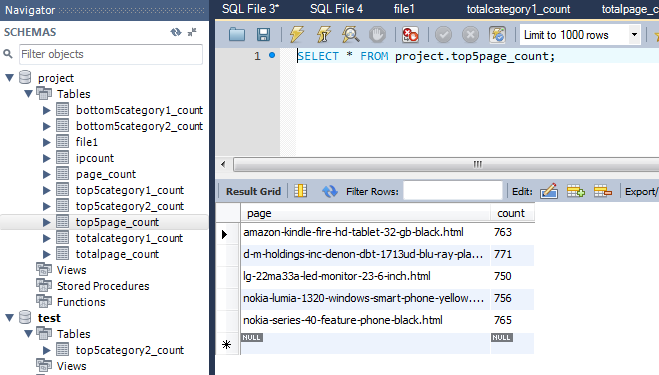


**TOP 5 Page count:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table top5page\_count –export-dir /cloudera/project/hadoop/pig\_6 –username root –P –fields-terminated-by ‘,’

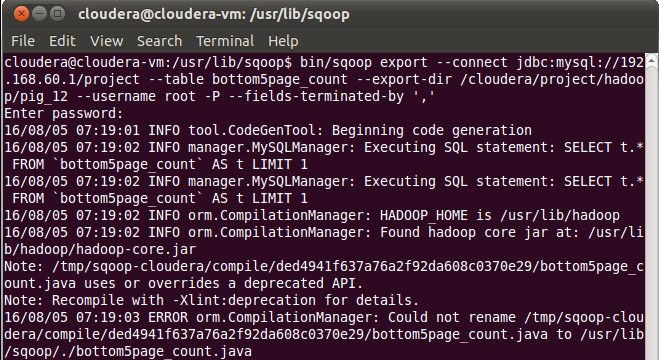


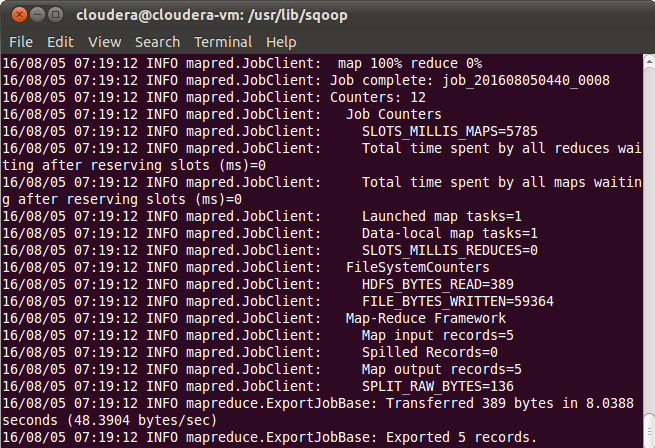


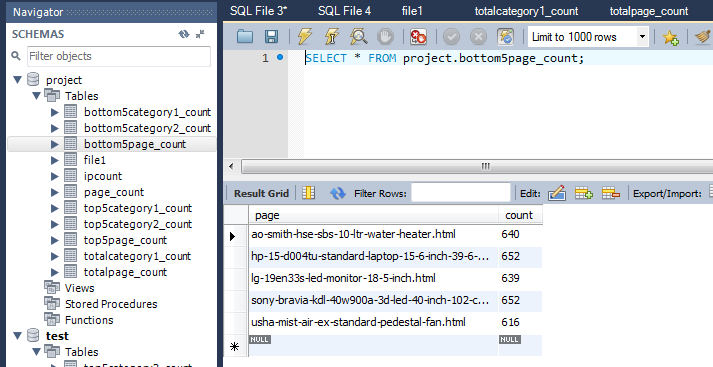


**BOTTOM 5 Page count:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table bottom5page\_count –export-dir /cloudera/project/hadoop/pig\_12 –username root –P –fields-terminated-by ‘,’

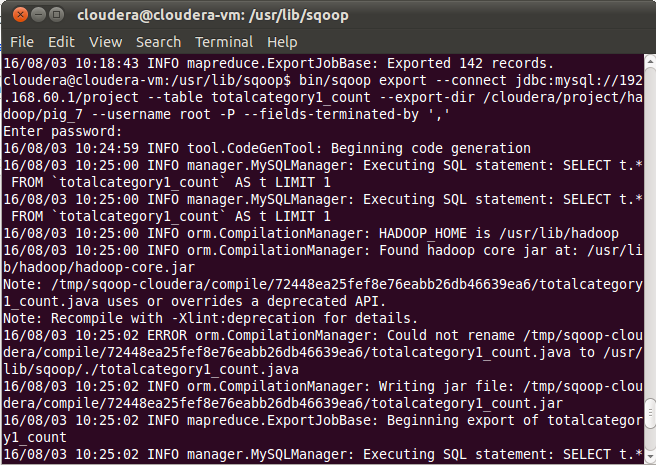


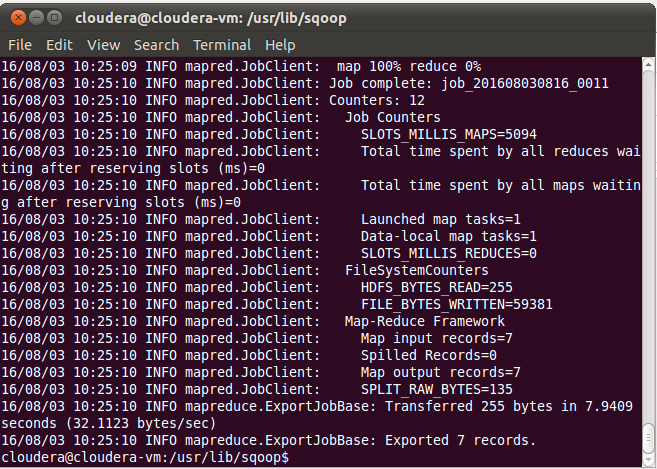


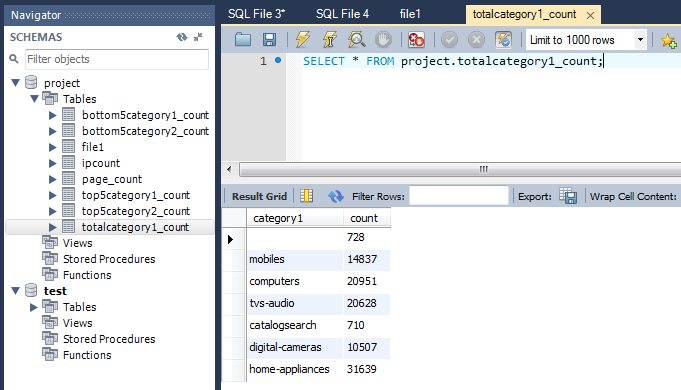


**Total category 1**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table totalcategory1\_count –export-dir /cloudera/project/hadoop/pig\_7 –username root –P –fields-terminated-by ‘,’

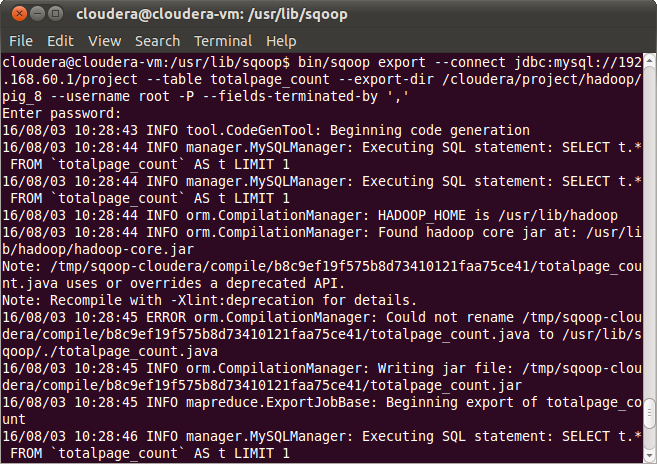


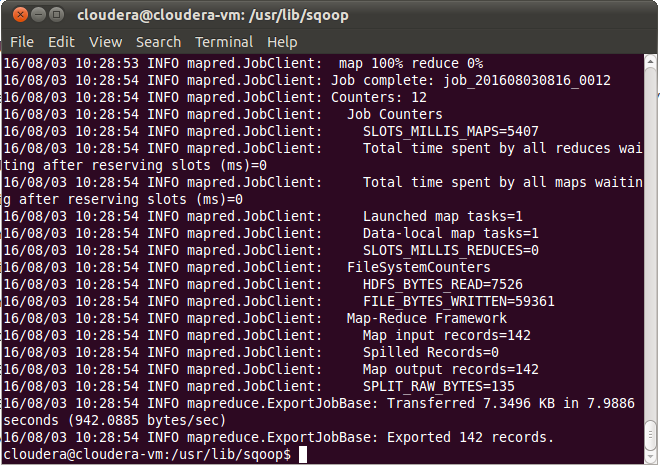


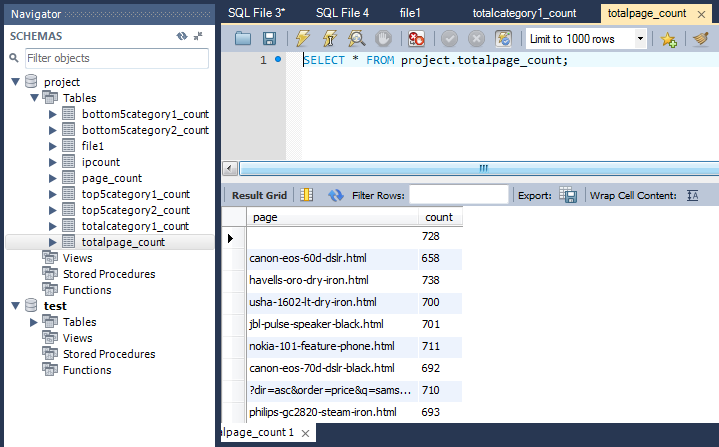


**Total page count**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table toatalpage\_count –export-dir /cloudera/project/hadoop/pig\_8 –username root –P –fields-terminated-by ‘,’

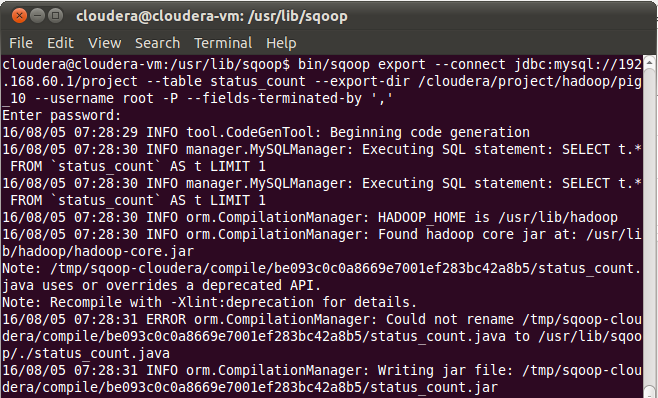


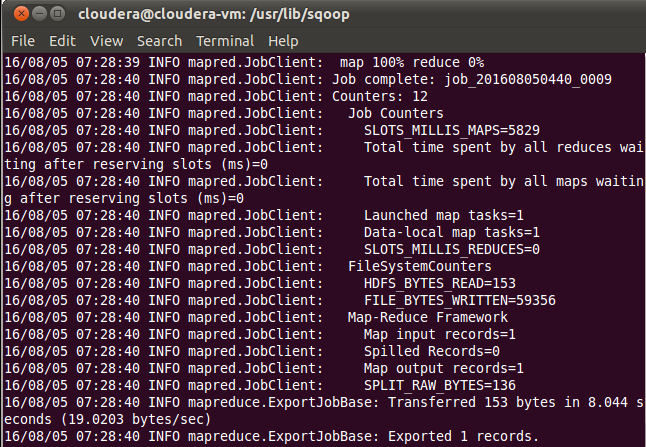


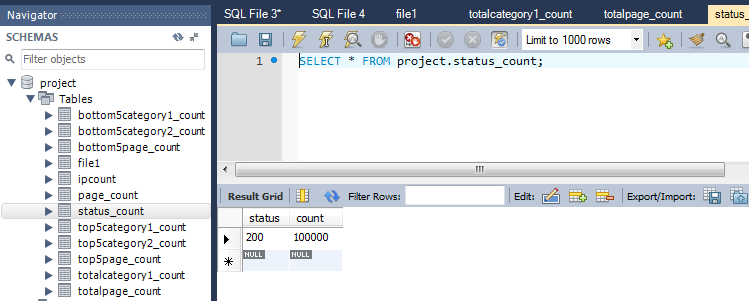


**Status code count:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table status\_count –export-dir /cloudera/project/hadoop/pig\_10 –username root –P –fields-terminated-by ‘,’

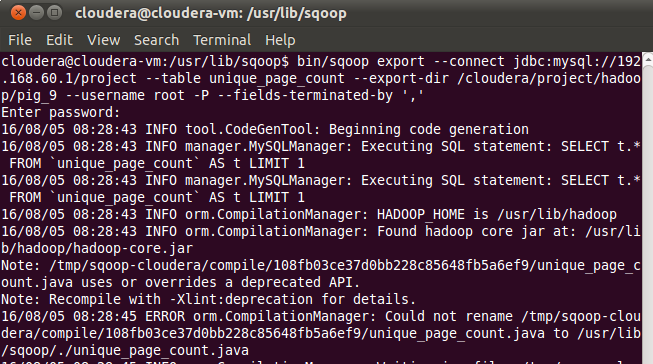


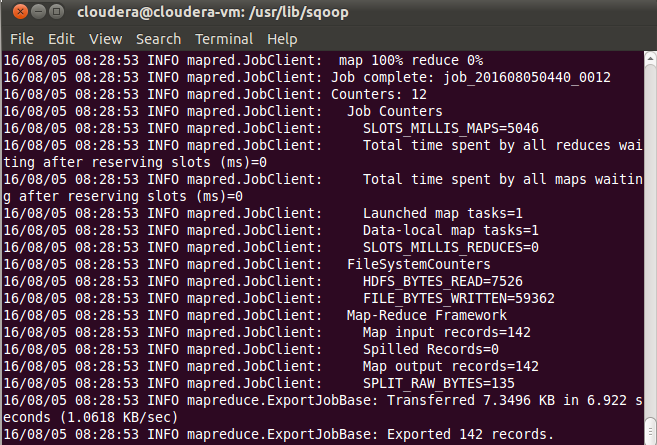


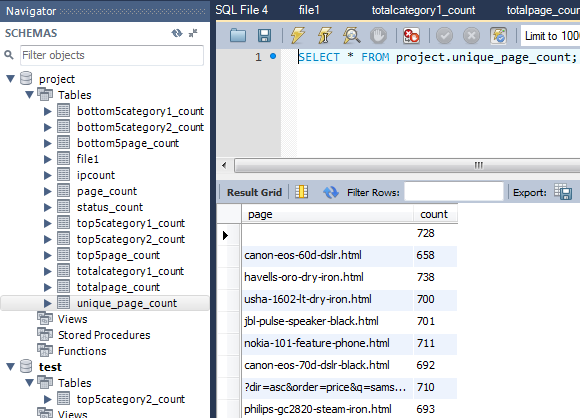


**Unique page count:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table unique\_page\_count –export-dir /cloudera/project/hadoop/pig\_9 –username root –P –fields-terminated-by ‘,’







**Total Category2 count:-**

bin/sqoop export –connect jdbc:mysql://192.168.60.1/project –table totalcategory2\_count –export-dir /cloudera/project/hadoop/pig\_11 –username root –P –fields-terminated-by ‘,’

