UNDERSTANDING HADOOP BY MAHESH MAHARANA

Saturday, January 7, 2017

HADOOP (PROOF OF CONCEPT) RETAIL DATA BY MAHESH CHANDRA MAHARANA

INDUSTRY: RETAIL
Data Input Format :- .xls (My Input Data is in excel 2007-2003 Format)

Kindly check my blog to read any kind of Excel sheet and use the Excel Input format, record reader and excel parser given in that blog. Please find link to my blog below:

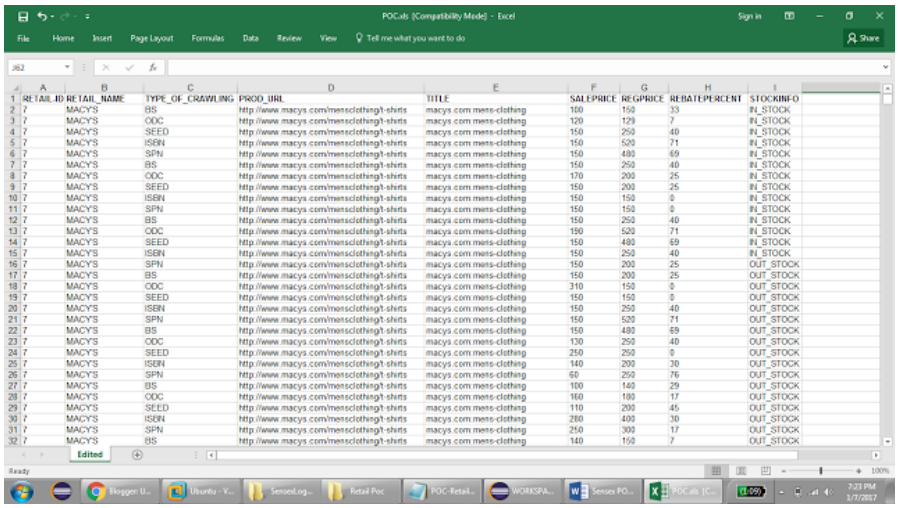
https://hadoop-poc-mahesh.blogspot.in/2017/01/hadoop-excel-input-format-to-read-any.html

This POC Input file and Problem statement was shared to me by Mr. Sunil Pashikanti like this below was created 3000 records:-

ATTRIBUTES are like:-
1. RETAIL_ID
2. RETAIL_NAME
3. TYPE_OF_CRAWLING
4. PRODUCT_URL
5. TITTLE
6. SALE_PRICE
7. REG_PRICE
8. REBATE_PERCENTAGE
9. STOCK_INFO

Example:

12 Amazon BS http://www.amazon.com/dell/lp Amazon.com:Dell Laptop 100.00 150.00 33 InStock



RETAIL_ID	RETAIL_NAME	TYPE_OF_CRAWLING	PROD_URL	TITLE	SALEPRICE	REGPRICE	REBATEPERCENT	STOCKINFO
7	MACY'S	BS	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	100	150	33	IN_STOCK
7	MACY'S	ODC	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	120	129	7	IN_STOCK
7	MACY'S	SEED	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	250	40	IN_STOCK
7	MACY'S	ISBN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	520	71	IN_STOCK
7	MACY'S	SPN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	480	69	IN_STOCK
7	MACY'S	BS	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	250	40	IN_STOCK
7	MACY'S	ODC	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	170	200	25	IN_STOCK
7	MACY'S	SEED	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	200	25	IN_STOCK
7	MACY'S	ISBN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	150	0	IN_STOCK
7	MACY'S	SPN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	150	0	IN_STOCK
7	MACY'S	BS	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	250	40	IN_STOCK
7	MACY'S	ODC	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	190	520	71	IN_STOCK
7	MACY'S	SEED	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	480	69	IN_STOCK
7	MACY'S	ISBN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	250	40	IN_STOCK
7	MACY'S	SPN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	200	25	OUT_STOCK
7	MACY'S	BS	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	200	25	OUT_STOCK
7	MACY'S	ODC	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	310	150	0	OUT_STOCK
7	MACY'S	SEED	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	150	0	OUT_STOCK
7	MACY'S	ISBN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	250	40	OUT_STOCK
7	MACY'S	SPN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	520	71	OUT_STOCK
7	MACY'S	BS	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	150	480	69	OUT_STOCK
7	MACY'S	ODC	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	130	250	40	OUT_STOCK
7	MACY'S	SEED	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	250	250	0	OUT_STOCK
7	MACY'S	ISBN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	140	200	30	OUT_STOCK
7	MACY'S	SPN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	60	250	76	OUT_STOCK
7	MACY'S	BS	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	100	140	29	OUT_STOCK
7	MACY'S	ODC	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	160	180	17	OUT_STOCK
7	MACY'S	SEED	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	110	200	45	OUT_STOCK
7	MACY'S	ISBN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	280	400	30	OUT_STOCK
7	MACY'S	SPN	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	250	300	17	OUT_STOCK
7	MACY'S	BS	http://www.macys.com/mensclothing/t-shirts	macys.com:mens-clothing	140	150	7	OUT_STOCK

DOWNLOAD MY INPUT FILE FROM BELOW LINK:

https://drive.google.com/file/d/0BzYUKIo7aWL_Sm5mT2l1cnZSQ0E/view?usp=sharing

PROBLEM STATEMENT: -

1. take the complete Excel Input data on HDFS

2. Develop a Map Reduce Use Case to get the below filtered results from the HDFS Input data(Excel data)

```
IF Type_Of_Crawling is -->'BS'
    -salePrice < 100.00  & RebatePercent>50  --> store "HighBuzzProducts"
    -RegPrice<150.00 & RebatePercent in 25-50 --> store "NormalProducts"
    -lengthOf(title)>100 ---> 'rare products'

IF Type_Of_Crawling is -->'ODC'
    - salePrice < 150.00 --> store "OnDemandCrawlProducts"
    - StockInfo --> "InStock"  -->store "AvailableProducts"
ELSE
    store in "OtherProducts"
```

NOTE: In the mentioned file names only 5 outputs have to be generated

3. Develop a PIG Script to filter the Map Reduce Output in the below fashion
   - Provide the Unique data
   - Sort the Unique data based on RETAIL_ID in DESC order

4. EXPORT the same PIG Output from HDFS to MySQL using SQOOP

5. Store the same PIG Output in a HIVE External Table.

NOTE:- For this POC I have used custom input format to read EXCEL files using external jar. So the corresponding jar files to be added during coding and to the lib directory of hadoop for successful execution. You can use poi-xml jar for the reading .xlsx file (2010 onwards excel format).

Below is the steps to make it work...

1. Download and Install ant from below link.

http://muug.ca/mirror/apache-dist//ant/binaries/apache-ant-1.9.8-bin.tar.gz

2. To install give following command in terminal:

tar -xzvf <apache ant Path>

3. Update bashrc:-

nano ~/.bashrc

Add below two lines:-

```
export ANT_HOME=${ant_dir}

export PATH=${ANT_HOME}/bin
```

Now Source bashrc by command:
source ~/.bashrc

4. Then restart the system. (Very Important for the effect to take place)

5. Download the required Jar files from below link:

https://github.com/sreejithpillai/ExcelRecordReaderMapReduce/blob/master/target/ExcelRecordReaderMapReduce-0.0.1-SNAPSHOT-jar-with-dependencies.jar

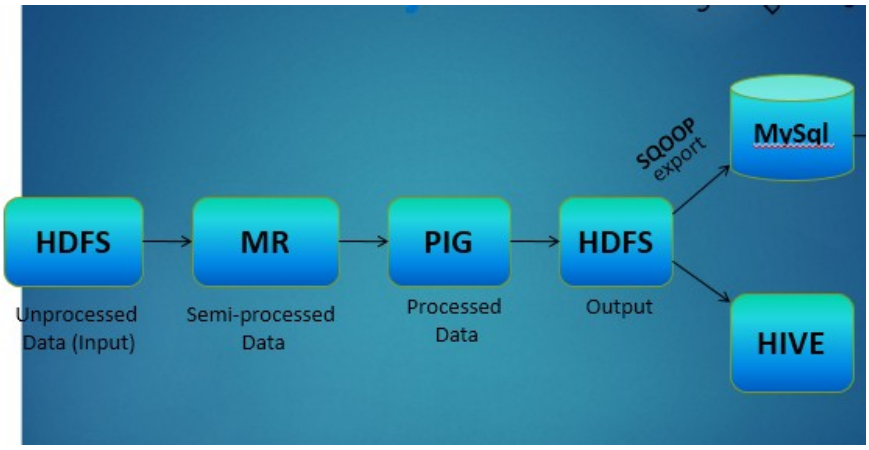
https://github.com/sreejithpillai/ExcelRecordReaderMapReduce/blob/master/target/ExcelRecordReaderMapReduce-0.0.1-SNAPSHOT.jar

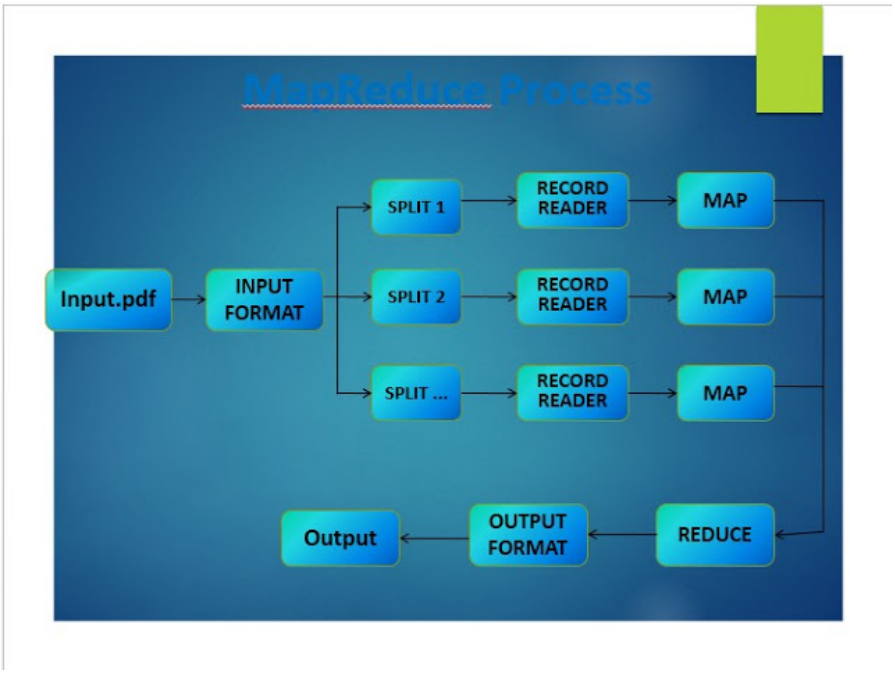
Place both jar files during Eclipse compilation and only SNAPSHOT.jar in hadoop lib directory.

6. If still not working try to add CLASSPATH:
export CLASSPATH=.:$CLASSPATH:<Path to the jar file 1>:<Path to jar file 2>

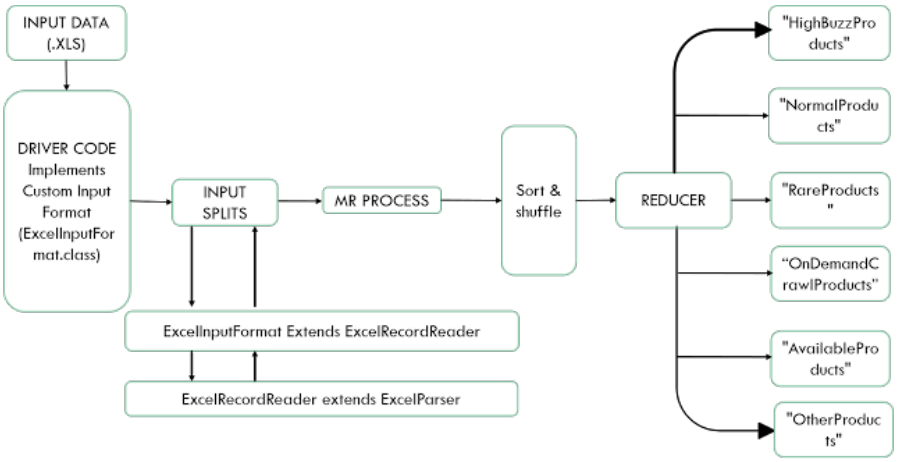
Hope it will work now.

POC Processing Details



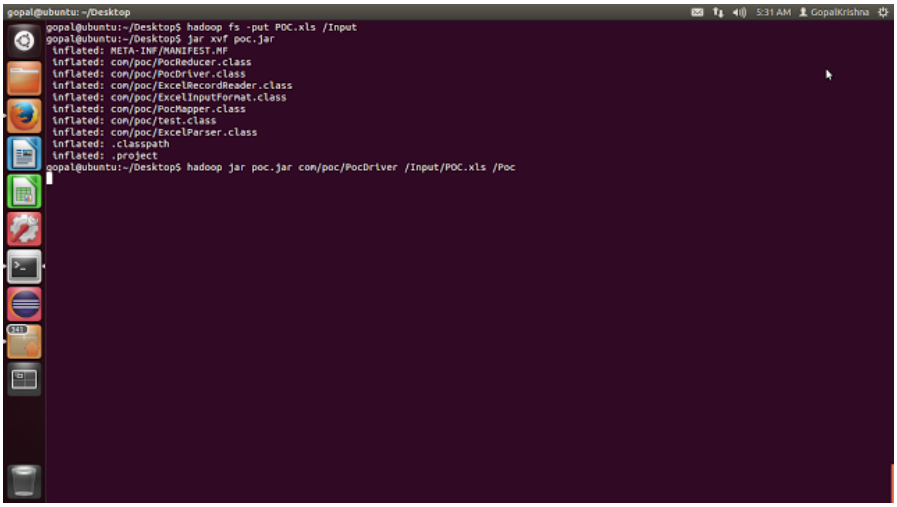


MAP REDUCE PROCESS IN DETAILS:-



1. TO TAKE XLS INPUT DATA ON HDFS

hadoop fs -mkdir /Input
hadoop fs -put POC.xls /Input
jar xvf poc.jar



2. MAP REDUCE CODES:-

EXCEL INPUT DRIVER
(DRIVER CLASS)

```
package com.poc;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
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.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.LazyOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class PocDriver {

 static public int count = 0;

 public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {
  Configuration conf = new Configuration();
```

```
  GenericOptionsParser parser = new GenericOptionsParser(conf, args);
  args = parser.getRemainingArgs();

  Job job = new Job(conf, "Retail_Poc");
  job.setJarByClass(PocDriver.class);

  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(Text.class);

  job.setInputFormatClass(ExcelInputFormat.class);
  job.setOutputFormatClass(TextOutputFormat.class);
  LazyOutputFormat.setOutputFormatClass(job, TextOutputFormat.class);

  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  // job.setNumReduceTasks(0);

  job.setMapperClass(PocMapper.class);
  job.setReducerClass(PocReducer.class);

  MultipleOutputs.addNamedOutput(job, "HighBuzzProducts", TextOutputFormat.class,
IntWritable.class, Text.class);
  MultipleOutputs.addNamedOutput(job, "NormalProducts", TextOutputFormat.class,
IntWritable.class, Text.class);
  MultipleOutputs.addNamedOutput(job, "RareProducts", TextOutputFormat.class, IntWritable.class,
Text.class);
  MultipleOutputs.addNamedOutput(job, "OnDemandCrawlProducts", TextOutputFormat.class,
IntWritable.class,
    Text.class);
  MultipleOutputs.addNamedOutput(job, "AvailableProducts", TextOutputFormat.class,
IntWritable.class, Text.class);
  MultipleOutputs.addNamedOutput(job, "OtherProducts", TextOutputFormat.class,
IntWritable.class, Text.class);

  System.exit(job.waitForCompletion(true) ? 0 : 1);

 }
}
```

EXCEL INPUT FORMAT
(CUSTOM INPUT FORMAT TO READ EXCEL FILES)

```
package com.poc;
import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.InputSplit;
import org.apache.hadoop.mapreduce.RecordReader;
import org.apache.hadoop.mapreduce.TaskAttemptContext;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

public class ExcelInputFormat extends FileInputFormat<LongWritable, Text> {
	@Override
	public RecordReader<LongWritable, Text> createRecordReader(InputSplit split, TaskAttemptContext
context)
				throws IOException, InterruptedException {
			return new ExcelRecordReader();
	}
}
```

EXCEL RECORD READER
(TO READ EXCEL FILE AND SEND AS KEY, VALUE FORMAT)

```
package com.poc;
import java.io.IOException;
import java.io.InputStream;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.InputSplit;
import org.apache.hadoop.mapreduce.RecordReader;
import org.apache.hadoop.mapreduce.TaskAttemptContext;
import org.apache.hadoop.mapreduce.lib.input.FileSplit;
import com.sreejithpillai.excel.parser.ExcelParser;

public class ExcelRecordReader extends RecordReader<LongWritable, Text> {
	private LongWritable key;
```

```
	private Text value;
	private InputStream is;
	private String[] strArrayofLines;
	@Override
	public void initialize(InputSplit genericSplit, TaskAttemptContext context)
		throws IOException, InterruptedException {
		FileSplit split = (FileSplit) genericSplit;
		Configuration job = context.getConfiguration();
		final Path file = split.getPath();
		FileSystem fs = file.getFileSystem(job);
		FSDataInputStream fileIn = fs.open(split.getPath());
		is = fileIn;
		String line = new ExcelParser().parseExcelData(is);
		this.strArrayofLines = line.split("\n");
	}
	@Override
	public boolean nextKeyValue() throws IOException, InterruptedException {
		if (key == null) {
			key = new LongWritable(0);
			value = new Text(strArrayofLines[0]);
		} else {
			if (key.get() < (this.strArrayofLines.length - 1)) {
				long pos = (int) key.get();
				key.set(pos + 1);
				value.set(this.strArrayofLines[(int) (pos + 1)]);
				pos++;
			} else {
				return false;
			}
		}
		if (key == null || value == null) {
			return false;
		} else {
			return true;
		}
	}
	@Override
	public LongWritable getCurrentKey() throws IOException, InterruptedException {
		return key;
	}
	@Override
	public Text getCurrentValue() throws IOException, InterruptedException {
		return value;
	}
	@Override
	public float getProgress() throws IOException, InterruptedException {
		return 0;
	}
	@Override
	public void close() throws IOException {
		if (is != null) {
			is.close();
		}
	}
}
```

EXCEL PARSER
(TO PARSE EXCEL SHEET)

```
package com.poc;
import java.io.IOException;
import java.io.InputStream;
import java.util.Iterator;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;

public class ExcelParser {
	private static final Log LOG = LogFactory.getLog(ExcelParser.class);
	private StringBuilder currentString = null;
	private long bytesRead = 0;
	public String parseExcelData(InputStream is) {
		try {
```

```
                        HSSFWorkbook workbook = new HSSFWorkbook(is);
                        HSSFSheet sheet = workbook.getSheetAt(0);
                        Iterator<Row> rowIterator = sheet.iterator();
                        currentString = new StringBuilder();
                        while (rowIterator.hasNext()) {
                                Row row = rowIterator.next();
                                Iterator<Cell> cellIterator = row.cellIterator();
                                while (cellIterator.hasNext()) {
                                        Cell cell = cellIterator.next();
                                        switch (cell.getCellType()) {
                                        case Cell.CELL_TYPE_BOOLEAN:
                                                bytesRead++;

currentString.append(cell.getBooleanCellValue() + "\t");
                                                break;
                                case Cell.CELL_TYPE_NUMERIC:
                                                bytesRead++;

currentString.append(cell.getNumericCellValue() + "\t");
                                                break;
                                        case Cell.CELL_TYPE_STRING:
                                                bytesRead++;
                                                currentString.append(cell.getStringCellValue()
+ "\t");
                                                break;
                                        }
                                }
                                currentString.append("\n");
                        }
                        is.close();
                } catch (IOException e) {
                        LOG.error("IO Exception : File not found " + e);
                }
                return currentString.toString();
        }
        public long getBytesRead() {
                return bytesRead;
        }
}
```

EXCEL MAPPER
(HAVING MAPPER LOGIC)

```
package com.poc;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class PocMapper extends Mapper<LongWritable, Text, Text, Text> {
 public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {
  try {
   if (value.toString().contains("RTL_NAME") && value.toString().contains("TYPE_OF_CRAWLING"))
    return;
   else {
    String[] str = value.toString().split(" ");
    String data = "";
    for (int i = 0; i < str.length; i++) {
     if (str[i] != null || str[i] != " ") {
      data += (str[i] + " ");

     }
    }
    String dr1 = data.trim().replaceAll("\\s+", "\t");
    String[] str1 = dr1.split("\t");

    int id = (int) Double.parseDouble(str1[0]);
    int regprice = (int) Double.parseDouble(str1[6]);
    int rebate = (int) Double.parseDouble(str1[7]);
    int saleprice = (int) Double.parseDouble(str1[5]);
    String dr = Integer.toString(id) + "\t" + str1[1] + "\t" + str1[2] + "\t" + str1[3] + "\t" + str1[4]+ "\t" +
Integer.toString(saleprice) + "\t" + Integer.toString(regprice) + "\t"+Integer.toString(rebate) + "\t" +
```

```
str1[8];

    context.write(new Text(""), new Text(dr));
   }
  } catch (Exception e) {
   e.printStackTrace();
  }
 }
}
```

EXCEL REDUCER
(HAVING REDUCER LOGIC)

```
package com.poc;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs;

public class PocReducer extends Reducer<Text, Text, IntWritable, Text> {
 MultipleOutputs<IntWritable, Text> mos;

 @Override
 public void setup(Context context) {
  mos = new MultipleOutputs<IntWritable, Text>(context);
 }

 @Override
 public void reduce(Text k1, Iterable<Text> k2, Context context) throws IOException,
InterruptedException {
  while (k2.iterator().hasNext()) {
   String sr = k2.iterator().next().toString();
   String sr1 = sr.trim().replaceAll("\\s+", "\t");

   String[] str1 = sr1.split("\t");

   int regprice = Integer.parseInt(str1[6]);
   int rebate = Integer.parseInt(str1[7]);
   int saleprice = Integer.parseInt(str1[5]);
   String dr = str1[0] + "\t" + str1[1] + "\t" + str1[2] + "\t" + str1[3] + "\t" + str1[4] + "\t" + str1[5]
     + "\t" + str1[6] + "\t" + str1[7] + "\t" + str1[8];
   if (str1[2].equalsIgnoreCase("BS")) {
    if (saleprice < 100 && rebate > 50) {
     mos.write("HighBuzzProducts", null, new Text(dr), "/Retail/HighBuzzProducts");
    } else if (regprice < 150 && rebate > 25 && rebate < 50) {
     mos.write("NormalProducts", null, new Text(dr), "/Retail/NormalProducts");

    } else if (str1[4].length() > 100) {
     mos.write("RareProducts", null, new Text(dr), "/Retail/RareProducts");
    } else {
     mos.write("OtherProducts", null, new Text(dr), "/Retail/OtherProducts");
    }
   } else if (str1[2].equalsIgnoreCase("ODC")) {
    if (saleprice < 150) {
     mos.write("OnDemandCrawlProducts", null, new Text(dr), "/Retail/OnDemandCrawlProducts");
    } else if (str1[8].equalsIgnoreCase("IN_STOCK")) {
     mos.write("AvailableProducts", null, new Text(dr), "/Retail/AvailableProducts");
    } else {
     mos.write("OtherProducts", null, new Text(dr), "/Retail/OtherProducts");
    }
   } else {
    mos.write("OtherProducts", null, new Text(dr), "/Retail/OtherProducts");
   }

  }
 }

 @Override
 protected void cleanup(Context context) throws IOException, InterruptedException {
  mos.close();
 }
}
```

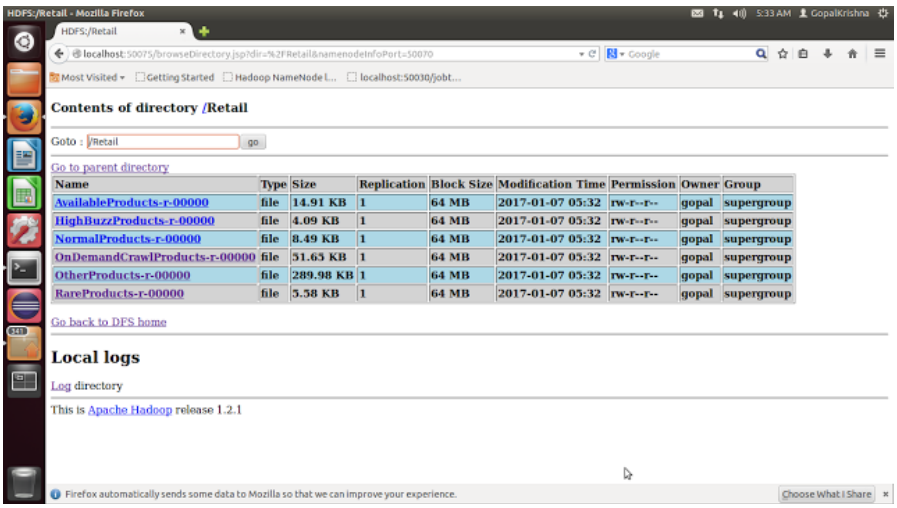
EXECUTING THE MAP REDUCE CODE

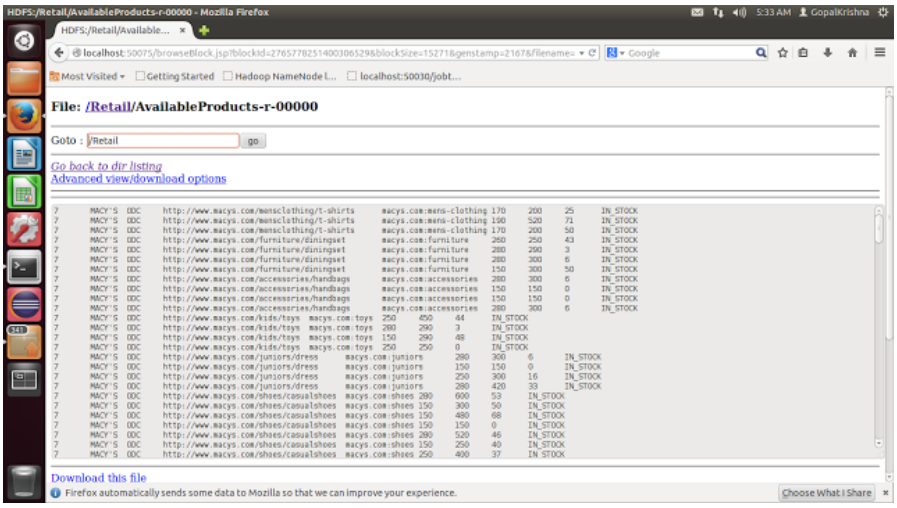
```
hadoop jar poc.jar com/poc/PocDriver /Input/POC.xls /Poc
```

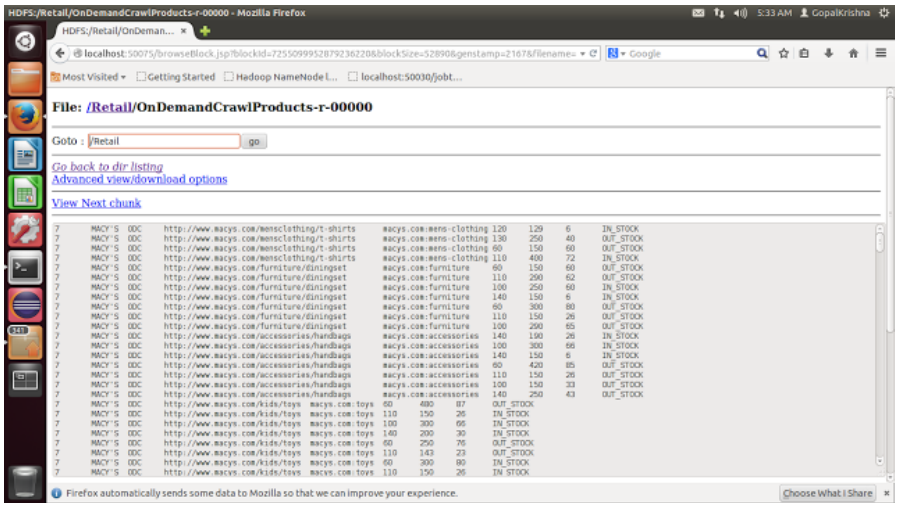
```
gopal@ubuntu: ~/Desktop
17/01/07 05:32:05 INFO mapred.JobClient:  map 0% reduce 0%
17/01/07 05:32:29 INFO mapred.JobClient:  map 100% reduce 0%
17/01/07 05:32:42 INFO mapred.JobClient:  map 100% reduce 33%
17/01/07 05:32:45 INFO mapred.JobClient:  map 100% reduce 100%
17/01/07 05:32:51 INFO mapred.JobClient: Job complete: job_201701070222_0024
17/01/07 05:32:51 INFO mapred.JobClient: Counters: 29
17/01/07 05:32:51 INFO mapred.JobClient:   Job Counters
17/01/07 05:32:51 INFO mapred.JobClient:     Launched reduce tasks=1
17/01/07 05:32:51 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=29702
17/01/07 05:32:51 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
17/01/07 05:32:51 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
17/01/07 05:32:51 INFO mapred.JobClient:     Launched map tasks=1
17/01/07 05:32:51 INFO mapred.JobClient:     Data-local map tasks=1
17/01/07 05:32:51 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=15760
17/01/07 05:32:51 INFO mapred.JobClient:   File Output Format Counters
17/01/07 05:32:51 INFO mapred.JobClient:     Bytes Written=0
17/01/07 05:32:51 INFO mapred.JobClient:   FileSystemCounters
17/01/07 05:32:51 INFO mapred.JobClient:     FILE_BYTES_READ=395930
17/01/07 05:32:51 INFO mapred.JobClient:     HDFS_BYTES_READ=567397
17/01/07 05:32:51 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=911499
17/01/07 05:32:51 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=383694
17/01/07 05:32:51 INFO mapred.JobClient:   File Input Format Counters
17/01/07 05:32:51 INFO mapred.JobClient:     Bytes Read=567296
17/01/07 05:32:51 INFO mapred.JobClient:   Map-Reduce Framework
17/01/07 05:32:51 INFO mapred.JobClient:     Map output materialized bytes=395930
17/01/07 05:32:51 INFO mapred.JobClient:     Map input records=3801
17/01/07 05:32:51 INFO mapred.JobClient:     Reduce shuffle bytes=395930
17/01/07 05:32:51 INFO mapred.JobClient:     Spilled Records=7600
17/01/07 05:32:51 INFO mapred.JobClient:     Map output bytes=387909
17/01/07 05:32:51 INFO mapred.JobClient:     Total committed heap usage (bytes)=239337472
17/01/07 05:32:51 INFO mapred.JobClient:     CPU time spent (ms)=28410
17/01/07 05:32:51 INFO mapred.JobClient:     Combine input records=0
17/01/07 05:32:51 INFO mapred.JobClient:     SPLIT_RAW_BYTES=101
17/01/07 05:32:51 INFO mapred.JobClient:     Reduce input records=3800
17/01/07 05:32:51 INFO mapred.JobClient:     Reduce input groups=1
17/01/07 05:32:51 INFO mapred.JobClient:     Combine output records=0
17/01/07 05:32:51 INFO mapred.JobClient:     Physical memory (bytes) snapshot=281661440
17/01/07 05:32:51 INFO mapred.JobClient:     Reduce output records=0
17/01/07 05:32:51 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=935030784
17/01/07 05:32:51 INFO mapred.JobClient:     Map output records=3800
gopal@ubuntu:~/Desktop$
```

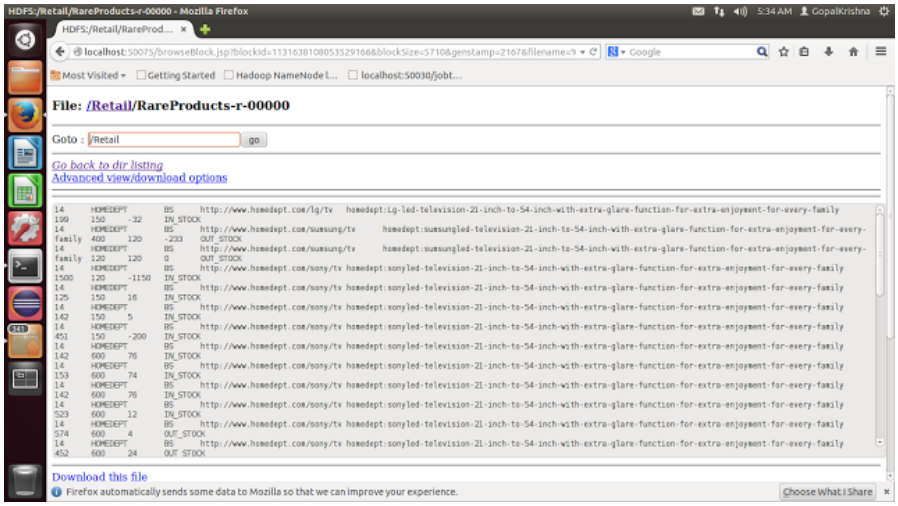
Goto Firefox and open name node page by following command:

http://localhost:50070 and browse the file system , then click on HealthCarePOC directory to check the files created.



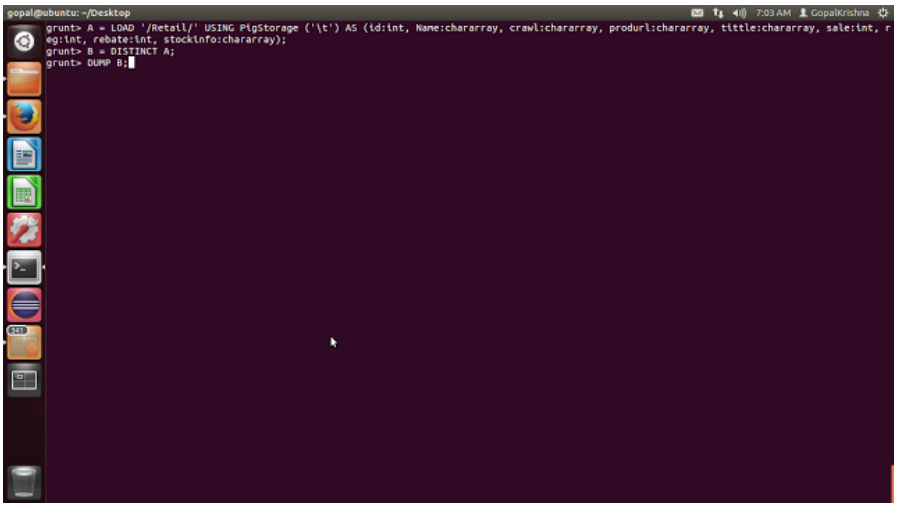


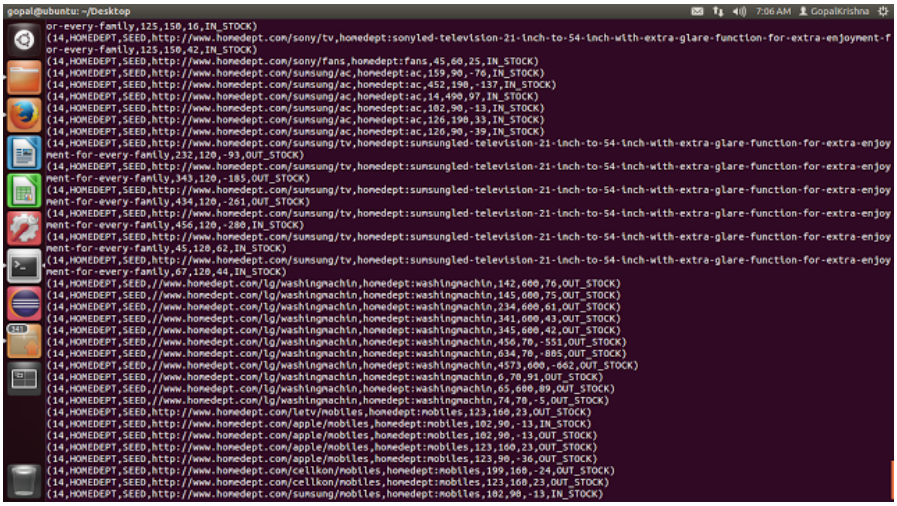




3. PIG SCRIPT

```
A = LOAD '/Retail/' USING PigStorage ('\t') AS (id:int, Name:chararray, crawl:chararray,
produrl:chararray, tittle:chararray, sale:int, reg:int, rebate:int, stockinfo:chararray);

B = DISTINCT A;
DUMP B;
```





PigScript2.pig

A = LOAD '/Retail/' USING PigStorage ('\t') AS (id:int, Name:chararray, crawl:chararray, produrl:chararray, tittle:chararray, sale:int, reg:int, rebate:int, stockinfo:chararray);

B = DISTINCT A;
C = ORDER B BY id;
STORE C INTO '/RETAILPOC';

```
gopal@ubuntu: ~/Desktop
grunt> A = LOAD '/Retail/' USING PigStorage ('\t') AS (id:int, Name:chararray, crawl:chararray, produrl:chararray, tittle:chararray, sale:int, r
eg:int, rebate:int, stockinfo:chararray);
grunt> B = DISTINCT A;
grunt> C = ORDER B BY id;
grunt> STORE C INTO '/RETAILPOC';
```

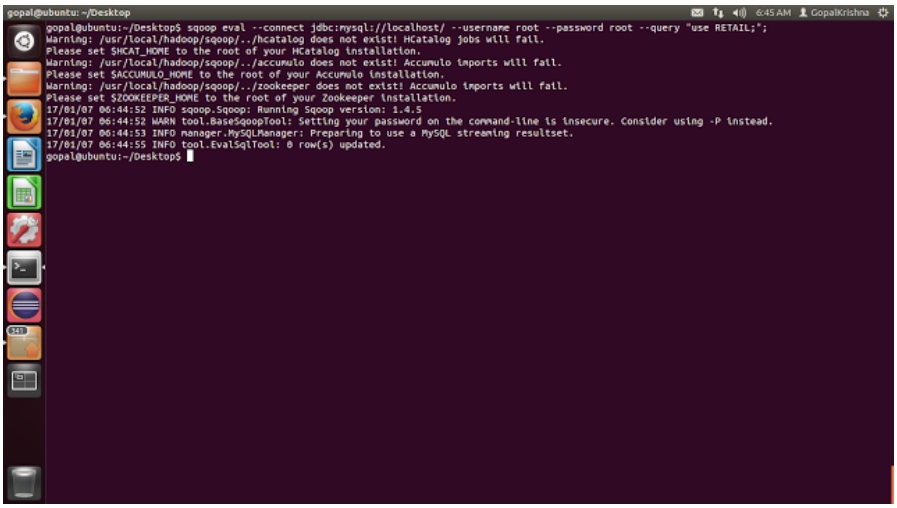
```
gopal@ubuntu: ~/Desktop
2017-01-07 07:09:39,492 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Running jobs are [job_2017
01070222_0033]
2017-01-07 07:10:01,384 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 99% complete
2017-01-07 07:10:01,384 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Running jobs are [job_2017
01070222_0033]
2017-01-07 07:10:15,627 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2017-01-07 07:10:15,682 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
1.2.1   0.14.0  gopal   2017-01-07 07:07:29     2017-01-07 07:10:15     ORDER_BY,DISTINCT

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianRe
ducetime        Alias   Feature Outputs
job_201701070222_0031   1       1       13      13      13      13      15      15      15      15      A       DISTINCT
job_201701070222_0032   1       1       10      10      10      10      17      17      17      17      C       SAMPLER
job_201701070222_0033   1       1       14      14      14      14      23      23      23      23      C       ORDER_BY        /RETAILPOC,

Input(s):
Successfully read 3808 records (384447 bytes) from: "/Retail"

Output(s):
Successfully stored 2985 records (309787 bytes) in: "/RETAILPOC"

Counters:
Total records written : 2985
Total bytes written : 309787
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_201701070222_0031   ->      job_201701070222_0032,
job_201701070222_0032   ->      job_201701070222_0033,
job_201701070222_0033


2017-01-07 07:10:15,940 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

4. EXPORT the PIG Output from HDFS to MySQL using SQOOP

sqoop eval --connect jdbc:mysql://localhost/ --username root --password root --query "create database if not exists RETAIL;";

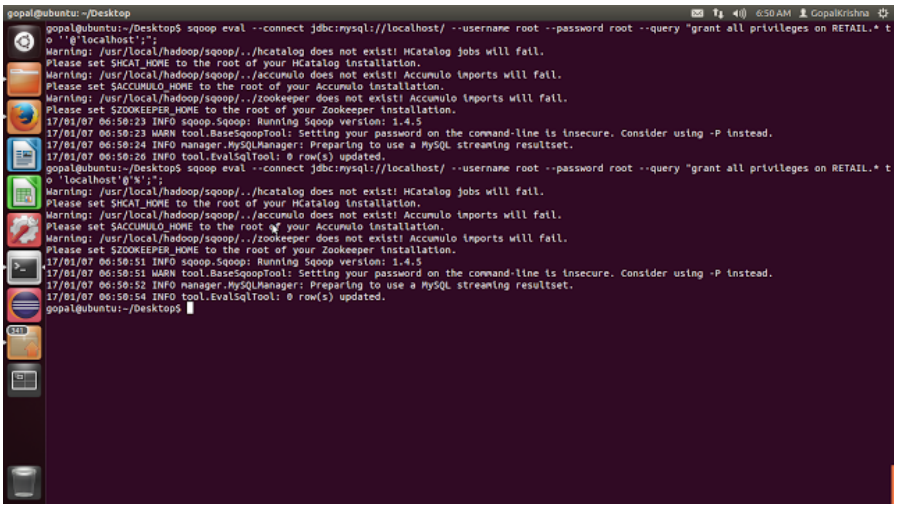
```
gopal@ubuntu: ~/Desktop
gopal@ubuntu:~/Desktop$ sqoop eval --connect jdbc:mysql://localhost/ --username root --password root --query "create database if not exists RETA
IL;";
Warning: /usr/local/hadoop/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/hadoop/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/hadoop/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
17/01/07 06:39:01 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5
17/01/07 06:39:01 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
17/01/07 06:39:02 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
17/01/07 06:39:03 INFO tool.EvalSqlTool: 1 row(s) updated.
gopal@ubuntu:~/Desktop$
```

sqoop eval --connect jdbc:mysql://localhost/ --username root --password root --query "use RETAIL;";



sqoop eval --connect jdbc:mysql://localhost/ --username root --password root --query "grant all privileges on RETAIL.* to 'localhost'@'%';";

sqoop eval --connect jdbc:mysql://localhost/ --username root --password root --query "grant all privileges on RETAIL.* to ''@'localhost';";



sqoop eval --connect jdbc:mysql://localhost/RETAIL --username root --password root --query "create table retailpoc(id int, name varchar(50), crawl varchar(50), produrl varchar(200), tittle varchar(200), sale int, reg int, rebate int, stockinfo varchar(50));";

```
gopal@ubuntu:~/Desktop
gopal@ubuntu:~/Desktop$ sqoop eval --connect jdbc:mysql://localhost/RETAIL --username root --password root --query "create table retailpoc(id in
t, name varchar(50), crawl varchar(50), produrl varchar(200), tittle varchar(200), sale int, reg int, rebate int, stockinfo varchar(50));";
Warning: /usr/local/hadoop/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/hadoop/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/hadoop/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
17/01/07 06:58:30 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5
17/01/07 06:58:30 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
17/01/07 06:58:31 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
17/01/07 06:58:33 INFO tool.EvalSqlTool: 0 row(s) updated.
gopal@ubuntu:~/Desktop$
```

sqoop export --connect jdbc:mysql://localhost/RETAIL--table retailpoc --export-dir /RETAILPOC --fields-terminated-by '\t';

```
gopal@ubuntu:~/Desktop
gopal@ubuntu:~/Desktop$ sqoop export --connect jdbc:mysql://localhost/RETAIL --table retailpoc --export-dir /RETAILPOC --fields-terminated-by '\
t';
Warning: /usr/local/hadoop/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/hadoop/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/hadoop/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
17/01/07 07:15:41 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5
17/01/07 07:15:42 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
17/01/07 07:15:42 INFO tool.CodeGenTool: Beginning code generation
17/01/07 07:15:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `retailpoc` AS t LIMIT 1
17/01/07 07:15:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `retailpoc` AS t LIMIT 1
17/01/07 07:15:44 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop
Note: /tmp/sqoop-gopal/compile/838b62d737070e32d73fc1b705059e83/retailpoc.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
17/01/07 07:15:59 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-gopal/compile/838b62d737070e32d73fc1b705059e83/retailpoc.jar
17/01/07 07:15:59 INFO mapreduce.ExportJobBase: Beginning export of retailpoc
17/01/07 07:16:05 INFO input.FileInputFormat: Total input paths to process : 1
17/01/07 07:16:05 INFO input.FileInputFormat: Total input paths to process : 1
17/01/07 07:16:05 INFO util.NativeCodeLoader: Loaded the native-hadoop library
17/01/07 07:16:05 WARN snappy.LoadSnappy: Snappy native library not loaded
17/01/07 07:16:05 INFO mapred.JobClient: Running job: job_201701070222_0034
17/01/07 07:16:06 INFO mapred.JobClient:  map 0% reduce 0%
```

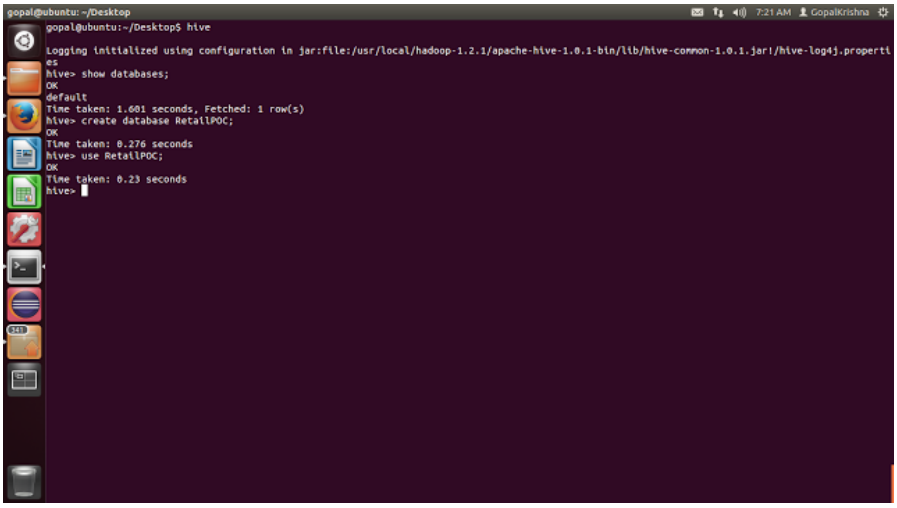
```
gopal@ubuntu:~/Desktop
Note: /tmp/sqoop-gopal/compile/838b62d737070e32d73fc1b705059e83/retailpoc.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
17/01/07 07:15:59 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-gopal/compile/838b62d737070e32d73fc1b705059e83/retailpoc.jar
17/01/07 07:15:59 INFO mapreduce.ExportJobBase: Beginning export of retailpoc
17/01/07 07:16:05 INFO input.FileInputFormat: Total input paths to process : 1
17/01/07 07:16:05 INFO input.FileInputFormat: Total input paths to process : 1
17/01/07 07:16:05 INFO util.NativeCodeLoader: Loaded the native-hadoop library
17/01/07 07:16:05 WARN snappy.LoadSnappy: Snappy native library not loaded
17/01/07 07:16:05 INFO mapred.JobClient: Running job: job_201701070222_0034
17/01/07 07:16:06 INFO mapred.JobClient:  map 0% reduce 0%
17/01/07 07:16:44 INFO mapred.JobClient:  map 50% reduce 0%
17/01/07 07:17:03 INFO mapred.JobClient:  map 75% reduce 0%
17/01/07 07:17:04 INFO mapred.JobClient:  map 100% reduce 0%
17/01/07 07:17:10 INFO mapred.JobClient: Job complete: job_201701070222_0034
17/01/07 07:17:10 INFO mapred.JobClient: Counters: 18
17/01/07 07:17:10 INFO mapred.JobClient:   Job Counters
17/01/07 07:17:10 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=93227
17/01/07 07:17:10 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
17/01/07 07:17:10 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
17/01/07 07:17:10 INFO mapred.JobClient:     Launched map tasks=4
17/01/07 07:17:10 INFO mapred.JobClient:     Data-local map tasks=4
17/01/07 07:17:10 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
17/01/07 07:17:10 INFO mapred.JobClient:   File Output Format Counters
17/01/07 07:17:10 INFO mapred.JobClient:     Bytes Written=0
17/01/07 07:17:10 INFO mapred.JobClient:   FileSystemCounters
17/01/07 07:17:10 INFO mapred.JobClient:     HDFS_BYTES_READ=313738
17/01/07 07:17:10 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=262812
17/01/07 07:17:10 INFO mapred.JobClient:   File Input Format Counters
17/01/07 07:17:10 INFO mapred.JobClient:     Bytes Read=0
17/01/07 07:17:10 INFO mapred.JobClient:   Map-Reduce Framework
17/01/07 07:17:10 INFO mapred.JobClient:     Map input records=2985
17/01/07 07:17:10 INFO mapred.JobClient:     Physical memory (bytes) snapshot=344236032
17/01/07 07:17:10 INFO mapred.JobClient:     Spilled Records=0
17/01/07 07:17:10 INFO mapred.JobClient:     CPU time spent (ms)=25910
17/01/07 07:17:10 INFO mapred.JobClient:     Total committed heap usage (bytes)=320864256
17/01/07 07:17:10 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1862782976
17/01/07 07:17:10 INFO mapred.JobClient:     Map output records=2985
17/01/07 07:17:10 INFO mapred.JobClient:     SPLIT_RAW_BYTES=566
17/01/07 07:17:10 INFO mapreduce.ExportJobBase: Transferred 306.3848 KB in 68.0053 seconds (4.5053 KB/sec)
17/01/07 07:17:10 INFO mapreduce.ExportJobBase: Exported 2985 records.
gopal@ubuntu:~/Desktop$
```

5. STORE THE PIG OUTPUT IN A HIVE EXTERNAL TABLE

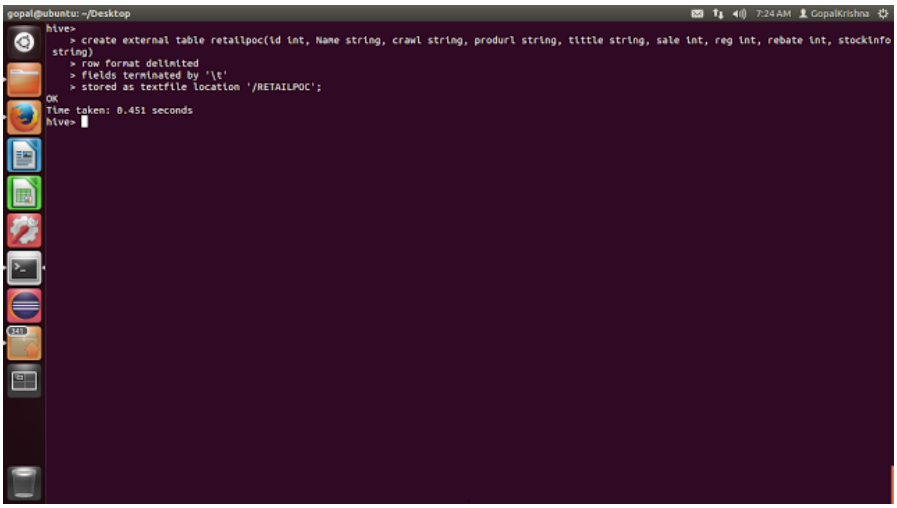
goto hive shell using command:

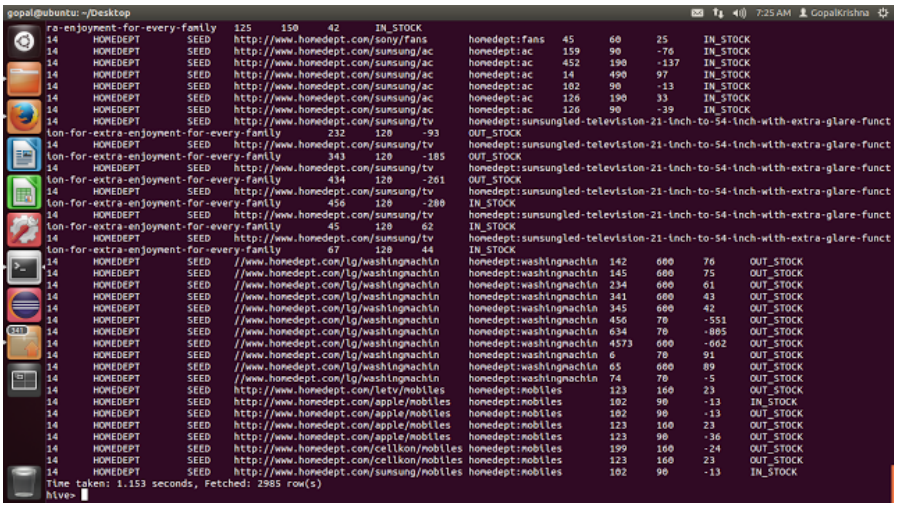
hive

show databases;
create database RetailPOC;
use RetailPOC;

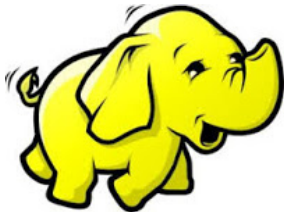


create external table retailpoc(id int, Name string, crawl string, produrl string, tittle string,
sale int, reg int, rebate int, stockinfo string)
row format delimited
fields terminated by '\t'
stored as textfile location '/RETAILPOC';





Hope you all understood the procedures...
Please do notify me for any corrections...
Kindly leave a comment for any queries/clarification...
(Detailed Description of each phase to be added soon).
ALL D BEST...



Posted by Mahesh Maharana at 7:28 AM

Labels: EXCEL DATA, Hadoop Excel Input Format, HADOOP POC, HIVE, PIG, Retail Data, SQOOP

19 comments

Add a comment as Shanmugam Ekambaram

Top comments

Shailendra Singh 4 days ago - Shared publicly
The Blog Content is very informative and helpful. Please share more content. Thanks.
http://aptrongurgaon.in/best-hadoop-training-in-gurgaon.html

1 · Reply

Andrew Son 1 week ago - Shared publicly
Hi admin,
Your blog gives a useful information.Share more like this.
https://www.fita.in/data-science-course-in-chennai/

1 · Reply

gracy layla 2 weeks ago - Shared publicly
Hi,
It is good article to understand Mapreduce concept. Thank u very much.
https://mindmajix.com/mapreduce-training

1 · Reply

Daniel Charlie 10 months ago - Shared publicly
nice blog. thanks for sharing Hadoop Tutorials. It's really good. Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment…
Keep sharing on Updated Tutorials????????

1 · Reply

Naveen Reddy 10 months ago - Shared publicly
great job brother.

1 · Reply

Raj Kamal 1 month ago - Shared publicly
TIB Academy is one of the best Hadoop Training Institute in Bangalore. We Offers Hands-On Training with Live project.

1 · Reply

suganthi tib academy 1 month ago - Shared publicly
Thanks to share..very useful blog.kindly share this type of blog.Hadoop is one of the most popular and demanding and open source data analytics technology that can be effectively applied to manage Big Data problems and achieve data

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Total Pageviews

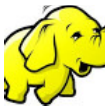
~~17187~~

Subscribe To

Posts

Comments

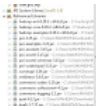
Popular Posts

- 
  HIVE 2.1.1 INSTALLATION IN HADOOP 2.7.3 IN UBUNTU 16
  Hello Friends, Welcome to the blog where I am going to explain and take you through the installation procedures of Hive 2.1.1 on Hadoo...
- 
  XML FILE PROCESSING IN HADOOP
  Dear Friends, Welcome back, after a long time. I was asked by one of my friend to explain about XML processing in hadoop. I went t...
- 
  HADOOP 2.X MULTI-NODE CLUSTER SETUP IN UBUNTU
  MULTI-NODE CLUSTER SETUP In this tutorial, I will describe the required steps for setting up a distributed, multi-node Hadoop 2.7.3 clu...
- 
  HADOOP 2.7.3 SINGLE NODE CLUSTER SETUP IN UBUNTU 16
  Setting up a Apache Hadoop 2.7.3 single node on Ubuntu 16 Day by day new & advance technology are being developed and we always lo...
- 
  HADOOP (PROOF OF CONCEPT) RETAIL DATA BY MAHESH CHANDRA MAHARANA
  INDUSTRY : RETAIL Data Input Format :- . xls (My Input Data is in excel 2007-2003 Format) Kindly check my blog to read any kind of ...
- 
  HIVE INTERVIEW RELATED PREPARATION
  Dear Friends.... Few days I spent preparing and giving interviews for job change in HADOOP and few HIVE questions were like most commo...
- 
  WAYS TO BULK LOAD DATA IN HBASE
  Dear Friends, Going ahead with my post, this one was asked by one of my friend about HBase, for which I am sharing my thoughts and work...
- 
  HADOOP POC ON EXCEL DATA WEATHER REPORT ANALYSIS
  Hello Friends, Glad to present this blog which is for analysis of Weather Report POC, which is in Excel Format. This POC was given to ...

Subscribe To

Posts

Comments

- 
  HIVE ON RESCUE- A HEALTHCARE USE_CASE ON STRUCTURED DATA
  Dear Friends, We know that Hadoop's HIVE component is very good for structured data processing. Structured data first depends ...
- 
  A USECASE ON TRAVEL APP
  Dear Friends, Welcome Back.... Day by Day I am learning different thing which I like to share with you all. As a great person s...