**Project 2**

**Consumer complaint Analysis**

**Data Description:**

The dataset consists of consumer complaints details. In the process consumer files the complaints to consumer forum and then consumer forum forwards the complaint to respective company.

It contains the following fields Date received, Product, Sub-product, Issue, Sub-issue, Consumer complaint narrative, Company public response, Company, State, ZIP code, Submitted via, Date sent to company, Company response to consumer, Timely response?, Consumer disputed?, Complaint ID,

**Sample data:**

10/16/2015, Debt collection, "Other (phone, health club, etc.)", Cont'd attempts collect debt not allowed, Debt was discharged in bankruptcy" Convergent Resources, Inc.",OH,438XX,Web,10/16/2015,Closed with explanation, yes, 1612132.

This entire column "Other (phone, health club, etc.)"

Should be product but, if we split this file based on comma then this column will be split into 3 columns which will result in wrong outputs. In order to tackle this we should remove the commas present only inside double quotes.

**MAP REDUCE TO REMOVE ALL COMMAS WITHIN DOUBLE QUOTES**

**DRIVER CLASS:**

**Line1 to Line 14:** Importing the necessary jar file.

**Line 18:** Declaring the Configuration object for the job.

**Line19:** Declaring the Job for the first task.

**Line 20:** Setting jar by class.

**Line 21 to Line 25:** Setting mapOutput Key value pairs and Setting reducer Output key value pairs.

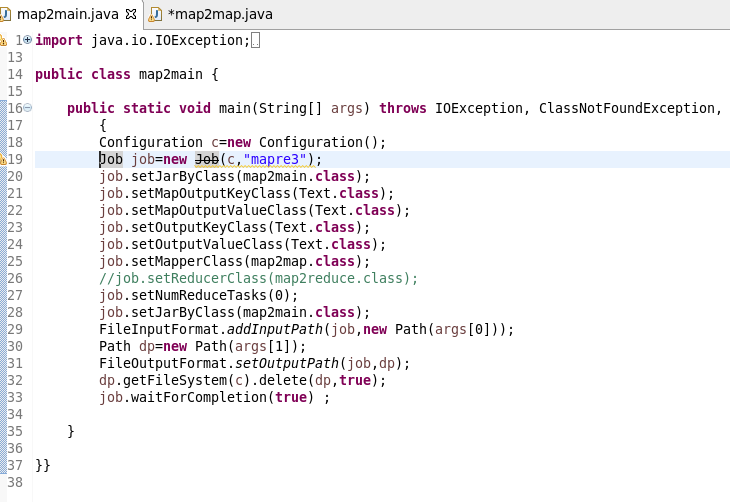
**Line 26:** Setting Mapper class.

**Line 27:** As it is a map only job setting setnum reduce tasks as”0”.

**Line 29:** Setting the input path from which the data will be processed.

**Line 31:** Setting the Output path.

**Line 33:** Job wait for completion as true.



**MAPPER CLASS:**

**LOGIC:** I WILL SET A BOOLEAN AS FALSE AND IT WILL REVERSES IT STATE WHEN EVER THERE IS A DOUBLE QUOTE.

**STEP 2:** WHENEVER IT ENCOUNTERS A COMMA AND IS WITHIN DOULE QUOTES IT REPLACES THE COMMA AND IT STORE OR ELSE THE CHARACTER IS WRITTEN WITH THE HELF OF If CONDITION AS HIGHLIGHTED.

**Description:**

**Line 1 to line 10:** Importing the necessary jars for map reduce.

**Line 12:** Overriding the map method of mapper class for our computation.

**Line 15:** Storing the entire value of mapper in the string “line”.

**Line16:** Defining a string line1 where the new records after removing commas within quotes will be concatenated.

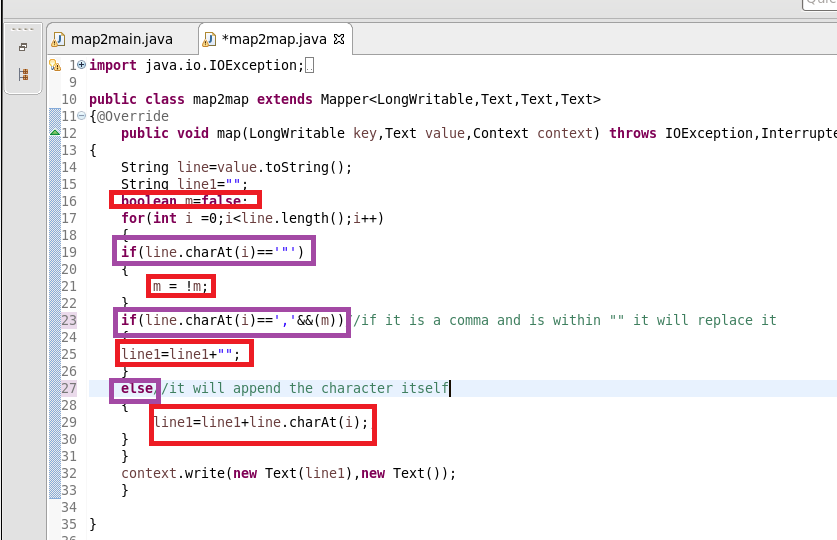
**Line 17:** Setting a Boolean m as False.

**Line 18:** running a for loop starting from starting index to last index of string line.

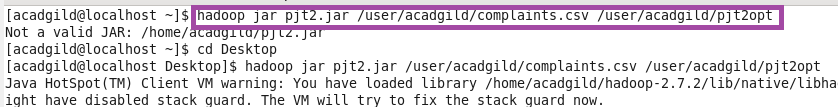
**Line 19 to 23:** WHENEVER the char at that index is double quotes (“), the Boolean gets reversed.

**Line 23 to 27:** if the char at the index is comma and Boolean is true indicating that within double quotes the comma is replaced.

**Line 28:** The character get appended with the new string line1.



**RUNNING JAR:**

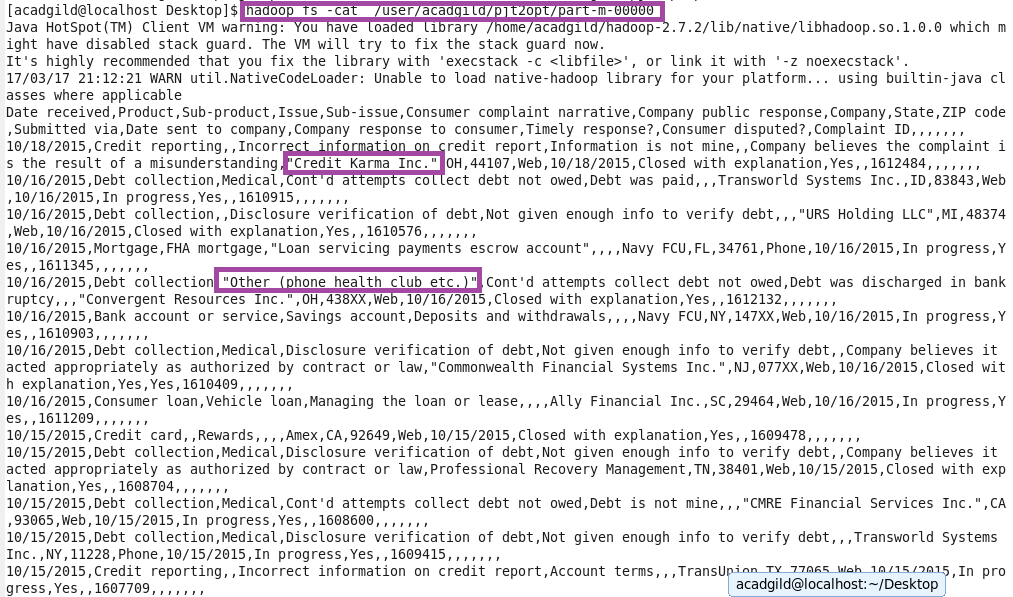


**OUTPUT:**

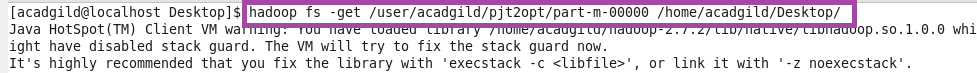


**AS FROM THE FIGURE YOU CAN SEE THAT THE COMMA WITHIN DOUBLE QUOTES ARE REPLACED:**

**AS SHOWN IN HIGHLIGHTED:**



**USED GET COMMAND TO COPY TO LOCAL FOR DOING PIG:**



**PIG COMMANDS**

1. **Write a pig script to find no of complaints which got timely response**

**LOGIC:**

**Step 1:** Loading cleared data

**Step2:** filter complaints where timely response is true

**Step 3:** group by timely response and make a count of id

1. **Write a pig script to find no of complaints where consumer forum forwarded the complaint same day they received to respective company**

**Step1:** Loading cleared data

**Step2:** Filter where date received =date sent to company

**Step 3:** Making COUNT AFTER GROUPIN BY ALL

1. **Write a pig script to find list of companies topping in complaint chart (companies with maximum number of complaints)**

**I AM MAKING A LIST OF TOP 10 COMPANIES**

**Step1:** Loading cleared data

**Step2:** Group by company name

**Step3:** Make a count of cid for each company

**Step4:** order by desc order of counts and limit by 10

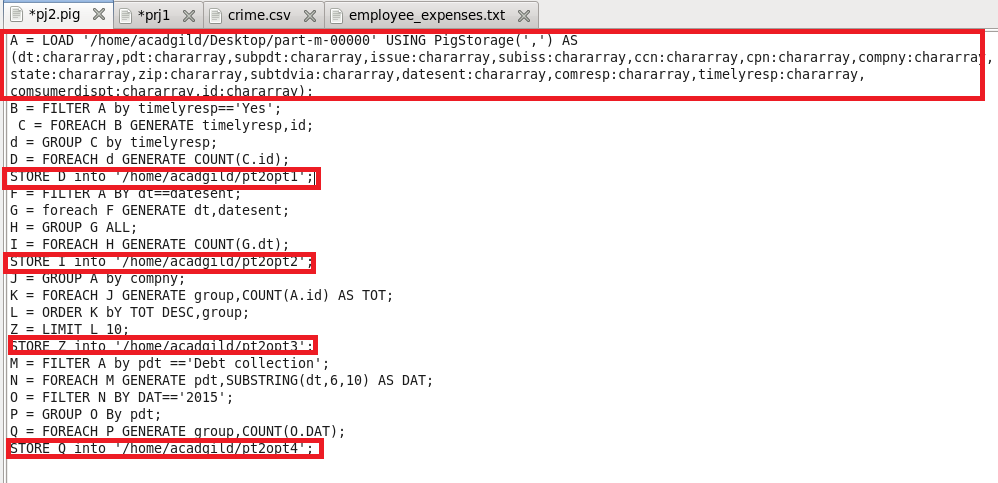
1. **Write a pig script to find no of complaints filed with product type has "Debt collection" for the year 2015**

**Step1:** Filter by product =”DEBT COLLECTION”

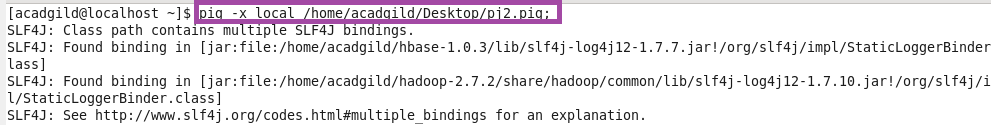
**Step2:** USED SUBSTRING FUNCTION TO GENERATE YEAR as **“DAT”**

**Step3:** Filter by DAT==2015

**STEP4:** MADE COUNT



**RUNNING PIG SCRIPT:**





**OUTPUT:**

