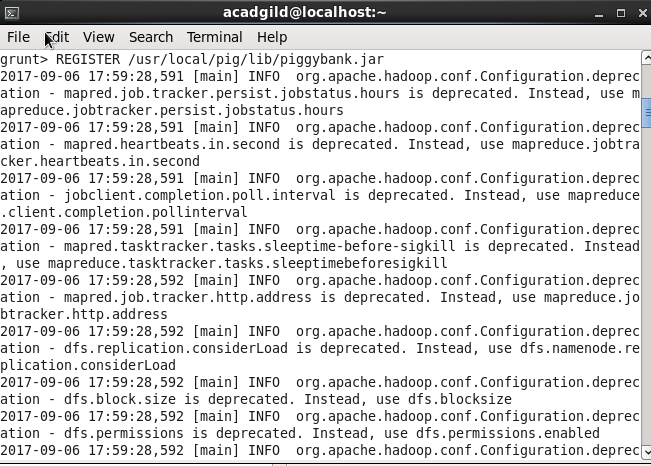
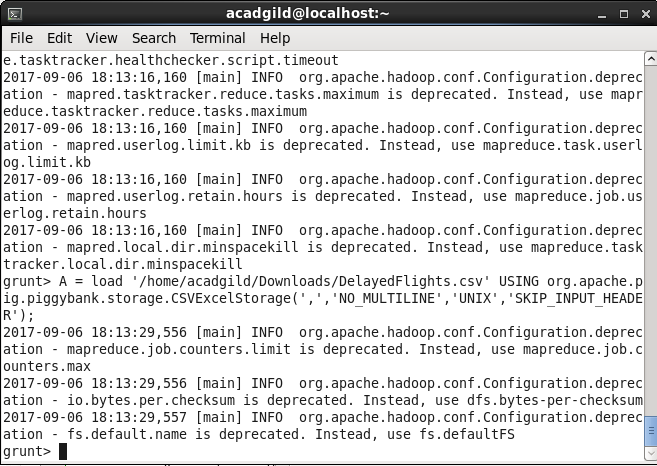
1. **Problem1**

**Register a piggybank.jar file to use the CSVEXCELStorage class**



**we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers**





**we are generating the columns that are required for processing and explicitly typecasting each of them.**



**we are filtering the null values from the “dest” column**



**Now we are grouping C by destination after that we are generating the grouped colum with counted of each group and after that we will sorted in descending order than we limit the record how much we want to see the result.These are all steps to find most visited dstinations**



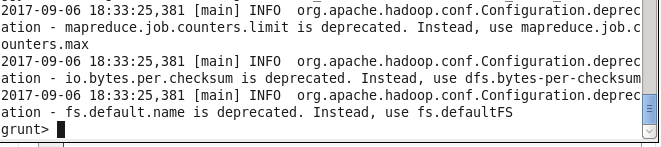






**Now we are loading other file to get the country and city name by using below command**



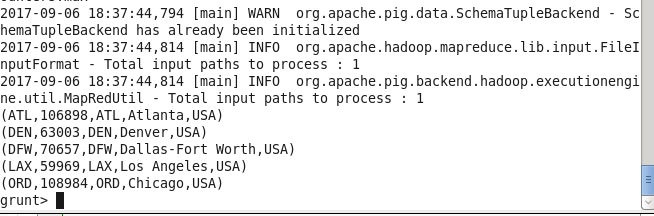


**we are generating dest, city, and country from the previous relation after that we are joining Result and A2 based on a common column, i.e., “dest”**



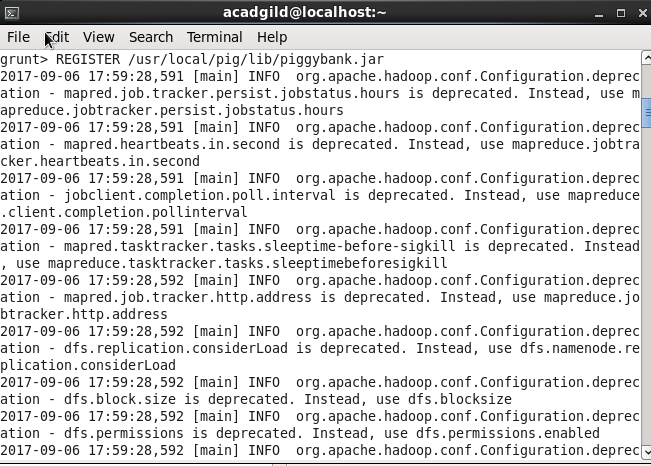


**OUTPUT**

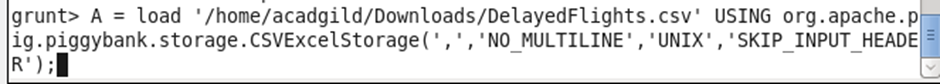


**Problem2**

**Repeat the same step as in Problem 1 first register the piggy bank jar to use CSVEXCELSTORAGE**

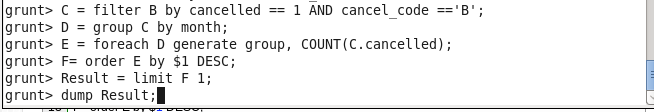


**After that we load the file using CSVEXCELStorage because of its effective technique to handle double quotes and header.**

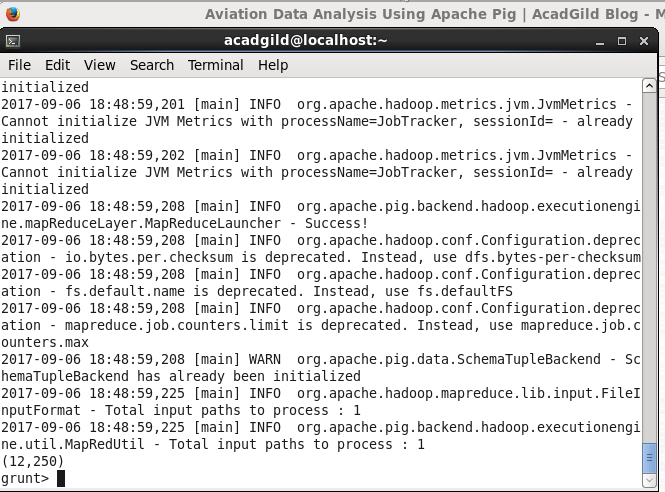


After that we will group it and filter it as per cancellation as well as cancellation code and then we group it per month and then we will count the no of cancellation per month and then we sorted it in descending order and then dump the result



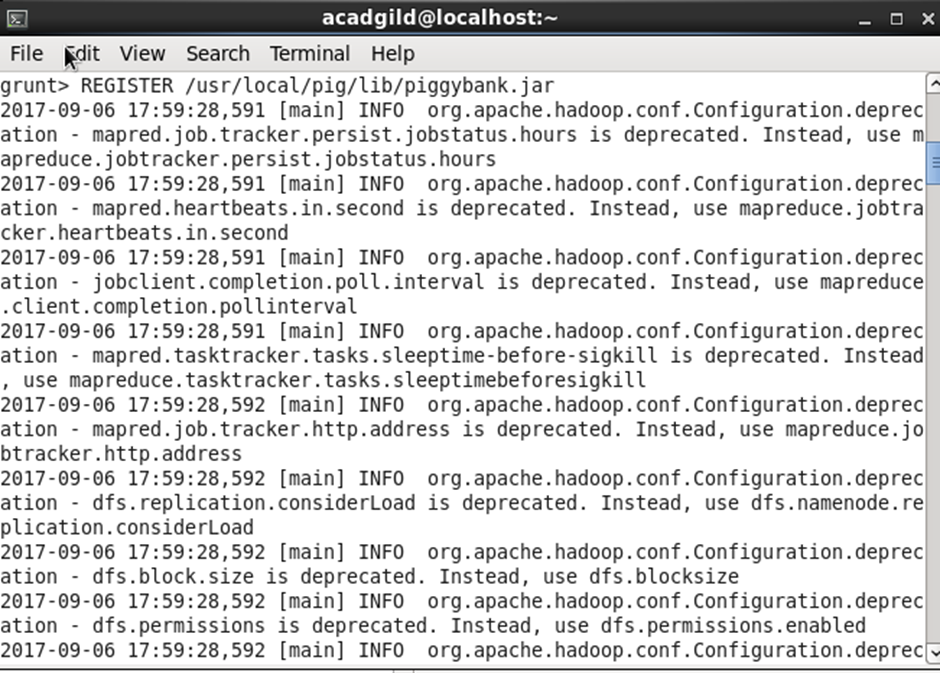


**OUTPUT**

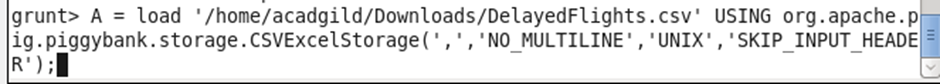


**Problem 3**

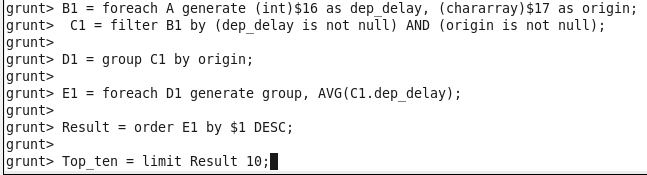
**Repeat the same step as in Problem 1 first register the piggy bank jar to use CSVEXCELSTORAGE**



**After that we load the file using CSVEXCELStorage because of its effective technique to handle double quotes and header**

****

**We are generating the columns for processing and then filter the columns by removing null values and then group it by origin and thenfor each group we will generate average delay and then we sorted it in descending order and then we limit to only 10 values**



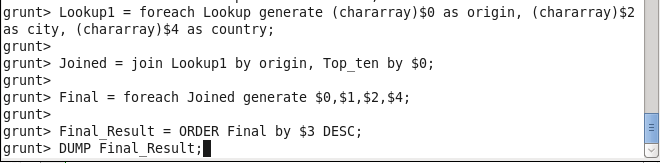
**In the relation Lookup, we are loading another table to which we will look up and find the city as well as the country.**

**In the relation Lookup1, we are generating the destination, city, and country from the previous relation.**

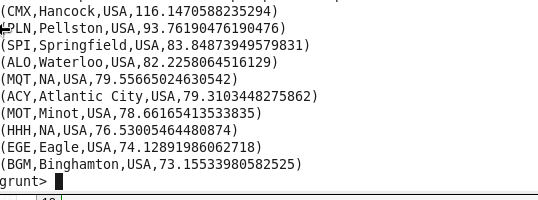
**In the relation Joined, we are joining relation Top\_ten and Lookup1 based on common a column, i.e., “origin.”**

**In the relation Final, we are generating required columns from the Joined table.**



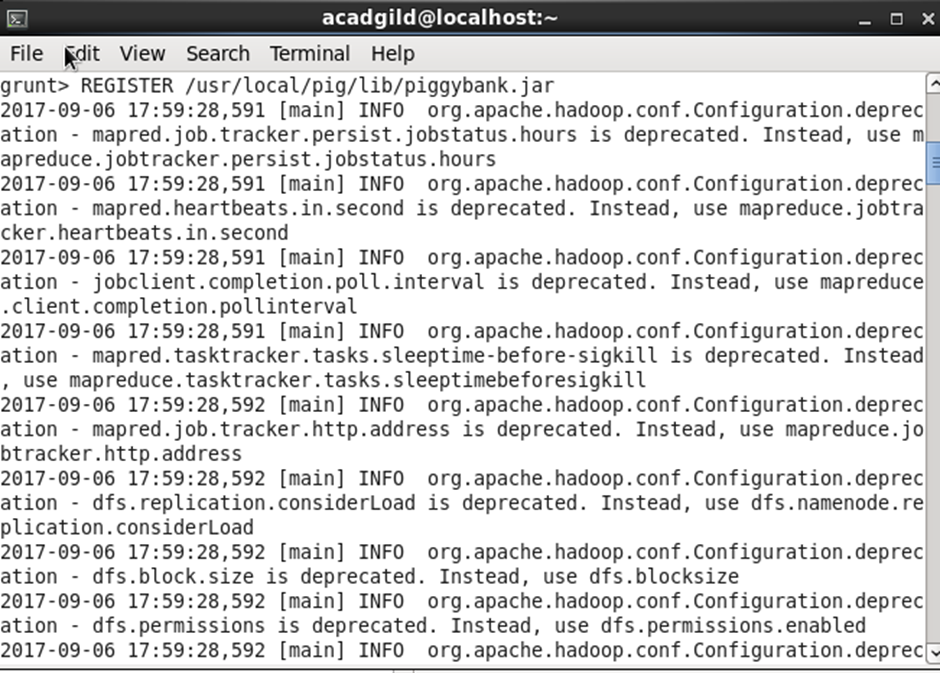


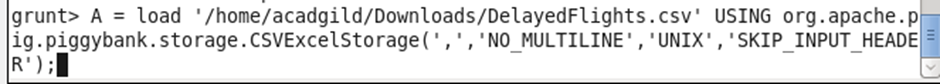
**OUTPUT**



**Problem 4**

**First 3 steps are same as above 3 problems**



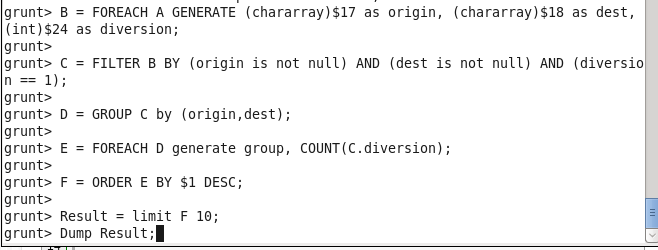
****

**we are filtering the data based on “not null” and diversion =1. This will remove the null records, if any, and give the data corresponding to the diversion taken.**

**In relation D, we are grouping the data based on origin and destination.**

**Relation D finds the count of diversion taken per unique origin and destination.**

**Relations F and Result orders the result and produces top 10 results**



**OUTPUT**

