CREDIT CARD DATA ANALYSIS

TEAM MEMBERS: Tejasvini Shelar, Vineetha Alluri

OBJECTIVE: To collect data, store it in a data warehouse and perform analytics on it to give us meaningful insights.

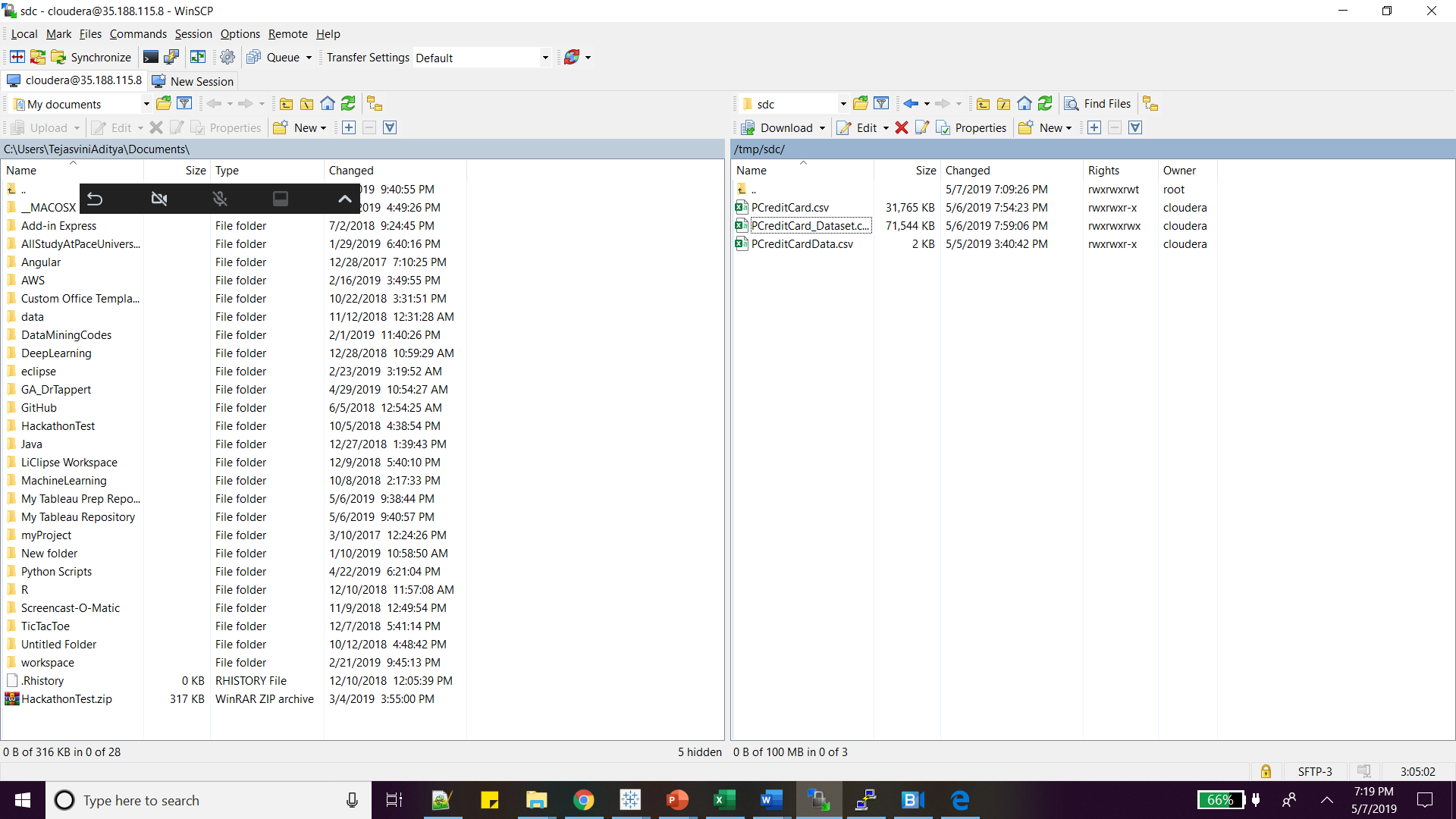
DATA SOURCE: <https://catalog.data.gov/dataset/purchase-card-pcard-fiscal-year-2013>

We plan on loading the data into the data lake using stream sets since it looks simple and interesting to explore. Among the different data lakes available we use Hadoop hive and impala as our data lakes because we have experience in sql and and we can also visualize the outputs in the form of graphical representation. We would also like to generate reports using tableau to make us familiar with the environment as well. In the future we can do data modelling and normalize our tables to make it more efficient.

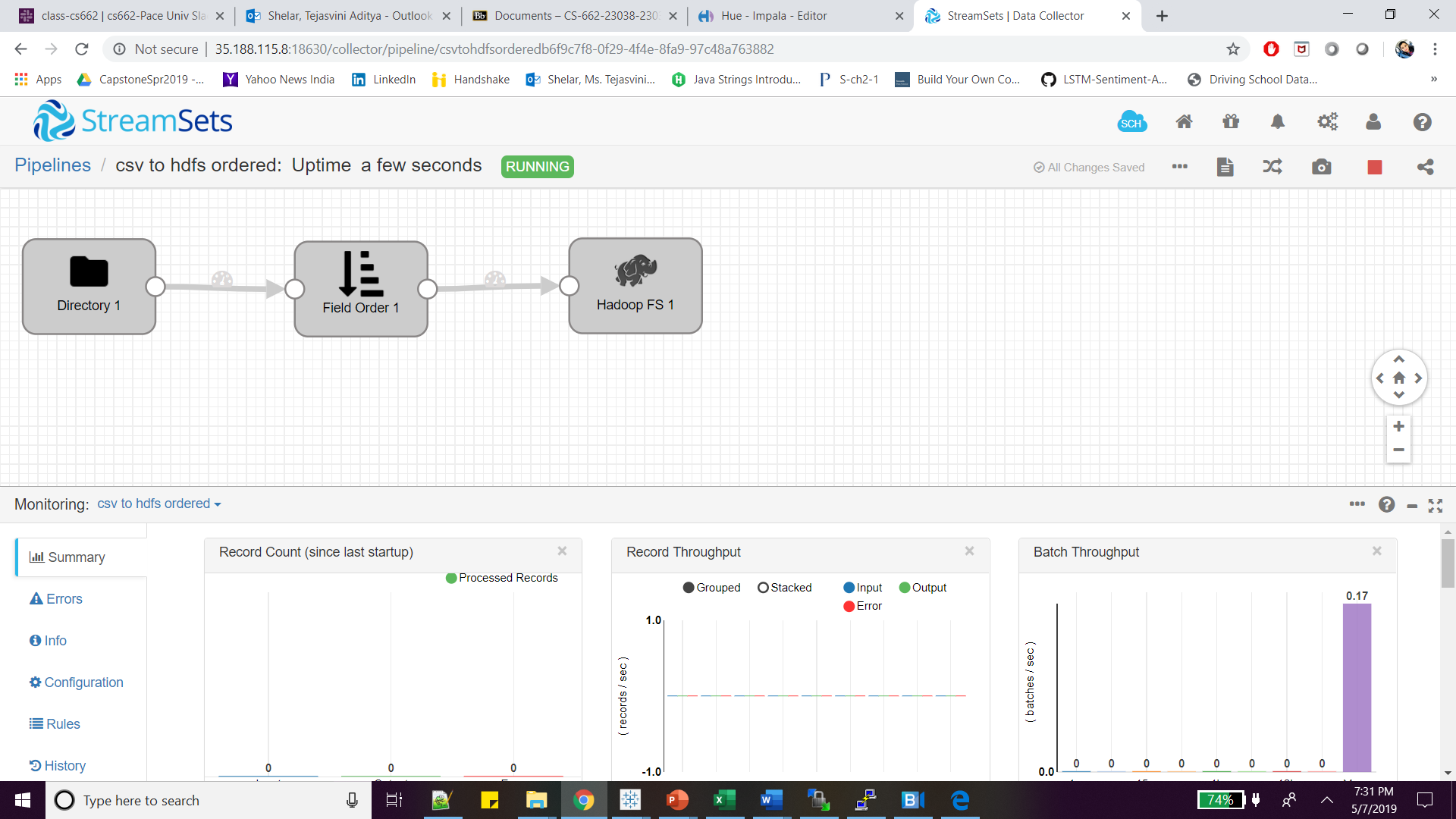
We use streaming to generate a continuous flow of data. HDFS is a primary data storage system used by Hadoop applications. It helps in providing high performance access to our data. Impala is an open source massively processing sql query engine used for querying data on Hadoop. Hive is used for similar purpose, but impala is faster than hive in executing queries since hive uses map reduce operation which takes a lot of time. Tableau is one of the may business intelligence tools which helps anyone understand the data and also helps in deriving meaningful insights.

Steps for the Project:

1. Download Pcard data from data.gov on local system.
2. Load Pcard data from local system to cloudera local file storage using WinSCP at location tmp/sdc:

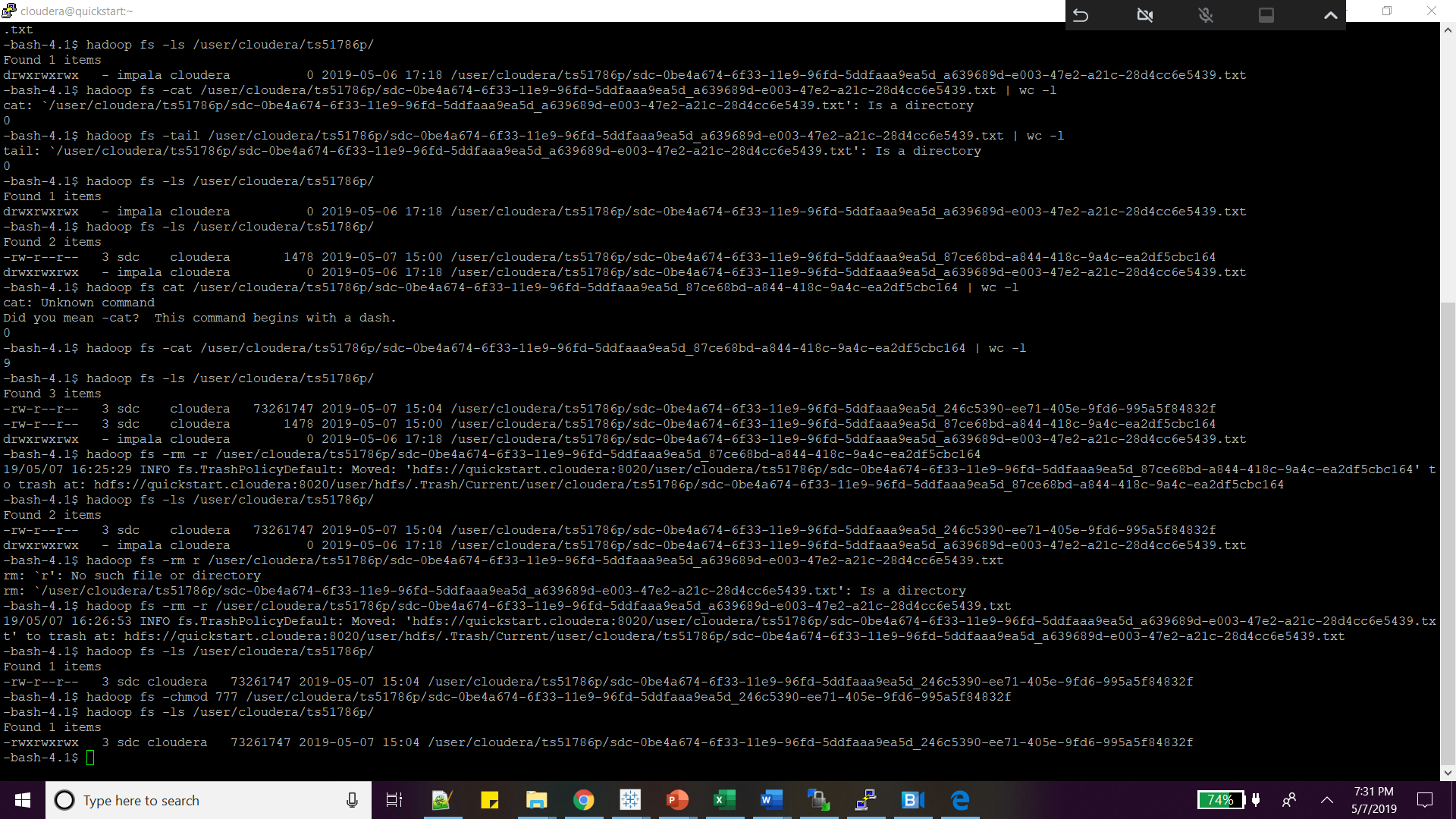


1. Load from cloudera local storage to HDFC using StreamSets:



Command to check data in HDFS:

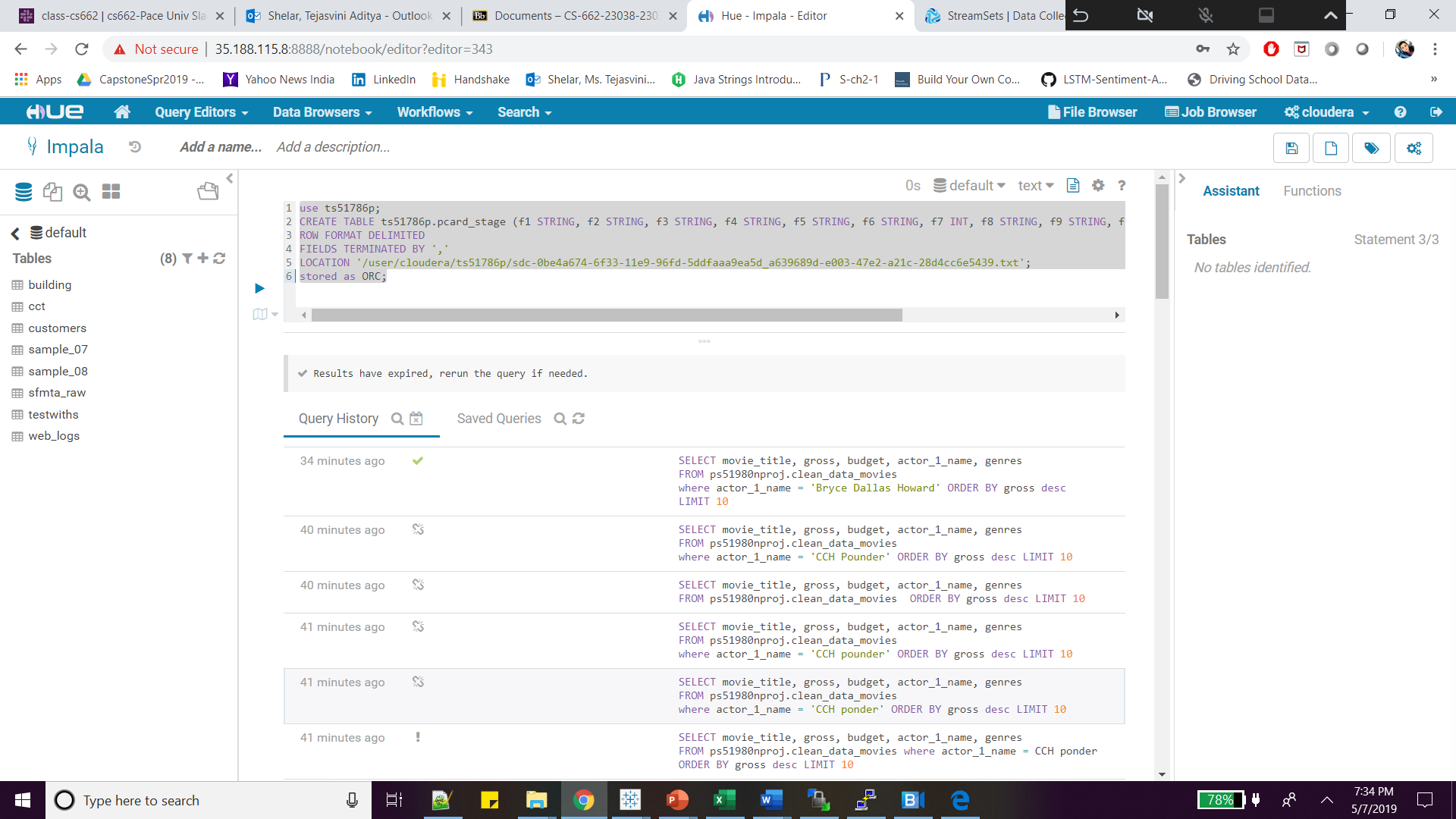
hadoop fs -ls /user/cloudera/ts51786p/



Command to change the access mode of hdfs file:

Hadoop fs -chmod 775 /user/cloudera/ts51786p/sdc-0be4a674-6f33-11e9-96fd-5ddfaaa9ea5d\_246c5390-ee71-405e-9fd6-995a5f84832f.txt

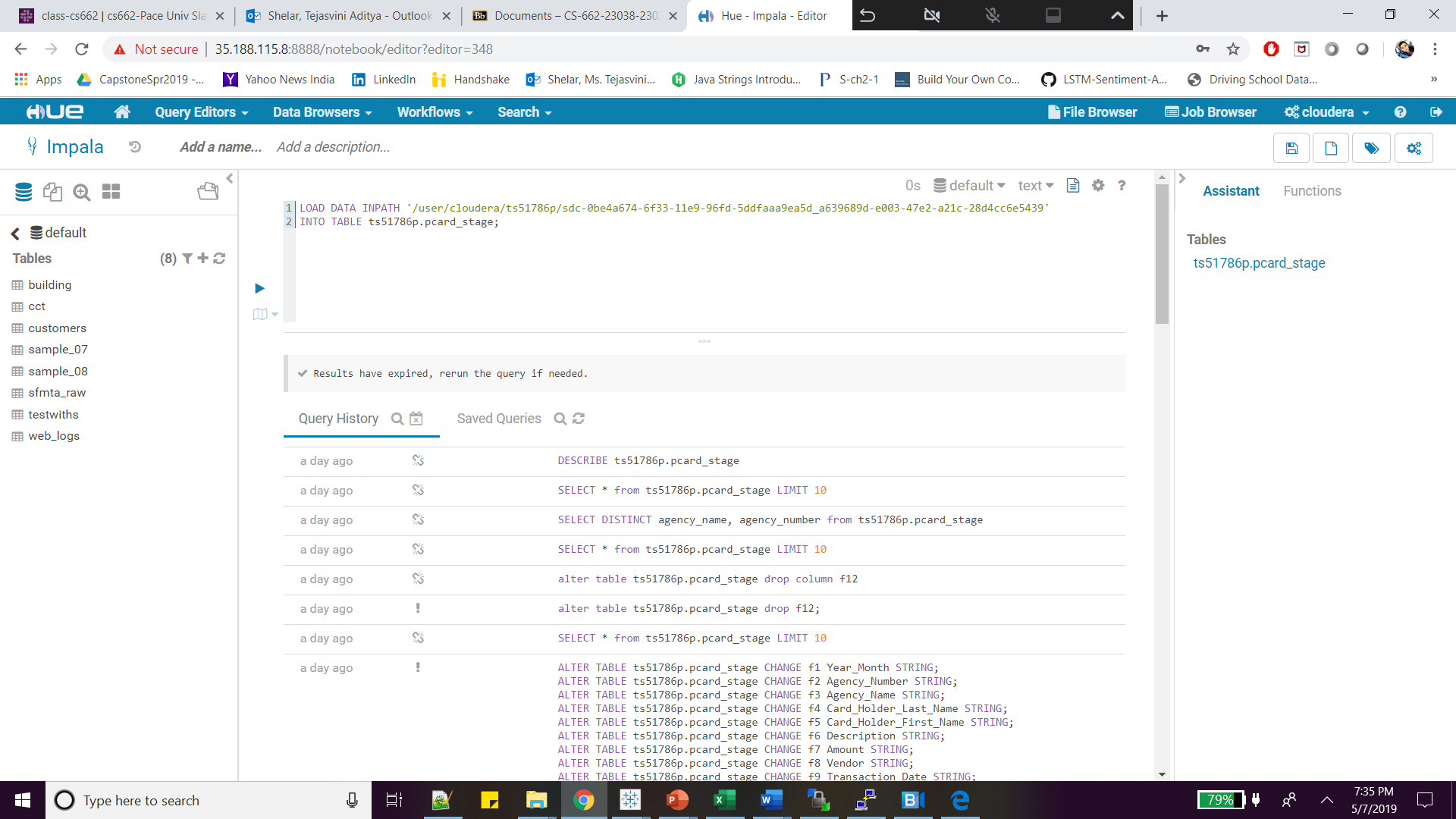
1. Create Impala table from HDFS data using below queries:



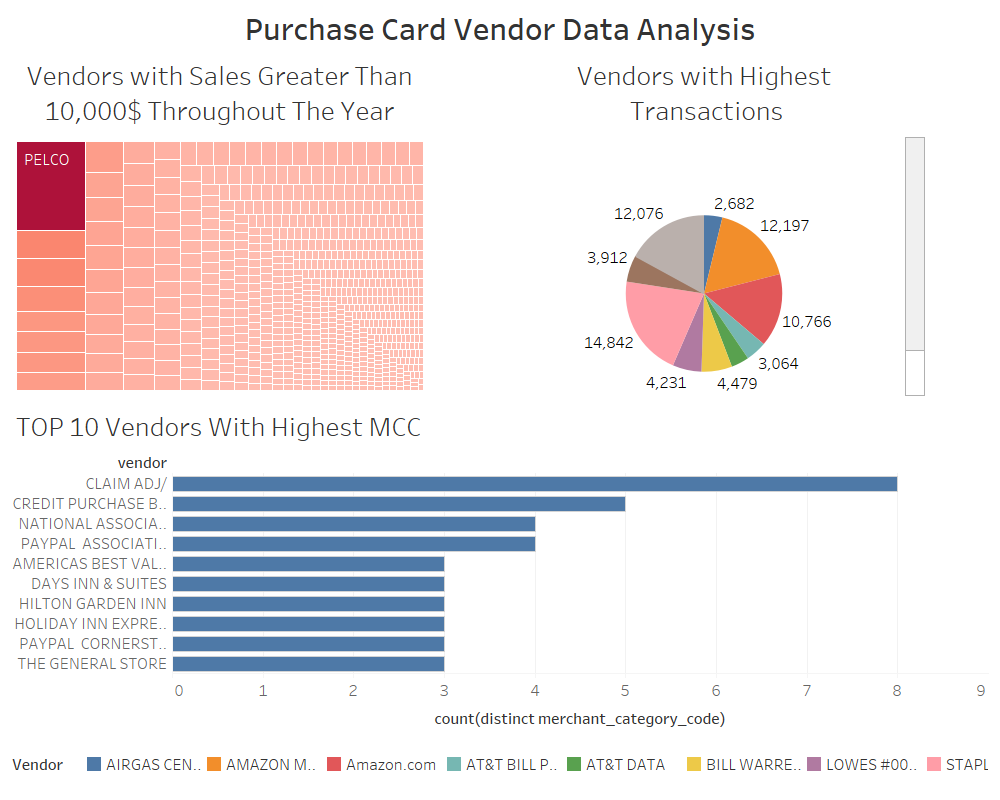
1. Load HDFS data to Impala table as using query:

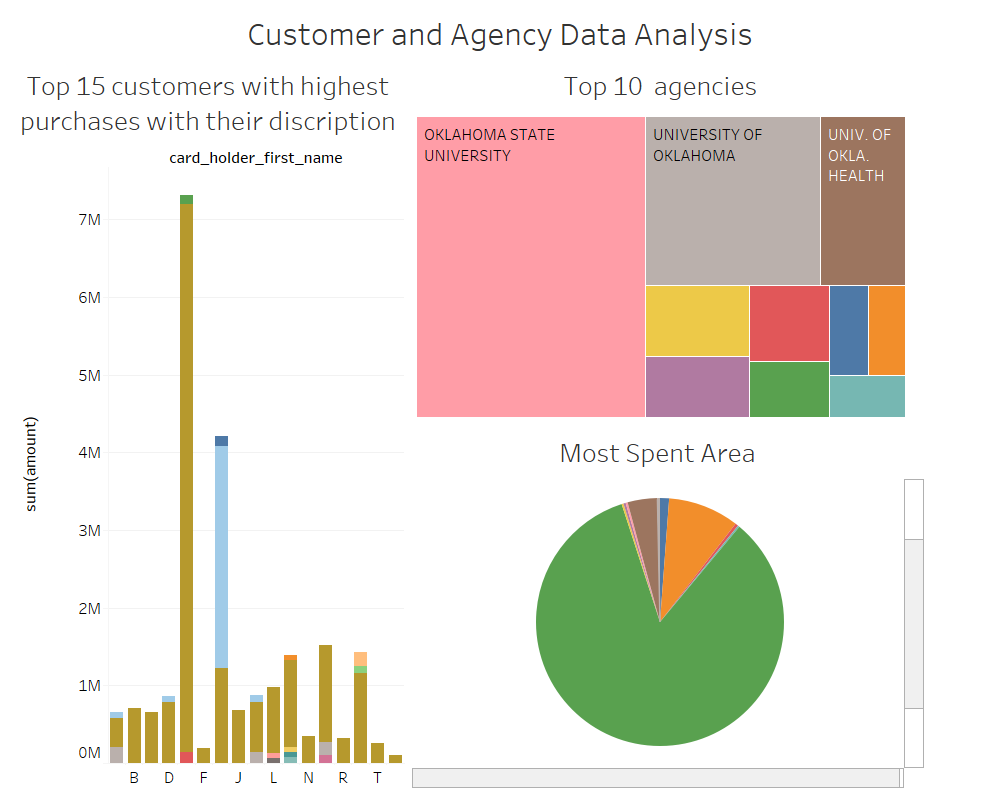
LOAD DATA INPATH '/user/cloudera/ts51786p/sdc-0be4a674-6f33-11e9-96fd-5ddfaaa9ea5d\_a639689d-e003-47e2-a21c-28d4cc6e5439'

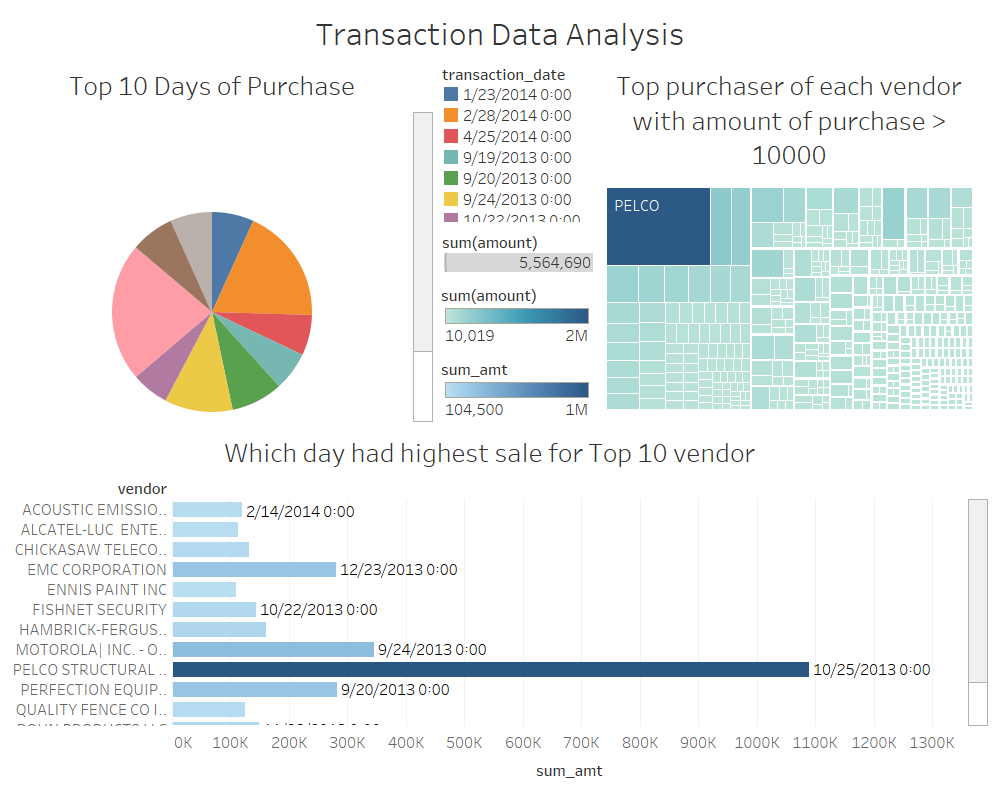
INTO TABLE ts51786p.pcard\_stage;



1. Used Tableau for generating different reports of Impala Pcard data table







1. Queries for generating reports:

/\* top 10 vendors with highest transactions \*/

select vendor, count(\*)

from ts51786p.pcard\_stage

group by vendor

having count(\*) > 20

order by count(\*) DESC

LIMIT 10;

/\* top 10 vendors with highest sales \*/

select vendor, sum(amount)

from ts51786p.pcard\_stage

group by 1

having sum(amount)>10000

order by sum(amount) DESC

LIMIT 10;

/\* bottom 10 vendors with lowest transactions \*/

select vendor, count(\*)

from ts51786p.pcard\_stage

group by vendor

order by count(\*) ASC

LIMIT 10;

/\* bottom 10 vendors with lowest sales \*/

select vendor, sum(amount)

from ts51786p.pcard\_stage

group by 1

order by sum(amount) ASC

LIMIT 10;

/\*Top 10 vendors with highest merchant\_category\_codes \*/

select vendor, count(distinct merchant\_category\_code),group\_concat(distinct merchant\_category\_code)

from ts51786p.pcard\_stage

group by vendor

order by count(distinct merchant\_category\_code) DESC

LIMIT 10;

/\*Top 10 purchases with highest sales \*/

select description, sum(amount)

from ts51786p.pcard\_stage

group by 1

having sum(amount)>100

order by sum(amount) DESC

LIMIT 10;

/\*Bottom 10 purchases with lowest sales \*/

select description, sum(amount)

from ts51786p.pcard\_stage

group by 1

having sum(amount)> 0

order by sum(amount) ASC

LIMIT 10;

/\*Top 10 customers with highest purchases \*/

select card\_holder\_last\_name, card\_holder\_first\_name, sum(amount)

from ts51786p.pcard\_stage

group by 1,2

having sum(amount) > 100

order by sum(amount) DESC

limit 10;

/\*top 10 agencies which sold most credit cards\*/

select agency\_number, agency\_name,

count(distinct card\_holder\_last\_name, card\_holder\_first\_name) as count1

from ts51786p.pcard\_stage

group by 1,2

having count1 > 1

order by count1 DESC

limit 10;

/\*Top purchaser of each vendor with amount of purchase > 10000\*/

select card\_holder\_last\_name, card\_holder\_first\_name, vendor, sum(amount)

from ts51786p.pcard\_stage

group by 1,2 ,3

having sum(amount) > 10000

order by sum(amount) DESC

/\*Top 10 days when purchase was the highest\*/

select transaction\_date, sum(amount)

from ts51786p.pcard\_stage

group by 1

having sum(amount) > 1

order by sum(amount) DESC

LIMIT 10;

/\*Which day had highest sale for each vendor\*/

select transaction\_date, vendor, rank1, sum\_amt

from (

select transaction\_date, vendor, rank() over (partition by vendor order by sum(amount) DESC) as rank1, sum(amount) as sum\_amt

from ts51786p.pcard\_stage

group by transaction\_date,vendor

having sum(amount) > 1

order by sum(amount) DESC

)a

where rank1 = 1

1. Future Work:

Build normalized data model for data warehouse.

For loading data every month use Oozie job scheduling.

1. Data Modelling

Dimension Table:

|  |
| --- |
| **Card\_Provider** |
| agency\_Number |
| agency\_name |

|  |
| --- |
| **Card\_Holder** |
| First\_name |
| Last\_name |

|  |
| --- |
| **Product\_Vendor\_Merchant** |
| description |
| Vendor |
| merchatCC |

Facts Table:

|  |
| --- |
| **Transaction** |
| Amount |
| Transaction\_date |
| posted\_date |

Conclusion:

We got to learn how to use different technologies in the course of doing the project. The insights we achieved as a result of it were immensely helpful in understanding the data. Given more time we would like to create facts table, dimension table and also perform more complex analytics.