PROJECT REPORT

Kuntal Patel

**A Strategy to Deploy Big Data Technology in a Firm**

**Factors**

* Developers are already using SQL: This is important to make sure the technology which we choose to analyze data should be like SQL, so it will be easy transition for developers.
* Developers analyzing data locally: local infrastructure will provide secure environment but with current enhancements and available option makes Cloud computing a good scalable and secure option when we must deal with very large amount of data.
* Company is public listed hence the amount of data which needs to analyzed will be huge.
* Other competitors are already adopted cloud infrastructure and newer technologies to become first in the completion.
* There is enough funding available.
* Strategy must be decided, and Proof of Concept should be ready within three weeks.

What we already have?

* Enough financial resource
* Past data from year 2014 to 2017.

What we need?

* Cloud infrastructure
* SQL developers
* Big data Framework
* Scalable solution

What we are trying to get?

* Fast decision making
* Comparative view from other competitors
* How fast we deploy concluded strategies in the market before competitor
* May include more segments in near future business. Right now, only dealing with Consumer, Home Office and corporate.
* Target most revenue generated cities, states and areas.
* Trying to recognize patterns among highly involved customers and cater their other requirement as well.

How we are going to get?

* Migration from existing system to new
* How much training or support required to deploy?
* Out of all available options which option of cloud infrastructure best suited.

How Big data will help Retail Business

* Data Makes You Customer-centric: Analyzing each step of the customer journey across various platforms to elevate remarketing efforts and secure a sale is one of the best use cases for Big Data.
* Data Decreases Churn: Data can reveal patterns of long-standing customers and help companies identify new leads and customers who are most likely to generate comparatively more profits in the future.
* Data Improves the Supply Chain: By aggregating internal data with external information such as weather or public news, organizations can increase efficiency and make the system highly adaptable to real market conditions.
* Strategic Planning: Recommendation engines can analyze past purchases and discover patterns of complementary products to be recommended for a shopping cart.
* Cost Reduction: It is possible to minimize costs, automate some flows and free up more time to focus on the core tasks instead of doing admin work, already covered by the algorithm-run data.

Reference:<https://towardsdatascience.com/5-amazing-improvement-big-data-can-bring-to-retail-2c70bdd5a871>

**Why Google Cloud platform?**

* Google-grade security: The Google security model is an end-to-end process, built on over 15 years of experience focused on keeping customers safe on Google applications like Gmail and Google Apps. With Google Cloud Platform your applications and data take advantage of the same security model.
* Billing by the second: Compute Engine instances are charged in one-second increments with a one-minute minimum, so you don't pay for compute minutes that you don't use.
* Big data: Google’s big data technology innovations like MapReduce, Bigtable, and Dremel, plus next-generation breakthrough services and frameworks for cloud data warehousing (Big Query), advanced machine learning (AI Platform), batch and real-time data processing (Cloud Dataflow, Cloud Pub/Sub, Cloud Dataproc), intelligent data preparation (Cloud Dataprep by Trifacta) and stunning visual analytics (Google Data Studio) help you transform your business with powerful data insights. GCP big data analytics solutions are serverless, removing the complexity of building and maintaining a data analytics system, so you can accelerate your time-to-insight.
* Global network: Google's backbone network uses advanced software-defined networking and edge caching services to deliver fast, consistent, and scalable performance.
* Environmentally friendly: GCP data centers run on half the energy of a typical data center and run on 100% renewable energy where available.

Reference: <https://cloud.google.com/free/docs/what-makes-google-cloud-platform-different>

**Why Apache Hadoop ecosystem?**

* License Free: Anyone can go to the Apache Hadoop Website, From there you Download Hadoop, Install and work with it.
* Open Source: Its Source code is available, you can modify, change as per your requirements.
* Meant for Big Data Analytics: It can handle Volume, Variety, Velocity & Value. Hadoop is a concept of handling Big Data, & it handles it with the help of the Ecosystem Approach.
* Ecosystem Approach: (Acquire, Arrange, Process, Analyze, Visualize) Hadoop is not just for storage & Processing, Hadoop is an ecosystem, that is the main feature of Hadoop. It can acquire the data from RDBMS, then arrange it on the Cluster with the help of HDFS, after then it cleans the data & make it eligible for analyzing by using processing techniques with the help of MPP(Massive Parallel Processing) which shared nothing architecture, then in last it Analyze the data & then it Visualize the data.
* Shared Nothing Architecture: Hadoop is a shared nothing architecture, that means Hadoop is a cluster with independent machines. (Cluster with Nodes), that every node performs its job by using its own resources.
* Distributed File System: Data is Distributed on Multiple Machines as a cluster & Data can stripe & mirror automatically without the use of any third-party tools. It has a built-in capability to stripe & mirror data. Hence, it can handle the volume. In this, there are a bunch of machines connected together & data is distributed among the bunch of machines on the back panel & data is striping & mirroring among them.
* Commodity Hardware: Hadoop can run on commodity hardware that means Hadoop does not require a very high-end server with large memory and processing power. Hadoop runs on JBOD (just bunch of disks), so every node is independent in Hadoop.
* Horizontal Scalability: We do not need to build large clusters; we just keep on adding nodes. As the data keeps on growing, we keep adding nodes.
* Distributors: With the help of distributors, we get the bundles, also built-in packages, we do not need to install each package individually. we just get the bundle & we will install what we need for.

Reference: <https://k21academy.com/big-data-hadoop/hadoop-key-features-advantages/>

**What components of Hadoop ecosystem are we using in this project?**

* HIVE: Apache Hive is a system built on top of Apache Hadoop that facilitates easy data summarization, ad-hoc queries, and the analysis of large datasets stored in various databases and file systems that integrate with Hadoop. Hive offers a simple way to apply structure to large amounts of unstructured data and then perform batch SQL-like queries on that data. Hive easily integrates with traditional data center technologies using the familiar JDBC/ODBC interface.
* SPARK: Apache Spark is a powerful unified analytics engine for large-scale distributed data processing and machine learning. The Hive metastore can be used with Spark SQL can run on the Spark execution engine, optimizing workflows and offering in-memory processing to improve performance significantly.

**Data Preprocessing**

**PART1**

**How to be checked that there is some issue in dataset?**

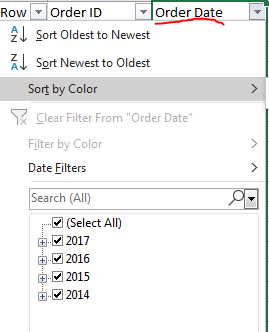
**Exploratory analysis (**Excel based analysis)

**Total columns :21**

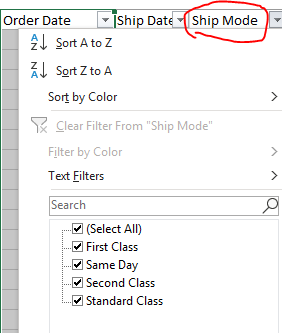
**Total Rows:9994 + Header = 9995**

**Important Columns Explanation:**

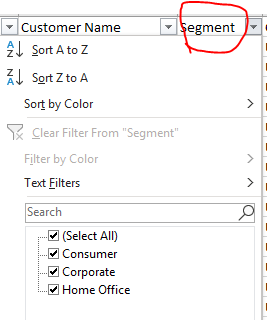
* Order Date: Sales data from 2014 to 2017



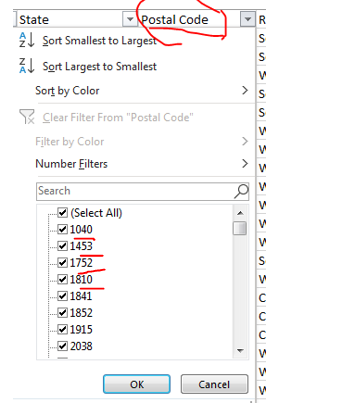
* Ship Mode: Total 4 ways to send delivery.



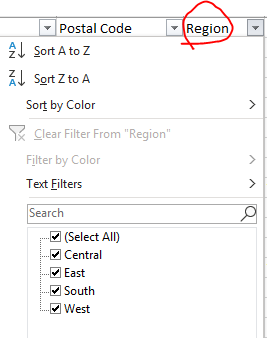
* Segment:Currently only dealing with three segments



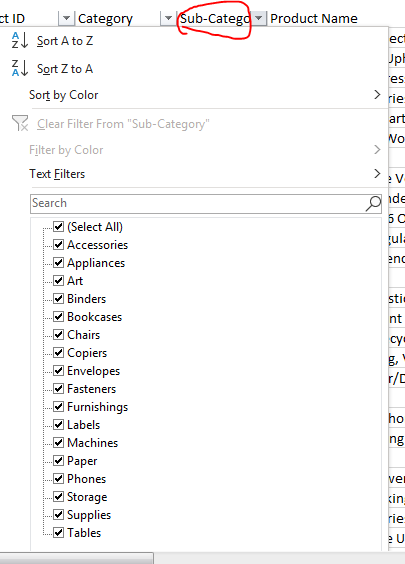
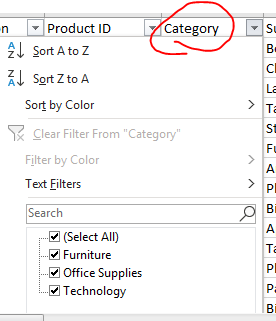
* Postal Code: Usually zip code is of 5 digit number but we have 4 digit zip codes too in the dataset.we need to pad extra zero in front of 4 digit zip codes so it will not create any issue when we query data based on zip code.



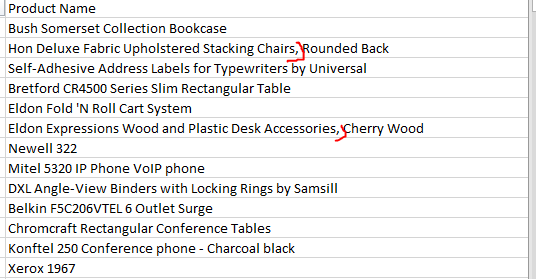
* Region:Company is dealing with all parts of USA.



* Category ans Sub-category: from total 3 categories total 17 sub categories are classified for all offered products



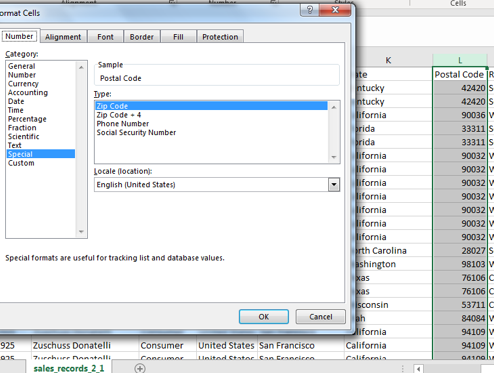
* Product Name: There are some extra commas appeared in below screenshot which need to address because CSV file separator is comma and hence underlying system will interpret it as new field.



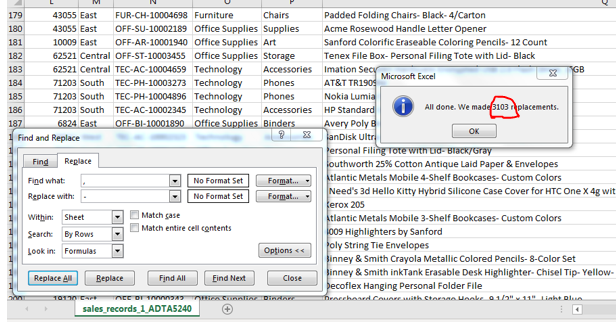
Preprocess columns (Excel)

**PART1**

* Postal Code: Used Excel for zip code validation. pad extra 0 when 4-digit zip code.



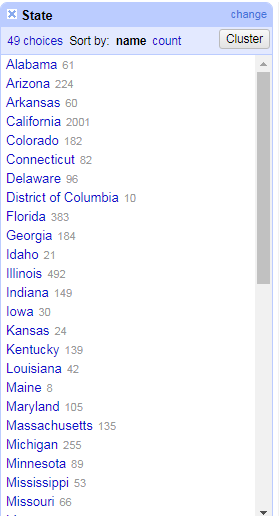
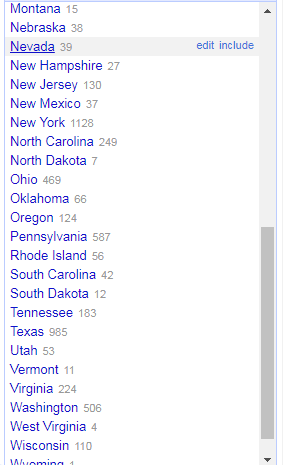
* Product Name: Replaced ‘ , ‘ with hyphen.



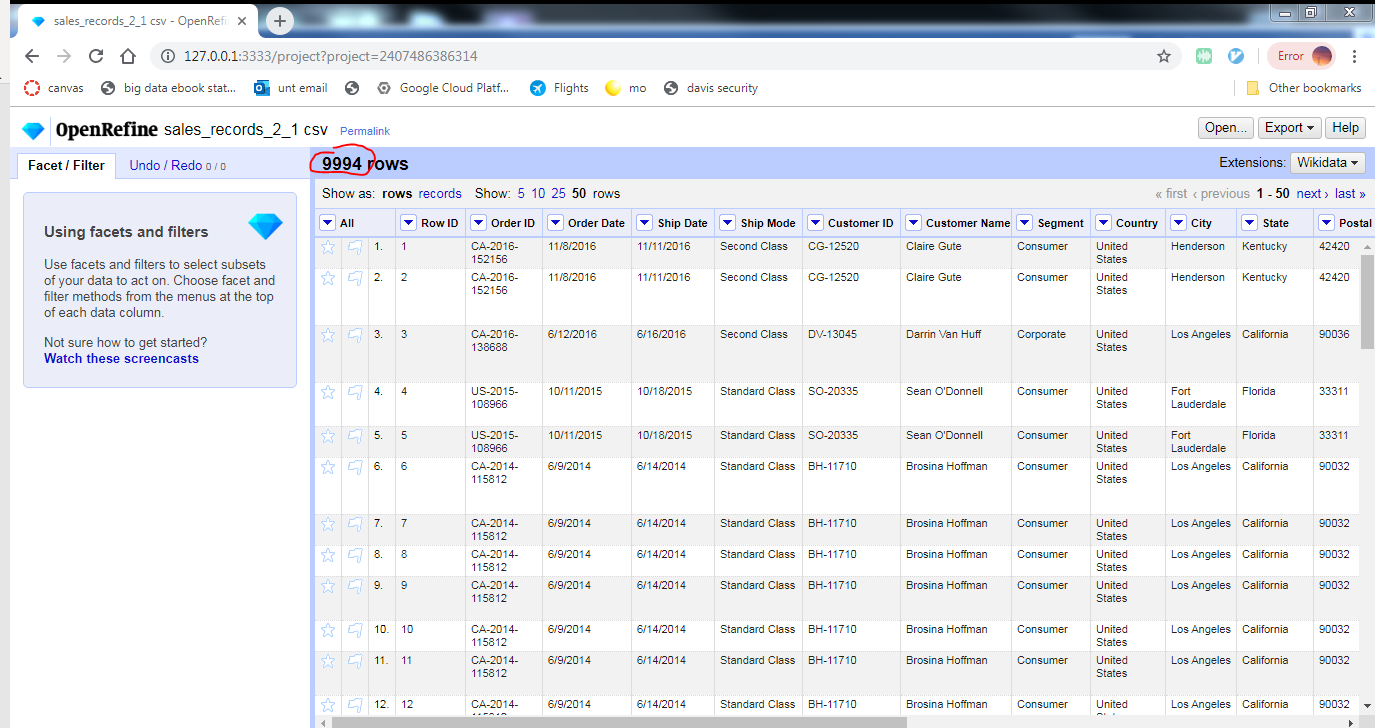
**PART2**

**Verifying in Open Refine for typo or spelling mistakes**

* There are no missing fields in the whole dataset
* Except Alaska, firm is having business in all 49 states. There are no typographical errors.

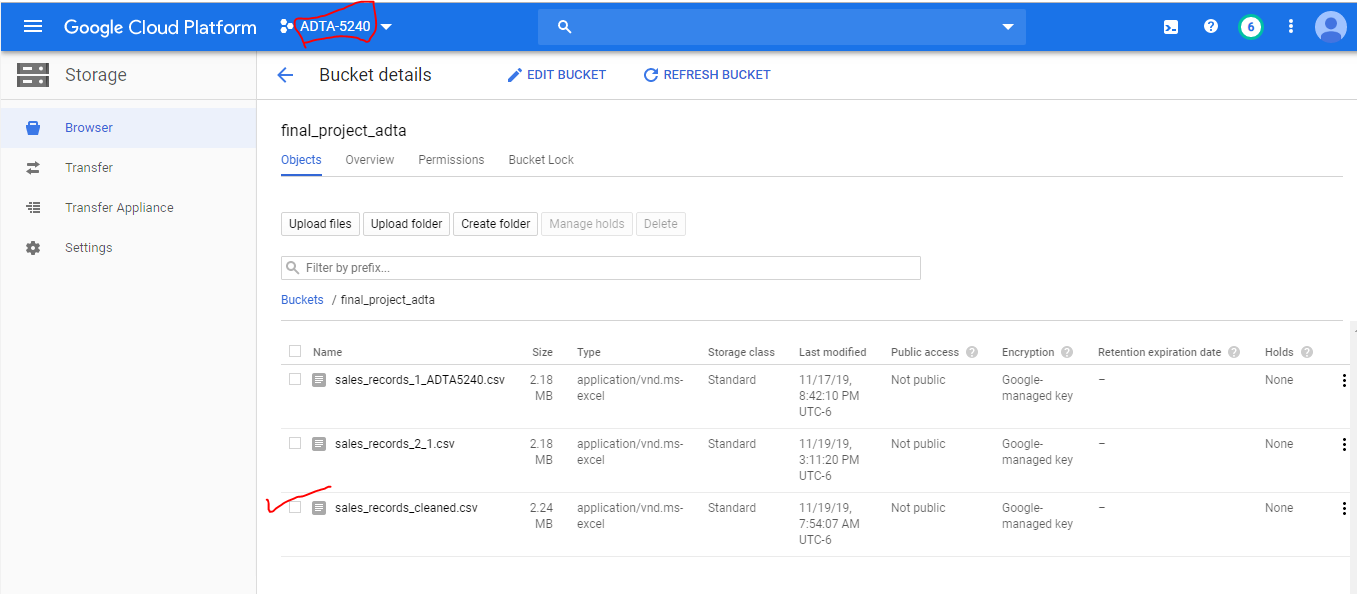
 

opneRefine view



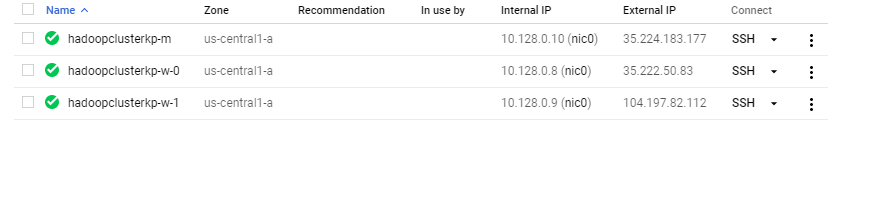
How to load datasets from local file system to Google cloud cluster.

* Create a bucket in Google cloud GUI.
* Upload files from local system to this bucket.



How to load dataset from local storage bucket to HDFS.

* Start one master node and two worker nodes



* Make one empty directory on HDFS using command line SSH for master node.

hdfs: Hadoop distributed file system

dfs: distributed file system

-mkdir: make new directory

/user/username/folder: path

hdfs dfs command path



* Copy dataset from google cloud bucket to master node.

gsutil : gsutil is a Python application that lets you access Cloud Storage from the command line.

gs:// gsutil: uses the prefix gs:// to indicate a resource in Cloud Storage

cp : copy

gsutil gs:// source (i.e. [BUCKET\_NAME]/[OBJECT\_NAME] ) destination



* Copy dataset from master node to HDFS in empty directory ‘finalcleandata’.

-put :will put file from master node to hdfs



hdfs dfs -put source file on master node destination empty folder in hdfs

* Start beeline client of Hive on default port 10000.

beeline: Command line interface

-u: database URL

jdbc: jdbc connector to connect database

hive2: subcomponent to run Hive database server



beeline -u URL with port number

* Create schema on dataset using beeline

CREATE EXTERNAL TABLE TABLENAME

(define column names with datatype)

Specify how dataset columns are separated from each other

Specify in what format it stored

Specify where is the table located from which we can map data to external table.

Command:

CREATE EXTERNAL TABLE IF NOT EXISTS finalschema

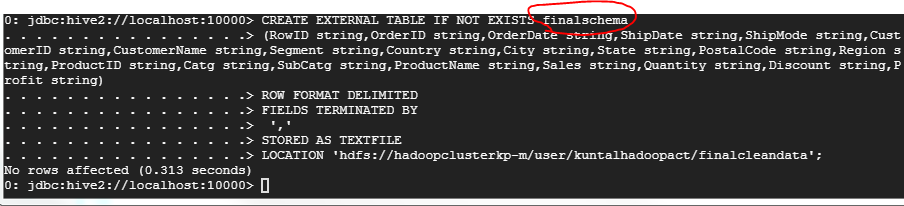
(RowID string,OrderID string,OrderDate string,ShipDate string,ShipMode string,CustomerID string,CustomerName string,Segment string,Country string,City string,State string,PostalCode string,Region string,ProductID string,Catg string,SubCatg string,ProductName string,Sales string,Quantity string,Discount string,Profit string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’

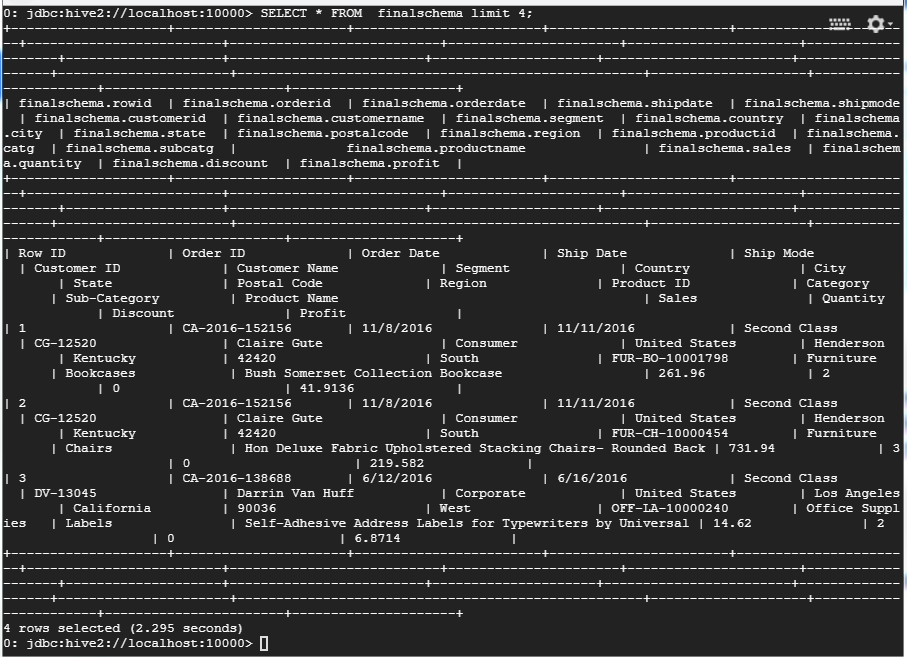
STORED AS TEXTFILE

LOCATION ‘hdfs://hadoopclusterkp-m/user/kuntalhadoopact/finalcleandata’;



* Check the first 3 line of contents of finalschema.

SELECT \* FROM finalschema LIMIT 4;

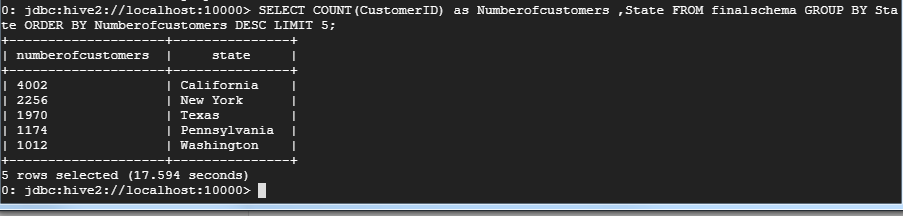


**Apache Hadoop Ecosystem with Hive**

Information that can be queried from the dataset:

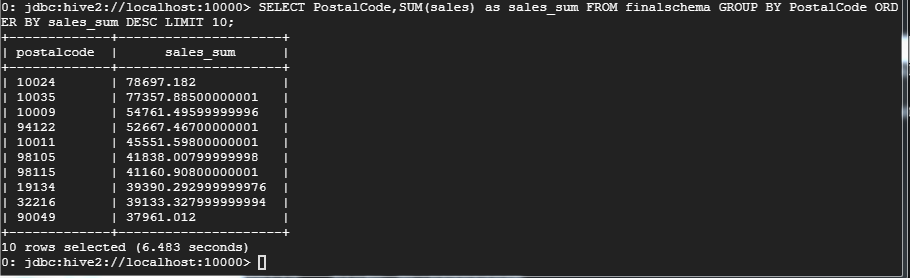
The top five states where the company has the most number of customers

SELECT COUNT(CustomerID) as Numberofcustomers ,State FROM finalschema GROUP BY State ORDER BY Numberofcustomers DESC LIMIT 5;



The top ten zip-code areas where the company gets the best sale amounts, i.e., the highest sale figures

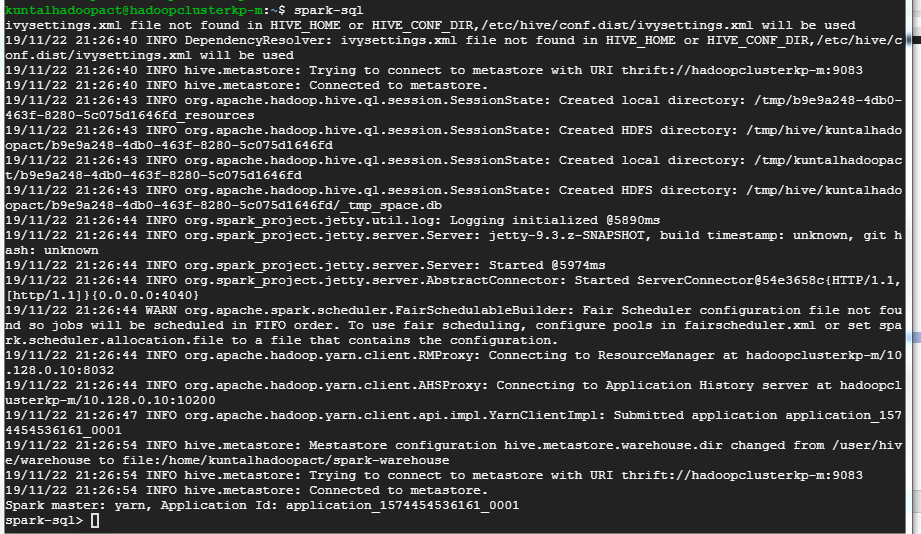
SELECT PostalCode,SUM(sales) as sales\_sum FROM finalschema GROUP BY PostalCode ORDER BY sales\_sum DESC LIMIT 10;



**Apache Hadoop Ecosystem with Spark**

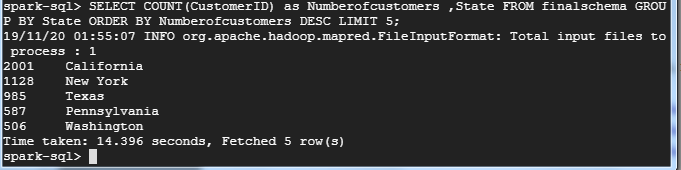
* Start spark sql

command : spark-sql

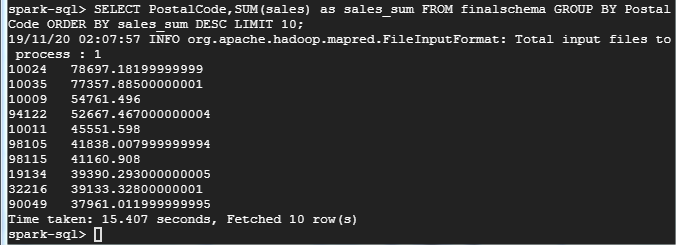


* Run same queries which we have run on Hive to query data.

SELECT COUNT(CustomerID) as Numberofcustomers ,State FROM finalschema GROUP BY State ORDER BY Numberofcustomers DESC LIMIT 5;

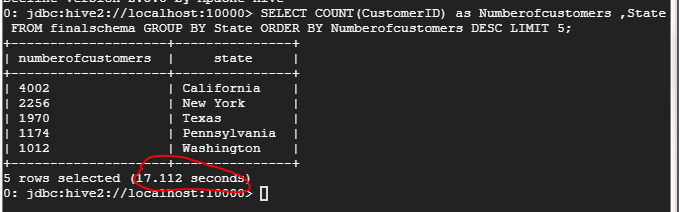


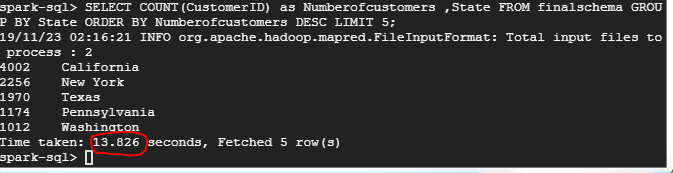
SELECT PostalCode,SUM(sales) as sales\_sum FROM finalschema GROUP BY PostalCode ORDER BY sales\_sum DESC LIMIT 10;



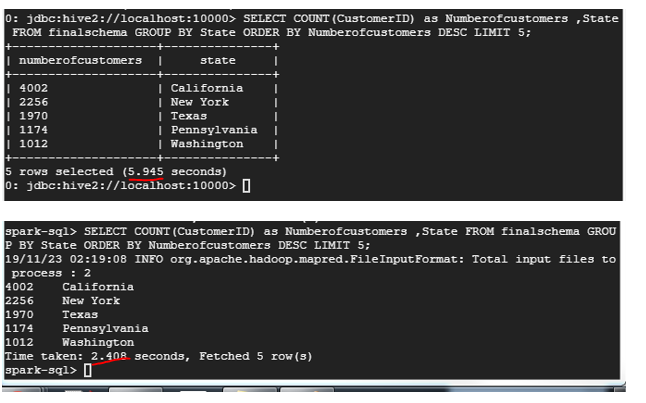
**Compare Speed/Performance of Hive/MapReduce vs. Spark**

**First run**

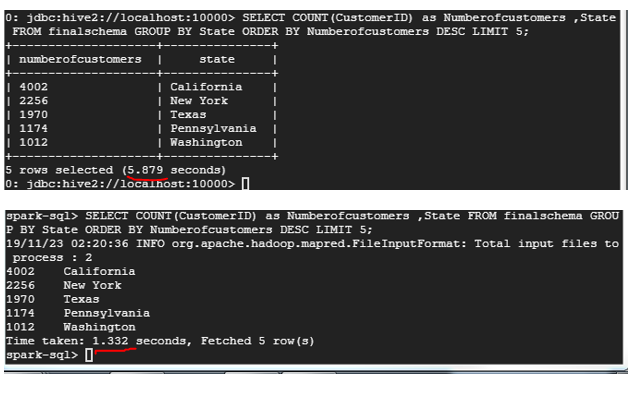




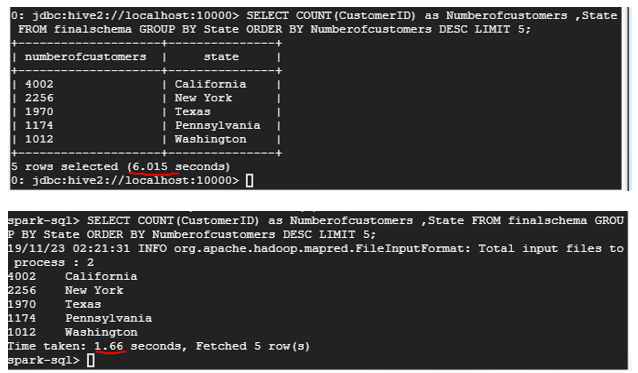
**Second run**



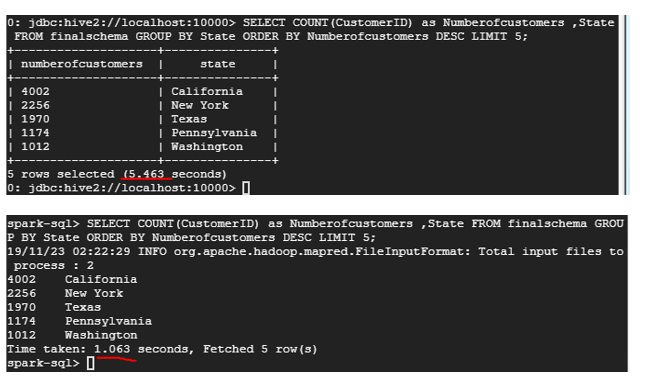
**Third run**



**Fourth run**



**Fifth run**



**Why spark sql is better than hive?**

* Spark SQL use Spark Core as its processing engine to perform the task. Spark supports in-memory processing which is usually 50–100 times faster than regular processing. Also, Spark SQL offers auto query optimization for better performance. It all happens in the back end and it is abstracted away from the developer.
* Spark uses Directed Acyclic Graphs (DAGs) programming technique to perform the tasks on a distributed network of clusters, Apache Hive on the other hand uses Map Reduce programming method under the hood for all processing.
* DAGs are better than Map Reduce because, out of multiple DAGs Spark generates during processing a dataset, some of those DAGs can be Map Reduce, but in Spark we are not limited to just Map Reduce.

Reference: <https://www.quora.com/Is-Spark-SQL-faster-than-Hive>

**Final Presentation Videos: YouTube Links**

**Clip 1:** <https://youtu.be/jJpGtM3yPpE>

**Clip 2:** https://youtu.be/ufQjFxRvCJU

YouTube links of the submitted final presentation videos: