Aviation Data Analysis Using Apache Hive

The U.S. Department of Transportation’s (DOT) Bureau of Transportation Statistics (BTS) tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled, and diverted flights appear in DOT’s monthly Air Travel Consumer Report, published about 30 days after the month’s end, as well as in summary tables posted on this website. Summary statistics and raw data are made available to the public at the time the Air Travel Consumer Report is released.

**Aviation Data Analysis**

Download the datasets from the following link:

[Delayed\_Flights.csv](https://drive.google.com/file/d/0B_Qjau8wv1KoWTVDUVFOdzlJNWM/view?usp=sharing)

Let us understand few columns of this dataset

There are 29 columns in this dataset. Some of them are:

* Year: 1987 – 2008
* Month: 1 – 12
* FlightNum: Flight number
* Canceled: Was the flight canceled?
* CancelleationCode: The reason for cancellation.

For complete details, refer to [this link](https://drive.google.com/file/d/0B_Qjau8wv1KoTFljWHlxMDE0Szg/view?usp=sharing)

Before we proceed, check the diagram below to understand the flow which is considered in this blog for processing data.

**Step 1: Loading raw data into the root directory of HDFS.**

**Step 2: Pre-processing using Pig.**

This is also known as data cleansing phase.

REGISTER /usr/lib/pig/piggybank.jar;

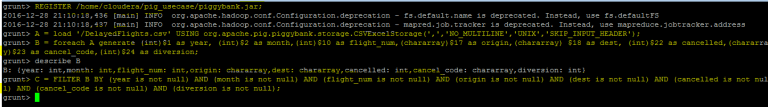
A = load '/user/cloudera/pig\_input/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER');

B = foreach A generate (int)$1 as year, (int)$2 as month, (int)$10 as flight\_num, (chararray)$17 as origin,(chararray) $18 as dest,(int)$22 as cancelled,(chararray)$23 as cancel\_code,(int)$24 as diversion;

C = filter B by (year is not null) and (month is not null) and (flight\_num is not null) and (origin is not null) and (dest is not null) and (cancelled is not null) and (cancel\_code is not null) and (diversion is not null);

Store the pre-processed data into HDFS.

STORE C INTO '/user/cloudera/pig\_avaitation\_output\_hive/' USING PigStorage (',');



In Line 1: As the file is comma separated, we will register and use piggybank jar in order to use the CSVExcelStorage class.

In relation A, we are loading the data using CSVExcelStorage because of its effective technique to handle double quotes and headers.

In relation B, we are generating the columns that are required for processing and explicitly typecasting each of them.

The question may pop up in your mind on my columns selection as why only these columns are considered? Don’t worry, you will understand the reason shortly

In relation C, we are filtering out the null values if any, from the generated columns.

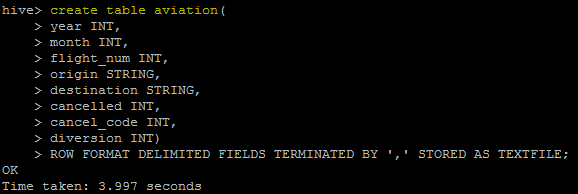
The pre-processing steps might look easy, but it is one of the major steps in any of the production level projects as the data might not always be properly structured and also you may not need all the columns to perform your analysis. Considering complete columns even when it does not have any role in the analysis will only lead to increased processing time. I think now you got the answer of question (in Relation B).

**Step 3: Loading pre-processed data from pig to hive.**

Once the data is cleaned, we need to transfer it to process and gain insights.

3.b) Next, create a hive table with the same schema as you had pre-processed in the Pig.

create table aviation (year INT, month INT, flight\_num INT, origin STRING, dest STRING, cancelled INT, cancel\_code INT, diversion INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;



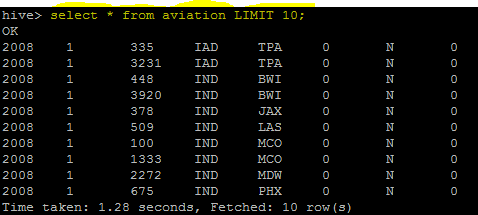
You need to maintain the same order as well while you create the table. You can see that we only need 8 columns out of 29 for our analysis.

3.c) Use below command to load the data to hive.

load data inpath "/user/cloudera/pig\_avaitation\_output\_hive/" into table aviation;

This will load the data into hive table which we had already created.

You can cross check the same using *SELECT \* FROM aviation LIMIT 10;* in hive shell.



**Step 4: Processing the data using Apache Hive**

Here, we have few problem statements which we will be solving using hive.

**PROBLEM STATEMENT 1:**

*Which month have seen the most number of cancellation due to bad weather?*

*Source Code:*

*SELECT month,COUNT(canceled) as t FROM aviation*

*WHERE canceled = 1 AND canel\_code = ‘B’*

*GROUP BY month*

*ORDER BY t DESC*

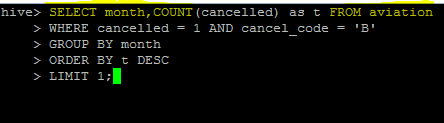
*LIMIT 1;*

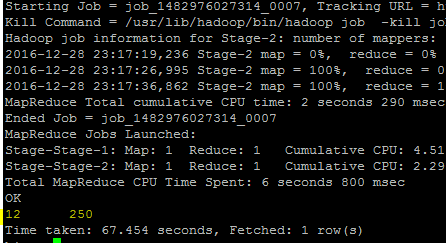
INSERT OVERWRITE DIRECTORY '/user/cloudera/hive\_result1'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;





**PROBLEM STATEMENT 2:**

*Top 10 route(origin and dest) that has seen maximum diversions?*

*Source Code:*

*SELECT origin,dest,COUNT(diversion) as t FROM aviation*

*WHERE diversion = 1*

*GROUP BY origin,dest*

*ORDER BY t DESC*

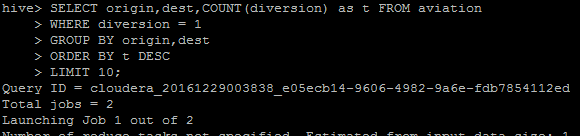
*LIMIT 10;*

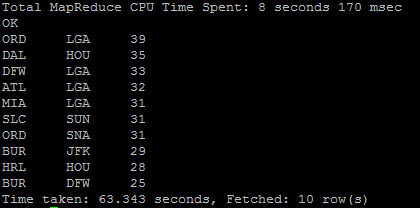
INSERT OVERWRITE DIRECTORY '/user/cloudera/hive\_result2'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;





**PROBLEM STATEMENT 3:**

*Top 5 visited destination.*

*Source Code:*

*SELECT dest,COUNT(dest) as x FROM aviation*

*GROUP BY dest*

*ORDER BY x DESC*

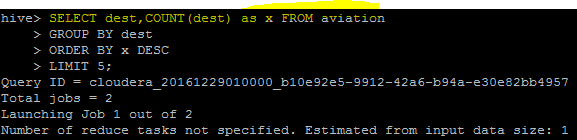
*LIMIT 5;*

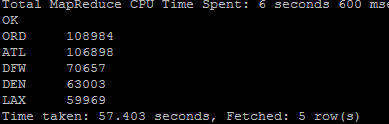
INSERT OVERWRITE DIRECTORY '/user/cloudera/hive\_result3'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;





#### ****Step 5: Data Visualization using Tableau****

As I’m using Cloudera, I can easily connect Hive Cloudera Hadoop with Tableau. After this, you can use this report generating tool for visualization. Below is the image for “Destination” Vs “Count of Visitors to that destination”.

