Flume command to copy 'Consumer\_Complaints.csv' file into HDFS at path "/projectusa/":

**flume-ng agent --conf-file /home/cloudera/Desktop/Shared/ProjectUsa/filecopy.conf --name agent1 -Dflume.root.logger=INFO,console**

===============================================================================

Solution for Problem 1:

Pig commands to analyze and store output in path: '/projectusa/output/prb1\_output':

**compl\_rel = load '/projectusa/Consumer\_Complaints.csv' USING PigStorage('\t') AS (date\_rec:chararray,product:chararray,sub\_product:chararray,issue:chararray,sub\_issue:chararray,complain\_nar:chararray,pub\_response:chararray,company:chararray,state:chararray,zip:chararray,channel:chararray,date\_sent:chararray,com\_res:chararray,timely\_resp:chararray,consumer\_disputed:chararray,complain\_id:long);**

**timely\_res\_data = filter compl\_rel by timely\_resp == 'Yes';**

**group\_timelyResData = group timely\_res\_data all;**

**count\_timelyResData = foreach group\_timelyResData generate COUNT(timely\_res\_data);**

**STORE count\_timelyResData INTO '/projectusa/output/prb1\_output';**

MySql table creation:

**create table timelyResComplaintsCount(countComplaints int);**

sqoop command to export data from HDFS path "/projectusa/output/prb1\_output/part-r-00000" to MySql table "timelyResComplaintsCount":

**sqoop export --connect jdbc:mysql://localhost/consumer\_complaints --username root --password cloudera --table timelyResComplaintsCount -m 1 --export-dir /projectusa/output/prb1\_output/part-r-00000**

==================================================================================Solution for Problem 2:

Pig commands to analyze and store output in path: '/projectusa/output/prb2\_output':

**register /home/cloudera/Desktop/Shared/ProjectUsa/checkSameDate.jar**

**compl\_rel = load '/projectusa/Consumer\_Complaints.csv' USING PigStorage('\t') AS (date\_rec:chararray,product:chararray,sub\_product:chararray,issue:chararray,sub\_issue:chararray,complain\_nar:chararray,pub\_response:chararray,company:chararray,state:chararray,zip:chararray,channel:chararray,date\_sent:chararray,com\_res:chararray,timely\_resp:chararray,consumer\_disputed:chararray,complain\_id:long);**

**rel\_date = foreach compl\_rel generate udf\_proj.DateComp(date\_rec,date\_sent);**

**filter\_val = filter rel\_date by $0 == 'Same Day';**

**grp\_all = group filter\_val all;**

**count\_same\_day\_fwd = foreach grp\_all generate COUNT(filter\_val);**

**STORE count\_same\_day\_fwd INTO '/projectusa/output/prb2\_output';**

MySql table creation:

**create table complaintsSameDayForwardCount(countComplaints int);**

sqoop command to export data from HDFS path "/projectusa/output/prb2\_output/part-r-00000" to MySql table "complaintsSameDayForwardCount":

**sqoop export --connect jdbc:mysql://localhost/consumer\_complaints --username root --password cloudera --table complaintsSameDayForwardCount -m 1 --export-dir /projectusa/output/prb2\_output/part-r-00000**

==================================================================================Solution for Problem 3:

Pig commands to analyze and store output in path: '/projectusa/output/prb3\_output':

**compl\_rel = load '/projectusa/Consumer\_Complaints.csv' USING PigStorage('\t') AS (date\_rec:chararray,product:chararray,sub\_product:chararray,issue:chararray,sub\_issue:chararray,complain\_nar:chararray,pub\_response:chararray,company:chararray,state:chararray,zip:chararray,channel:chararray,date\_sent:chararray,com\_res:chararray,timely\_resp:chararray,consumer\_disputed:chararray,complain\_id:long);**

**grp\_by\_company = group compl\_rel by company;**

**count\_complaints = foreach grp\_by\_company generate group as comp, COUNT(compl\_rel) as cnt;**

**order\_companiesByComplaints = order count\_complaints by cnt DESC;**

**topCompaniesWithComplaints = limit order\_companiesByComplaints 10;**

**STORE topCompaniesWithComplaints INTO '/projectusa/output/prb3\_output';**

MySql table creation:

**create table topCompaniesWithComplaints(company varchar(30),countComplaints int);**

sqoop command to export data from HDFS path "/projectusa/output/prb3\_output/part-r-00000" to MySql table "topCompaniesWithComplaints":

**sqoop export --connect jdbc:mysql://localhost/consumer\_complaints --username root --password cloudera --table topCompaniesWithComplaints -m 1 --export-dir /projectusa/output/prb3\_output/part-r-00000 --input-fields-terminated-by '\t'**

**==================================================================================**Solution for Problem 4:

Pig commands to analyze and store output in path: '/projectusa/output/prb4\_output':

**compl\_rel = load '/projectusa/Consumer\_Complaints.csv' USING PigStorage('\t') AS (date\_rec:chararray,product:chararray,sub\_product:chararray,issue:chararray,sub\_issue:chararray,complain\_nar:chararray,pub\_response:chararray,company:chararray,state:chararray,zip:chararray,channel:chararray,date\_sent:chararray,com\_res:chararray,timely\_resp:chararray,consumer\_disputed:chararray,complain\_id:long);**

**product\_type\_details = foreach compl\_rel generate ToDate(date\_rec, 'MM/dd/yyyy') as (date\_received:DateTime), product;**

**parse\_year = foreach product\_type\_details generate GetYear(date\_received) as year, product as prd;**

**filter\_year = filter parse\_year by year == 2015 AND prd == 'Debt collection';**

**group\_year\_product = group filter\_year by (year, prd);**

**count\_complaints = foreach group\_year\_product generate FLATTEN(group), COUNT(filter\_year);**

**STORE count\_complaints INTO '/projectusa/output/prb4\_output';**

MySql table creation:

**create table debtCollectionProductComplaints(year int,product varchar(30), countComplaints int);**

sqoop command to export data from HDFS path "/projectusa/output/prb4\_output/part-r-00000" to MySql table "debtCollectionProductComplaints":

**sqoop export --connect jdbc:mysql://localhost/consumer\_complaints --username root --password cloudera --table debtCollectionProductComplaints -m 1 --export-dir /projectusa/output/prb4\_output/part-r-00000 --input-fields-terminated-by '\t'**