

# E-Commerce Big Data Analytics Using Hadoop

Jahnavi Valisetty – M.S. Computer Science

## Abstract

In the past few year's usage of internet by the people all over the world is increased enormously. Increase in the usage of internet there has been increase in online shopping that is e-commerce. It results the amount of information generated to increase exponentially. The main challenge in increasing data is how to identify and extract useful information from this massive amount of data and filtering the meaningful data for the decision makers. Due to heavy competitions in the marketplace, it is very sensitive to take better decisions to reach the goal. The impact of big data analytics in business intelligence allows overcoming these challenges. To enhance the business process in e-commerce the companies need to adapt business intelligence through Big Data Analytics. The impact of big data analytics plays a vital role in e-commerce for the competitive business environment. In this project we will analyze the ecommerce data sets which are available online and will find the insights of the data to improve the business and analyze the sales. Below are some insights that we get from our analysis

- Find the most regular customers - So that they can give some offers to such customers to improve the business more.
- Find the best-selling products by season wise - So that such particular products quantity will be increased according to the demand
- Find the products which are having lowest sale – So that they can provide some offers on that products to increase the sales.

# **INDEX**

<b>List of Figures</b>	7
<b>1. INTRODUCTION</b>	10
1.1 Scope	12
1.2 Existing System	12
1.3 Proposed System	12
<b>2. SYSTEM ANALYSIS</b>	13
2.1 Functional Requirement Specifications	13
2.2 Performance Requirements	17
2.3 Software Requirements	17
2.4 Hardware Requirements	17
<b>3. SYSTEM DESIGN</b>	18
3.1 Architecture Design	18
3.2 Modules	18
3.3 UML Diagrams	19
3.3.1 Activity Diagrams	19
3.3.3 Use case Diagrams	20
3.3.4 Sequence Diagrams	21
<b>4. SYSTEM IMPLEMENTATION</b>	21
<b>5. OUTPUT SCREENS</b>	48

<b>6. SYSTEM TESTING</b>	71
<b>7. CONCLUSION</b>	78
<b>8. BIBLIOGRAPHY</b>	79

## **LIST OF FIGURES**

<b>S.NO</b>	<b>FIG.NO</b>	<b>NAME</b>	<b>PG.NO</b>
1	1	Proposed Architecture	18
2	2	Activity Diagram	20
3	3	Use Case Diagram	20
4	4	Sequence Diagram	21
5	5	HDFS Design Architecture	24
6	6	Blocks Replication in Data Nodes	26
7	7	Hive Architecture	29
8	8	Hive job execution flow	30
9	9	Key features of Tableau	31
10	10	Data Visualization Formats	32
11	11	Tables	48
12	12	Structure of tables	48
13	13	State wise orders	49
14	14	Total amounts collected in ten categories	49
15	15	Quarter wise sales	50
16	16	Top ten cities with highest number Of orders	50
17	17	Top ten states with highest number Of orders	51
18	18	Top ten cities with highest discounted Price	51

19	19	Top ten states with highest discounted Price	52
20	20	Total amounts collected on sales each month	52
21	21	Top ten highest categories in subcategory	53
22	22	Top ten highest rating products	53
23	23	Top ten products with highest discounted Price	54
24	24	Top ten categories having highest records	54
25	25	Top ten products with highest discounted price	55
26	26	Top ten products with highest records	55
27	27	Top ten states with highest discounted price	56
28	28	Top ten cities with highest discounted price	56

## 1. INTRODUCTION

E-commerce or electronic commerce is an exchange of purchasing or selling online. It draws on technologies, for example, mobile commerce, electronic finances transfer, production network management, and substantially more. The e-commerce field is increasing quickly over the world. The e-commerce business in India will be worth 38 billion dollars by 2016 and it is estimated to reach 159 billion dollars by 2020.

The e-commerce firms are becoming quickly everywhere throughout the world with a large number of exchanges made every day. Along these lines, one needs to analyze that information and draw some useful bits of knowledge from it.

Here, we convey to you a business use case of an e-commerce organization which needs to analyze their exchanges and draw some useful bits of knowledge out of it, which will be useful for their business development. The increase in online shopping increases the competition between different companies in e-commerce field. Due to advancement of technologies the current business development processes are outdated. To enhance the business process in e-commerce the companies need to adjust business intelligence through huge information examination. Big Data is a term which is continually evolving. It is a large measure of structured and unstructured information that can be mined for data. These informational collections are very large and complex that conventional information processing isn't capable to process them. Huge information is being used in numerous sectors. We will see the influence of Big Data Analytics in changing the E-Commerce business, so the business evaluated as these E-commerce can benefit the most users n associations from utilizing Big Data, because there will be data of the information collected on everyday bases. Numerous enormous retailers value this current information's data and helps them for predicting the user interests and provide their customers relative and interested searches when they shop on their site, with the goal that they draw in the customer by giving the required and relevant searches of the items or items. These preferences are altogether generated from Big Data examination. Enormous information comprises of two types of information one is structured and the other one is unstructured. The structured information deals with simple essential and regular information like name and address. Unstructured information deals with numerous information which is retrieved from places like web-based life and includes videos. This data assumes a fundamental role to e-commerce businesses, which can better serve the customer. E-Commerce businesses and retailers which operates online, to give a better service for this competitive and quick environment, pulls in the customers by online advertisements. As per the online approaches are part up with very specific investigation instruments for separate information and information's. Decisions will be optimized because of the huge volume of information obtained from multiple sources and in different configurations. These are the focal point of Big Data where it gives out superior, information driven yields. Usage of enormous information in ecommerce is given below

- By making use of real time analysis prices are changed in order to compete with other retailers.

- By analyzing the type of most products and least selling products we they can focus on the reason behind that and can take the decisions to improve their business.
- By analyzing the user ratings and reviews the business can be improved in such way that the user satisfied.
- The most vital role of big data is to provide a better experience for the customer when they make use of the website and also try to satisfy the user needs by giving the relevancy search.
- To predict the user interest and behaviors Predictive analytics is been used so that to provide the required products of user interest and satisfy his demands by giving proper online advertisements based on the predictive data.
- Big data is been used for personalization which in turn personalizes the users information such as mail id and address in order to increase the rate of conversation.

**Big Data** is a widely inclusive term for any gathering of data sets so vast and complex that it ends up hard to process utilizing available data the board apparatuses or conventional data preparing applications. The difficulties incorporate catch, term, stockpiling, look, sharing, exchange, investigation and perception. The pattern to bigger data sets is expected to the extra data logical from examination of a solitary huge arrangement of related data, when contrasted with isolated littler sets with a similar aggregate sum of data, enabling relationships to be found to "spot business patterns, counteract ailments, battle wrongdoing, etc.

**Hadoop** is a free, open-source Java-based system that underpins the preparing of substantial informational indexes in an appropriated figuring condition. It is a piece of the Apache venture supported by the Apache Software Foundation. Hadoop makes it conceivable to run applications on frameworks with a large number of hubs including a huge number of terabytes. Its circulated document framework encourages fast information exchange rates among hubs and enables the framework to keep working continuous if there should arise an occurrence of a hub disappointment. This methodology brings down the danger of cataclysmic framework disappointment, regardless of whether a critical number of hubs become out of commission. The Hadoop system is utilized by real players including Google, Yahoo and IBM, generally for applications including web crawlers and promoting. The favored working frameworks are any kind of Linux, windows (here we have to utilize Cygwin) yet Hadoop can likewise work with BSD and OS X.

**MapReduce** is for to preparing the information which s put away in HDFS. HDFS is Hadoop's execution of a disseminated document framework. It is intended to hold a lot of information and give access to this information to numerous customers dispersed over a system. MapReduce is a fantastic model for conveyed processing, presented by Google in 2004. Each MapReduce work is made out of a specific number of guide and lessen undertakings. The MapReduce model for serving various occupations comprises of a processor sharing line for the Map Tasks and a multi-server line for the Reduce Tasks. To run a MapReduce work, clients ought to outfit a guide work, a decrease work, input information, and a yield information area as appeared in figure 2. Whenever executed, Hadoop completes the accompanying advances: Hadoop breaks the info information into various information things by new lines and runs the guide work once for every datum thing, giving the thing as the contribution for the capacity. Whenever executed, the

guide work yields at least one key-esteem sets. Hadoop gathers all the key-esteem sets produced from the guide work, sorts them by the key, and gathers together the qualities with a similar key. For each particular key, Hadoop runs the diminish work once while passing the key and rundown of qualities for that key as information. The lessen capacity may yield at least one key-esteem sets, and Hadoop thinks of them to a record as the last outcome. Hadoop enables the client to design the activity, submit it, control its execution, and question the state. Each activity comprises of autonomous assignments, and every one of the undertakings need a framework space to run.

### **Data processing with Hive**

Hive is a Data Warehouse programming that encourages questioning and overseeing gigantic information living in conveyed stockpiling. Instead of composing enormous crude guide decrease programs in some programming language, Hive gives a SQL-like interface to information put away in Hadoop File System. Hive in light of its SQL like question language is regularly utilized as the interface to an Apache Hadoop based information warehouse. Hive is viewed as friendlier and progressively recognizable to clients who are accustomed to utilizing SQL for questioning information. Pig fits in through its information stream qualities where it assumes the errands of bringing information into Apache Hadoop and working with it to get it into the structure for questioning. A decent diagram of how this function is in Alan Gates posting on the Yahoo Developer blog titled Pig and Hive at Yahoo! from a specialized perspective both Pig and Hive are including finished so you can do undertakings in either instrument. Anyway, you will discover one apparatus or the other will be favored by the distinctive gatherings that need to utilize Apache Hadoop. The great part is they have a decision, and the two devices cooperate.

#### **1.1 SCOPE**

In this project, we take the shopping data with many attributes like the customer details, product details, customer address etc. The aim of this project is the analysis of the huge data like finding the top customers doing a greater number of sales and the products which are getting sale in more numbers and which are collecting highest amount and at what cities and states the sales are the highest and lowest etc. using the Hadoop through map reduce programs. We also used hive at the same time to write simple queries which in fact gives the desired result faster without having to write long programs and for visualization tableau is being used to show the reports.

#### **1.2 EXISTING SYSTEM**

The e-commerce firms are growing rapidly all over the world with millions of transactions made every day. There is a great challenge not only to store and manage such a large amount of data, but also to analyze and extract meaningful information from it and getting the benefit out of that analysis. There are several approaches to collecting, storing, processing, and analyzing big data. Present these analysis activities are happening using data warehousing technologies which are based on RDBMS. But it is more expensive and time consuming. To help better in this area, we are using the Hadoop and Hadoop Eco-systems.

#### **1.3 PROPOSED SYSTEM**

Our proposed system makes use of big data tools in ecommerce which allows firms to build better models, which produce results with higher precision. Big data enables you to hear the voice of every customer as against to customers at large. Many E-Commerce companies use this information to personalize their communications with their costumers, which in turns leads to meeting consumer expectations and satisfied customers. Here we will be showing a use case to find the insights of ecommerce data and get the benefits for business growth. Following are some insights that we find from the analysis

- State wise count of customer.

From this, we can know which the states are have highest number of customers and which have lowest number of customers. If the customers count is very low, they can do, more promotions in such areas and they can grow their business.

- City wise number of orders.

From this, we can know which cities have highest number of orders and lowest number of orders.

- Quarter wise sales.etc.

## 2. SYSTEM ANALYSIS

### 2.1. FUNCTIONAL REQUIREMENTS

#### **Yield Design**

A quality yield is one, which meets the necessities of the end client and presents the data plainly. In any framework aftereffects of handling are conveyed to the clients and to other framework through yields. In yield structure it is resolved how the data is to be uprooted for quick need and furthermore the printed copy yield. It is the most significant and direct source data to the client. Proficient and canny yield configuration improves the framework's relationship to help client basic leadership.

1. Structuring PC yield ought to continue in a composed, all around considered way; the correct yield must be created while guaranteeing that each yield component is planned with the goal that individuals will discover the framework can utilize effectively and viably. At the point when examination structure PC yield, they ought to Identify the particular yield that is expected to meet the prerequisites.

2. Select strategies for showing data.

3. Make record, report, or different configurations that contain data created by the framework.

- The yield type of a data framework ought to achieve at least one of the accompanying goals.
- Convey data about past exercises, current status or projections of the Future.

- Signal significant occasions, openings, issues, or alerts.
- Trigger an activity.
- Confirm an activity.

### **Yield Definition**

The yields ought to be characterized as far as the accompanying focuses:

- Type of the yield.
- Content of the yield.
- Format of the yield.
- Location of the yield.
- Frequency of the yield.
- Volume of the yield.
- Sequence of the yield.

It isn't constantly attractive to print or show information as it is hung on a PC. It ought to be chosen as which type of the yield is the most appropriate.

### For Example

- Will decimal focuses be embedded
- Should driving zeros be stifled.

### **Yield Media:**

In the following stage it is to be chosen what medium is the most fitting for the yield. The fundamental contemplations when choosing about the yield media are:

- The appropriateness for the gadget to the specific application.
- The requirement for a printed copy.
- The reaction time required.
- The area of the clients

- The programming and equipment accessible.

Keeping in view the above portrayal the venture is to have yields mostly going under the class of interior yields. The principal yields wanted by the necessity determination are:

The yields were should have been produced as a hot duplicate and just as inquiries to be seen on the screen. Keeping in view these yields, the arrangement for the yield is taken from the yields, which are as of now being gotten after manual preparing. The standard printer is to be utilized as yield media for printed copies.

### **Information Design:**

The information configuration is the connection between the data framework and the client. It contains the creating detail and strategies for information planning and those means are important to put exchange information into a usable structure for handling can be accomplished by investigating the PC to peruse information from a composed or printed record or it can happen by having individuals entering the information straightforwardly into the framework. The plan of information centers around controlling the measure of information required, controlling the mistakes, maintaining a strategic distance from postponement, evading additional means and keeping the procedure straightforward. The info is structured in such a way along these lines, that it furnishes security and usability with holding the protection. Information Design thought about the accompanying things:

- What information ought to be given as information?
- How the information ought to be orchestrated or coded?
- The discourse to direct the working faculty in giving info.
- Methods for getting ready information approvals and ventures to pursue when mistake happen.

### **Destinations**

1. Info Design is the way toward changing over a client situated portrayal of the contribution to a PC based framework. This structure is essential to stay away from blunders in the information input procedure and demonstrate the right heading to the administration for getting right data from the modernized framework.
2. It is accomplished by making easy to understand screens for the information section to deal with expansive volume of information. The objective of structuring input is to make information section simpler and to be free from mistakes. The information passage screen is planned so that every one of the information controls can be performed. It additionally gives record seeing offices.

3. At the point when the information is entered it will check for its legitimacy. Information can be entered with the assistance of screens. Fitting messages are given as when required with the goal that the client won't be in maize of moment. In this way the goal of information configuration is to make an info design that is anything but difficult to pursues

### **Info Stages:**

The fundamental information stages can be recorded as underneath:

- Data recording
- Data interpretation
- Data change
- Data confirmation
- Data control
- Data transmission
- Data approval
- Data remedy

### **Information Types:**

- It is important to decide the different sorts of information sources. Information sources can be classified as pursues:
  - External inputs, which are prime contributions for the framework.
  - Internal inputs, which are client correspondences with the framework.
  - Operational, which are PC office's correspondences to the framework?
  - Interactive, which are inputs entered amid a discourse.

### **Information Media:**

At this stage decision must be made about the info media. To close about the info media thought must be given to.

- Type of information
- Flexibility of arrangement

- Speed
- Accuracy
- Verification strategies
- Rejection rates
- Ease of amendment
- Storage and taking care of prerequisites
- Security
- Easy to utilize
- Portability

Keeping in view the above portrayal of the info types and information media, it tends to be said that the majority of the sources of info are of the type of inside and intelligent. As

Information is to be the straightforwardly entered in by the client, the console can be viewed as the most reasonable info gadget.

### **Error Avoidance**

At this stage care is to be taken to guarantee that input information stays precise structure the phase at which it is recorded up to the phase in which the information is acknowledged by the framework. This can be accomplished just by methods for cautious control each time the information is dealt with.

### **Error Detection**

Despite the fact that each exertion is made to evade the event of blunders, still a little extent of mistakes is in every case liable to happen, these kinds of blunders can be found by utilizing approvals to check the information.

### **Information Validation**

Techniques are intended to distinguish blunders in information at a lower dimension of detail. Information approvals have been incorporated into the framework in pretty much every region where there is a probability for the client to submit blunders. The framework won't acknowledge invalid information. At whatever point an invalid information is entered in, the framework quickly prompts the client, and the client needs to again enter in the information and the framework will

acknowledge the information just if the information is right. Approvals have been incorporated where important.

The framework is intended to be an easy to use one. As it were the framework has been intended to discuss successfully with the client. The framework has been structured with spring up menus.

## **UI Design**

It is fundamental to counsel the framework clients and talk about their requirements while structuring the UI:

UI frameworks can be extensively classified as:

1. Client started interface the client is in control, controlling the advancement of the client/PC discourse. In the PC started interface, the PC chooses the following stage in the collaboration.
2. PC started interfaces.

In the PC started interfaces the PC controls the advancement of the client/PC exchange. Data is shown and the client reaction of the PC makes a move or shows additional data.

## **User Initiated Interfaces**

Client started interfaces fall into two rough classes:

1. Order driven interfaces: In this sort of interface the client inputs directions or inquiries which are translated by the PC.
2. Structures arranged interface: The client calls up a picture of the structure to his/her screen and fills in the structure. The structures arranged interface is picked in light of the fact that it is the best decision.

## **PC Initiated Interfaces**

The accompanying PC – started interfaces were utilized:

1. The menu framework for the client is given a rundown of options and the client picks one; of options.
2. Questions – answer type discourse framework where the PC poses inquiry and takes dependent based on the clients answer.

Directly from the begin the framework will be menu driven, the opening menu shows the accessible alternatives. Picking one alternative gives another popup menu with more choices. Along these lines each choice leads the clients to information section structure where the client can enter in the information.

### **Blunder message structure:**

The plan of mistake messages is a significant piece of the UI structure. As client will undoubtedly submit a few mistakes or other while planning a framework the framework ought to be intended to be useful by giving the client data in regard to the blunder, he/she has submitted.

This application must almost certainly produce yield at various modules for various sources of info.

## **2.2. PERFORMANCE REQUIREMENTS**

Execution is estimated as far as the yield given by the application.

Prerequisite particular has a significant impact in the examination of a framework. Just when the necessity particulars are appropriately given, it is conceivable to structure a framework, which will fit into required condition. It rests to a great extent in the piece of the clients of the current framework to give the necessity determinations since they are the general population who at last utilize the framework. This is on the grounds that the prerequisites must be known amid the underlying stages with the goal that the framework can be structured by those necessities. It is extremely hard to change the framework once it has been structured and then again planning a framework, which does not take into account the necessities of the client, is of no utilization.

The necessity particular for any framework can be extensively expressed as given beneath:

- The framework ought to have the capacity to interface with the current framework
- The framework ought to be exact
- The framework ought to be superior to the current framework

The current framework is totally reliant on the client to play out every one of the obligations.

## **2.3. SOFTWARE REQUIREMENTS**

Operating System	:	Linux
Technology	:	Hadoop
Tools	:	Hive
Reporting Tool	:	Tableau
Java Version	:	JDK1.6 or Higher version

## **2.4. HARDWARE REQUIREMENTS FOR EACH NODE/MACHINE IN A CLUSTER**

Processor	: Intel
Speed	: 2.5 GHz
RAM	: 8 GB or More
Hard Disk	: 500 GB or More

## **3. SYSTEM DESIGN**

### **3.1. ARCHITECTURE DESIGN**

In the wake of breaking down every one of the necessities I have planned and going to actualize the fallowing engineering. As we find in the fallowing figure first we are going to stack the online business datasets into Hadoop HDFS and afterward that information we are going to process with some other huge information innovation called Hive and after that we will create the reports on that.

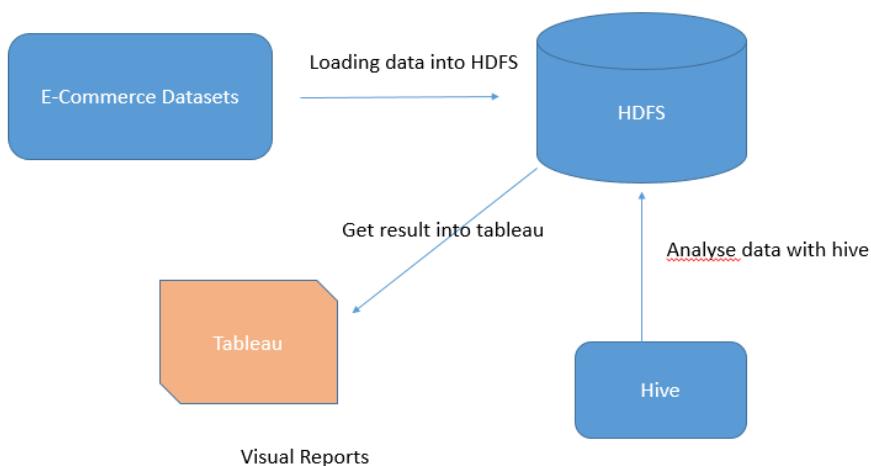


Fig.1: Proposed Architecture

### **3.2. MODULES**

In our undertaking we have the fallowing modules

- Collecting the information and stacking into HDFS
- Analyzing the information
- Creating the reports

## **Module Description**

Gathering the information and stacking into HDFS:

As an underlying advance of doing this task/Big information use case, we have to gather the online business information. There are bunches of web-based business datasets accessible on the web, in light of the fact that pretty much every web-based business application will have some standard arrangement of table's information like client's information, client address, items data, request subtleties, and so on. So we gathered the informational indexes at first and after that by utilizing shell scripting/directions we load the information into HDFS.

Dissecting the information: As the span of informational indexes being gathered and broke down in the online business knowledge is developing quickly, making conventional warehousing arrangements restrictively costly. Hadoop is a famous open-source map-reduce usage which is being utilized as a choice to store and process incredibly expansive informational collections on product equipment. Be that as it may, the guide lessen programming model is exceptionally low dimension and expects engineers to compose custom projects which are difficult to keep up and reuse. In this venture, we are utilizing Hive, an open-source information warehousing arrangement based over Hadoop as appeared in Fig.1. Hive underpins inquiries communicated in a SQL-like revelatory language - HiveQL, which are accumulated into guide decrease employments executed on Hadoop. Furthermore, HiveQL bolsters custom guide diminish contents to be connected to questions.

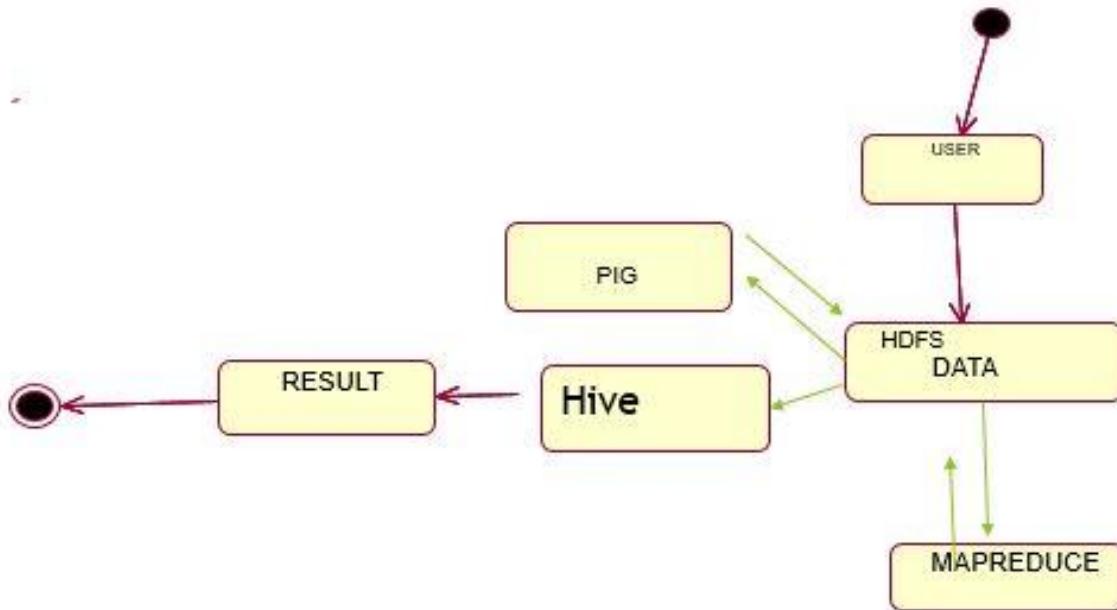
Producing the Reports:

Information is futile if everything it does is sitting in the information distribution center. Subsequently, the introduction layer is of high significance. The vast majority of the OLAP sellers as of now have a front-end introduction layer that enables clients to call up pre-characterized reports or make specially appointed reports. Revealing devices are generally used to make such reports to help basic leadership and measure execution. Organizations use them for money related solidification, for assessment of procedures and approaches and frequently only for plain announcing. In our task we are going to utilize scene as the revealing apparatus which will create the reports in the required configuration (i.e., diagrams, outlines etc...).

### 3.3. UML DIAGRAMS

UML Diagrams for our application are as follows:

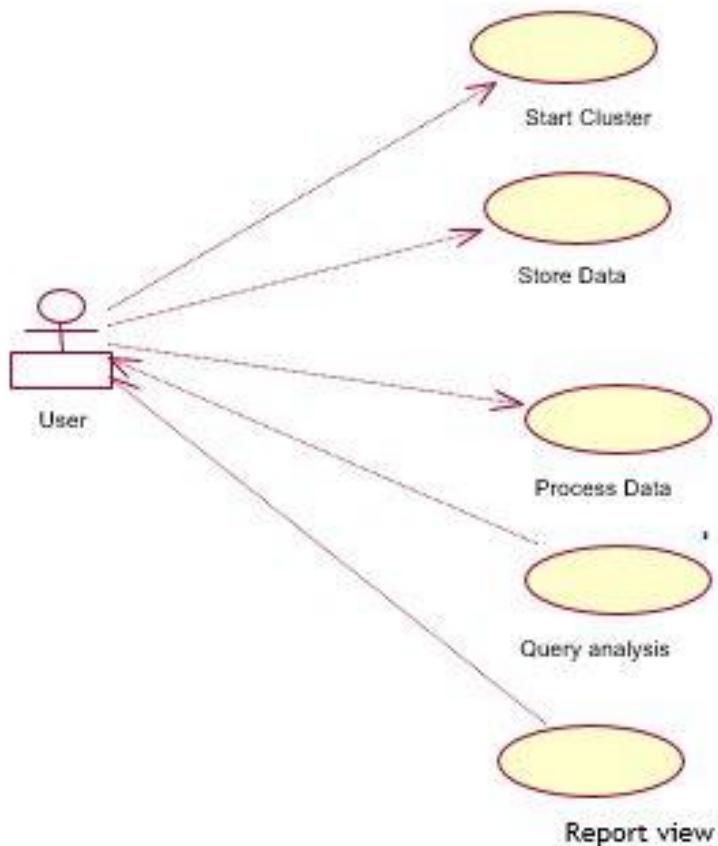
#### 3.3.1. ACTIVITY DIAGRAM



**Fig. 2 Activity diagram**

The activity diagram depicts the main basic structure of the project which tells how the Hadoop and hive and map reduce are linked to produce the desired analysis. As shown in the diagram, the data is loaded into the HDFS through the user. And then this data is used by the MapReduce programs using driver, mapper and reducer classes and the hive tool is also used to analyze data in a simpler way and its result is visualized in tableau in the form of graphs.

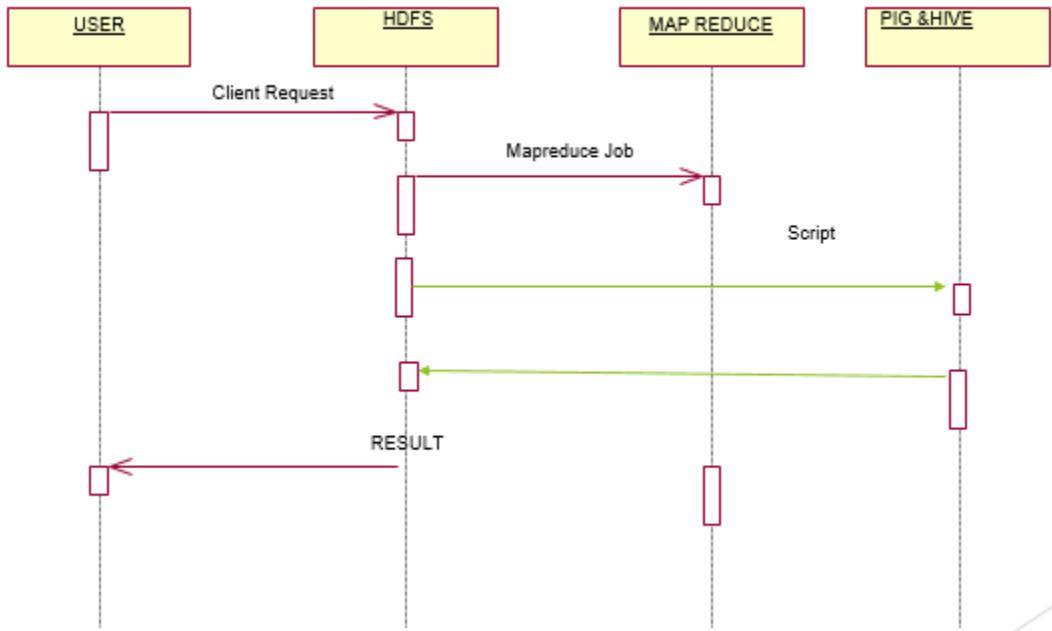
### 3.3.2. USE CASE DIAGRAM



**Fig. 3. Use case diagram**

The use case diagram depicts that the user can start the cluster by forming it and store the data which can be very huge and can process it by using query analysis in hive analyzing various useful data which can be used further to improve the business using various queries and can view the reports produced as the result of the queries.

### 3.3.3. SEQUENCE DIAGRAM



**Fig. 4. Sequence diagram**

The sequence diagram depicts that the user and the HDFS interact with each other when sending the client request from user to HDFS and at last when sending the result back from HDFS to user and the as HDFS stores data, it data is used by the MapReduce to analyze the data and also in the same way by the hive or pig tool in the form of MapReduce job and script respectively.

## 4. SYSTEM IMPLEMENTATION

### Technologies USED:

Coming up next are the advances which we are going to use in our genuine usage of our venture

1. Apache Hadoop
2. Apache Hive
3. Tableau

### Technologies Description:

#### What is Hadoop?

The Apache Hadoop programming library is a structure that takes into consideration the conveyed handling of substantial informational indexes crosswise over bunches of PCs utilizing basic

programming models. It is intended to scale up from single servers to a huge number of machines, each offering neighborhood calculation and capacity. As opposed to depend on equipment to convey high accessibility, the library itself is intended to recognize and deal with disappointments at the application layer, so conveying a very accessible administration over a group of PCs, every one of which might be inclined to disappointments. The center parts of Hadoop are

- HDFS (Hadoop Distributed File System)
- MapReduce Programming Model

### **HDFS (Hadoop Distributed File System):**

The Hadoop Distributed File System (HDFS) is a dispersed record framework intended to keep running on item equipment. It has numerous similitudes with existing dispersed document frameworks. In any case, the distinctions from other dispersed document frameworks are huge. HDFS is very issue tolerant and is intended to be sent on ease equipment. HDFS gives high throughput access to application information and is reasonable for applications that have substantial informational collections. HDFS loosens up a couple POSIX prerequisites to empower gushing access to record framework information. HDFS was initially worked as framework for the Apache Nutch web internet searcher venture. HDFS is a piece of the Apache Hadoop Core venture

### **HDFS suppositions and objectives:**

HDFS is an appropriated document framework intended to deal with expansive informational indexes and keep running on ware equipment. HDFS is profoundly issue tolerant and is intended to be conveyed on minimal effort equipment. HDFS gives high throughput access to application information and is appropriate for applications that have substantial informational collections. HDFS loosens up a couple POSIX prerequisites to empower spilling access to record framework information. HDFS was initially worked as framework for the Apache Nutch web internet searcher venture.

### **Equipment Failure**

One result of scale is that equipment disappointment is the standard instead of the exemption. A HDFS example may comprise of hundreds or thousands of server machines, each putting away piece of the record framework's information. The way that there are countless and that every segment has a non-paltry likelihood of disappointment implies that some part of HDFS is quite often carrying on gravely. Indeed, even with RAID gadgets, disappointments will happen much of the time. In this way, location of issues and snappy, programmed recuperation from them is a center building objective of HDFS.

### **Spilling Data Access**

Applications that keep running on HDFS need spilling access to their informational collections. They are not standard applications that commonly keep running on universally useful document frameworks. HDFS is structured more for cluster handling instead of intuitive use by clients. The

accentuation is on high throughput of information get to instead of low idleness of information get to. POSIX forces numerous hard necessities that are not required for applications that are focused for HDFS. POSIX semantics in a couple of key territories have been loose to pick up an expansion in information throughput rates.

## **Extensive Data Sets**

Applications that keep running on HDFS have extensive informational collections. A run of the mill document in HDFS is gigabytes to terabytes in size. In this manner, HDFS is tuned to help substantial records. It ought to give high total information data transmission and scale to several hubs in a solitary bunch. It should bolster countless records in a solitary occasion.

## **Basic Coherency Model**

HDFS applications need a compose once-read-many access model for records. A document once made, composed, and shut need not be changed aside from adds. This supposition improves information coherency issues and empowers high throughput information get to. A MapReduce application or a web crawler application fits impeccably with this model.

## **HDFS Architecture Design**

The figure underneath gives a run-time perspective on the design indicating three kinds of location spaces: the application, the Name Node and the Data Node. A basic bit of HDFS is that there are various occasions of Data Node.

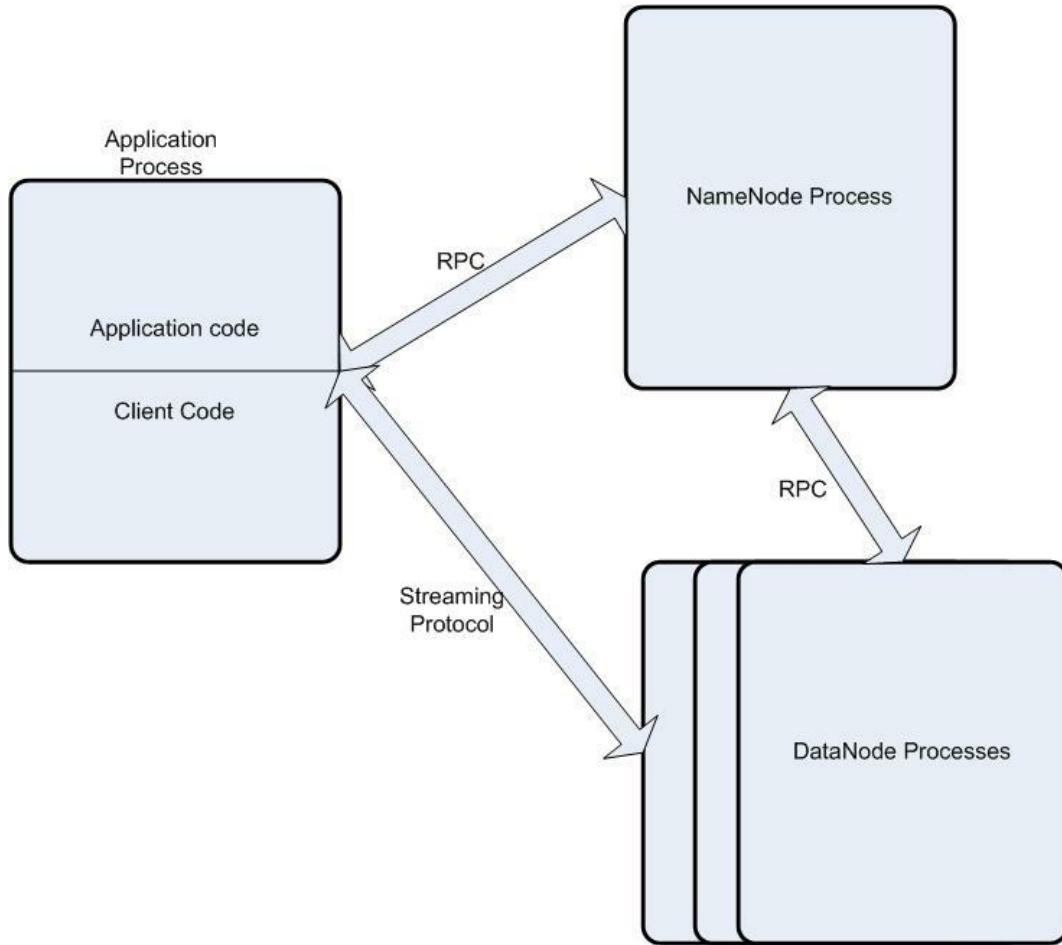


Fig.5. .HDFS Design Architecture

The application fuses the HDFS customer library into its location space. The customer library deals with all correspondence from the application to the Name Node and the Data Node. A HDFS bunch comprises of a solitary Name Node—an ace server that deals with the document framework namespace and controls access to records by customers. What's more, there are various Data Nodes, typically one for each PC hub in the bunch, which oversee capacity joined to the hubs that they keep running on.

The Name Node and Data Node are bits of programming intended to keep running on product machines. These machines regularly run a GNU/Linux working framework (OS). HDFS is fabricated utilizing the Java language; any machine that bolsters Java can run the Name Node or the Data Node programming. Use of the Java language implies that HDFS can be sent on a wide scope of machines. A run of the mill sending has a committed machine that runs just the Name Node programming. Every one of different machines in the bunch runs one example of the Data Node programming. The design does not block running various Data Nodes on a similar machine yet in a genuine arrangement that is once in a while the case.

## HDFS Files

There is a qualification between a HDFS document and a local (Linux) record on the host PC. A PC in a HDFS establishment is (commonly) designated to one Name Node or one Data Node. Every PC has its own record framework and data around a HDFS document—the metadata—is overseen by the Name Node and industrious data is put away in the Name Node's host record framework. The data contained in a HDFS document is overseen by a Data Node and put away on the Data Node's host PC record framework.

HDFS uncovered a record framework namespace and enables client information to be put away in HDFS documents. A HDFS record comprises of various squares. Each square is ordinarily 64MByes. Each square is reproduced some predetermined number of times. The imitations of the squares are put away on various Data Nodes picked to think about stacking a Data Node just as to give both speed in move and flexibility if there should be an occurrence of disappointment of a rack. See Block Allocation for a portrayal of the distribution calculation.

A standard catalog structure is utilized in HDFS. That is, HDFS records exist in registries that may thusly be sub-indexes of different catalogs, etc. There is no understanding of a present catalog inside HDFS. HDFS documents are alluded to by their completely qualified name which is a parameter of a significant number of the components of the association between the Client and different components of the HDFS design.

The Name Node executes HDFS record framework namespace activities like opening, shutting, and renaming documents and indexes. It additionally decides the mapping of squares to Data Nodes. The rundown of HDFS documents having a place with each square, the present area of the square imitations on the Data Nodes, the condition of the record, and the entrance control data is the metadata for the group and is overseen by the Name Node.

The Data Nodes are in charge of serving perused and compose demands from the HDFS document framework's customers. The Data Nodes likewise perform square copy creation, cancellation, and replication upon guidance from the Name Node. The Data Nodes are the judge of the condition of the repeats and they report this to the Name Node.

The presence of a solitary Name Node in a bunch extraordinarily rearranges the engineering of the framework. The Name Node is the referee and storehouse for all HDFS metadata. The customer sends information legitimately to and peruses straightforwardly from Data Nodes with the goal that customer information never courses through the Name Node.

## Square Allocation

Each square is imitated some number of times—the default replication factor for HDFS is three. Whenever addBlock() is conjured, space is dispensed for every copy. Every imitation is allotted on an alternate Data Node. The calculation for playing out this designation endeavors to adjust execution and unwavering quality. This is finished by thinking about the accompanying components:

- The dynamic burden on the arrangement of Data Nodes. Inclination is given to all the more daintily stacked Data Nodes.

- The area of the Data Nodes. Correspondence between two hubs in various racks needs to experience switches. As a rule, arrange data transfer capacity between machines in a similar rack is more prominent than system transmission capacity between machines in various racks.
- For the basic case, when the replication factor is three, HDFS's position arrangement is to put one imitation on one hub in the nearby rack, another on a hub in an alternate (remote) rack, and the keep going on an alternate hub in a similar remote rack. This arrangement cuts the between rack compose traffic which for the most part improves compose execution. The possibility of rack disappointment is far not as much as that of a hub disappointment; accordingly this co-area strategy does not unfavorably affect information unwavering quality and accessibility ensures. In any case, it reduces the total system transmission capacity utilized when perusing information since a square is set in just two novel racks as opposed to three. With this strategy, the copies of a record don't equally convey over the racks. 33% of reproductions are on one hub on some rack; the other 66% of copies are on particular hubs one an alternate rack. This arrangement improves compose execution without trading off information unwavering quality or read execution.

The figure beneath shows how squares are reproduced on various Data Nodes.

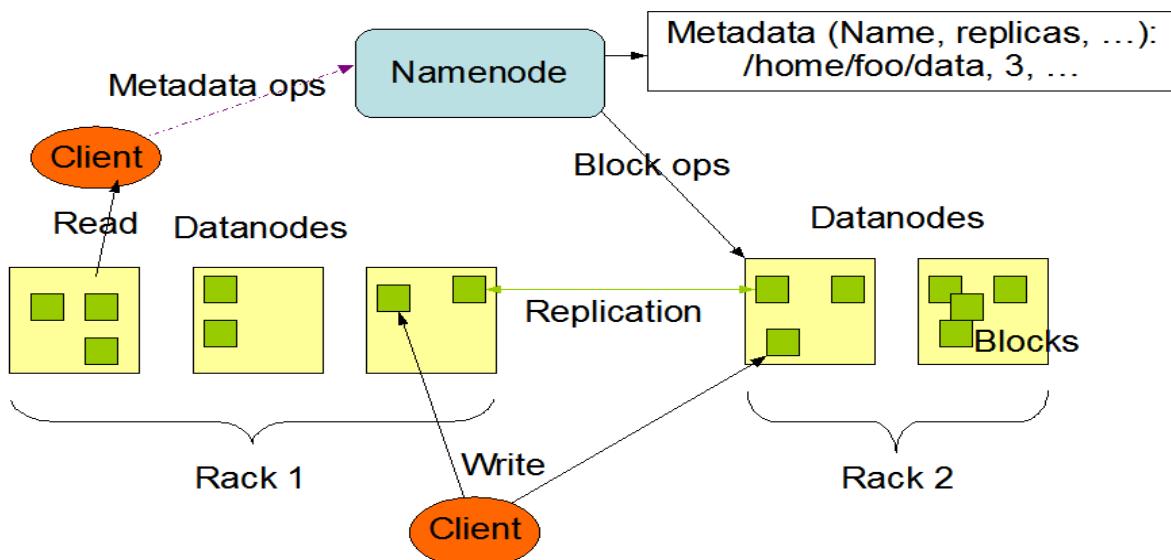


Fig.6..Blocks Replication in Data Nodes

Squares are connected to the document through INode. Each square is given a timestamp that is utilized to decide if an imitation is current.

## MapReduce

Hadoop MapReduce is a product structure for effectively composing applications which process huge measures of information (multi-terabyte informational collections) in-parallel on expansive bunches (a large number of hubs) of item equipment in a solid, flaw tolerant way.

A MapReduce work for the most part parts the information informational collection into autonomous pieces which are handled by the guide undertakings in a totally parallel way. The system sorts the yields of the maps, which are then contribution to the decrease errands. Normally both the info and the yield of the activity are put away in a record framework. The structure deals with booking undertakings, checking them and re-executes the fizzled assignments.

Normally the register hubs and the capacity hubs are the equivalent, that is, the mapreduce structure and the Hadoop Distributed File System (see HDFS Architecture Guide) are running on a similar arrangement of hubs. This setup enables the structure to successfully plan errands on the hubs where information is now present, bringing about high total data transfer capacity over the group.

The MapReduce system comprises of a solitary ace JobTracker and one slave TaskTracker per bunch hub. The ace is in charge of planning the occupations' segment undertakings on the slaves, checking them and re-executing the fizzled errands. The slaves execute the errands as coordinated by the ace.

Negligibly, applications determine the info/yield areas and supply map and lessen capacities by means of usage of proper interfaces as well as unique classes. These, and other occupation parameters, involve the activity setup. The Hadoop work customer at that point presents the activity (container/executable and so on.) and setup to the JobTracker which at that point accepts the accountability of circulating the product/design to the slaves, booking assignments and observing them, giving status and indicative data to the activity customer.

## Sources of info and Outputs

The MapReduce system works solely on `<key, value>` sets, that is, the structure sees the contribution to the activity as a lot of `<key, value>` matches and delivers a lot of `<key, value>` combines as the yield of the activity, possibly of various sorts.

The key and esteem classes must be serializable by the structure and thus need to actualize the Writable interface. Also, the key classes need to actualize the WritableComparable interface to encourage arranging by the system.

Info and Output kinds of a MapReduce work:

(input) `<k1, v1>` -> map -> `<k2, v2>` -> join -> `<k2, v2>` -> decrease -> `<k3, v3>` (yield)

## Hadoop Features

### Versatile

Hadoop is an exceptionally adaptable capacity stage, since it can store and disseminate extremely expansive informational collections crosswise over many economical servers that work in parallel. In contrast to customary social database frameworks (RDBMS) that can't scale to process a lot of

information, Hadoop empowers organizations to run applications on a large number of hubs including a great many terabytes of information.

### **Financially savvy**

Hadoop likewise offers a financially savvy stockpiling answer for organizations' detonating informational collections. The issue with customary social database the executives frameworks is that it is incredibly cost restrictive to scale to such an extent so as to process such monstrous volumes of information. With an end goal to lessen costs, numerous organizations in the past would have needed to down-example information and group it dependent on specific suppositions about which information was the most profitable. The crude information would be erased, as it would be too cost-restrictive to keep. While this methodology may have worked for the time being, this implied when business needs changed, the total crude informational index was not accessible, as it was too costly to even think about storing. Hadoop, then again, is planned as a scale-out design that can moderately store the majority of an organization's information for later use. The cost investment funds are stunning: rather than costing thousands to a huge number of pounds per terabyte, Hadoop offers figuring and capacity abilities for many pounds per terabyte.

### **Adaptable**

Hadoop empowers organizations to effectively get to new information sources and tap into various kinds of information (both organized and unstructured) to create an incentive from that information. This implies organizations can utilize Hadoop to get profitable business bits of knowledge from information sources, for example, web based life, email discussions or clickstream information. What's more, Hadoop can be utilized for a wide assortment of purposes, for example, log handling, proposal frameworks, information warehousing, advertise battle investigation and misrepresentation discovery.

### **Quick**

Hadoop's interesting capacity strategy depends on a conveyed record framework that fundamentally 'maps' information wherever it is situated on a bunch. The instruments for information handling are frequently on similar servers where the information is found, bringing about a lot quicker information preparing. In case you're managing extensive volumes of unstructured information, Hadoop can productively process terabytes of information in not more than minutes, and petabytes in hours.

### **Versatile to disappointment**

A key favorable position of utilizing Hadoop is its adaptation to non-critical failure. At the point when information is sent to an individual hub, that information is additionally repeated to different hubs in the group, which implies that in case of disappointment, there is another duplicate accessible for use. Our design gives security from both single and different disappointments. With regards to taking care of substantial informational collections in a safe and financially savvy way, Hadoop has the preferred standpoint over social database the board frameworks, and its incentive for any size business will keep on expanding as unstructured information keeps on developing.

## **Apache Hive**

Hive is an information distribution center that utilizes MapReduce to break down information put away on HDFS. Specifically, it gives a question language called HiveQL that intently takes after the regular Structured Query Language (SQL) standard.

### **Why Hive ?**

As a matter of fact we are creating MapReduce Programs, we have Hadoop Streaming and clarified that one vast advantage of Streaming is the means by which it permits quicker pivot in the advancement of MapReduce occupations. Hive makes this a stride further. Rather than giving without end of all the more rapidly creating guide and decrease assignments, it offers a question language dependent on the business standard SQL. Hive takes these HiveQL articulations and promptly and consequently makes an interpretation of the questions into at least one MapReduce employments. It at that point executes the general MapReduce program and returns the outcomes to the client. While Hadoop Streaming diminishes the required code/incorporate/submit cycle, Hive evacuates it and rather just requires the arrangement of HiveQL explanations. This interface to Hadoop not just quickens the time required to create results from information examination, it fundamentally widens who can utilize Hadoop and MapReduce. Rather than requiring programming improvement abilities, anybody with a nature with SQL can utilize Hive. The mix of these properties is that Hive is frequently utilized as an apparatus for business and information experts to perform impromptu inquiries on the information put away on HDFS. Direct utilization of MapReduce requires map and lessen undertakings to be composed before the activity can be executed which implies an important postponement from the possibility of a conceivable inquiry to its execution. With Hive, the information examiner can take a shot at refining HiveQL inquiries without the continuous inclusion of a product designer. There are obviously operational and down to earth confinements (a seriously composed question will be wasteful paying little heed to innovation) yet the wide guideline is convincing.

### **Hive Internal Working:**

Hive interior plan incorporates the fallowing

UI - The UI for clients to submit inquiries and different tasks to the framework. As of now the framework has a direction line interface and an electronic GUI is being created.

Driver - The segment which gets the inquiries. This segment actualizes the thought of session handles and gives execute and get APIs demonstrated on JDBC/ODBC interfaces.

Compiler - The segment that parses the inquiry, does semantic investigation on the distinctive question squares and inquiry articulations and in the long run produces an execution plan with the assistance of the table and segment metadata gazed upward from the metastore.

Metastore - The segment that stores all the structure data of the different tables and segments in the stockroom including section and segment type data, the serializers and deserializers important

to peruse and compose information and the comparing hdfs records where the information is put away.

Execution Engine - The segment which executes the execution plan made by the compiler. The arrangement is a DAG of stages. The execution motor deals with the conditions between these distinctive phases of the arrangement and executes these phases on the fitting framework segments.

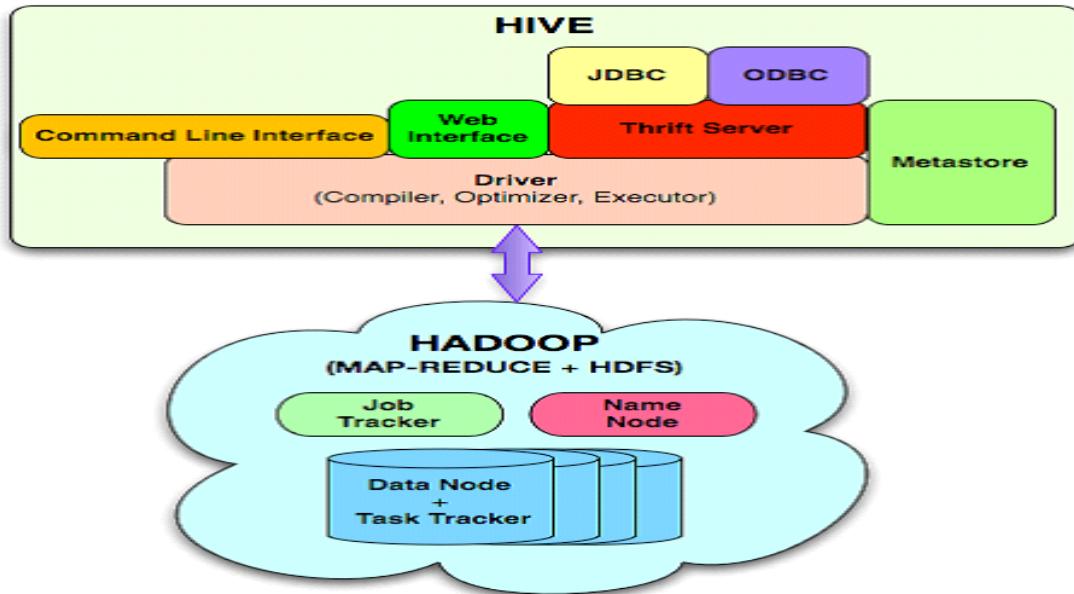


Fig.7..Hive Architecture

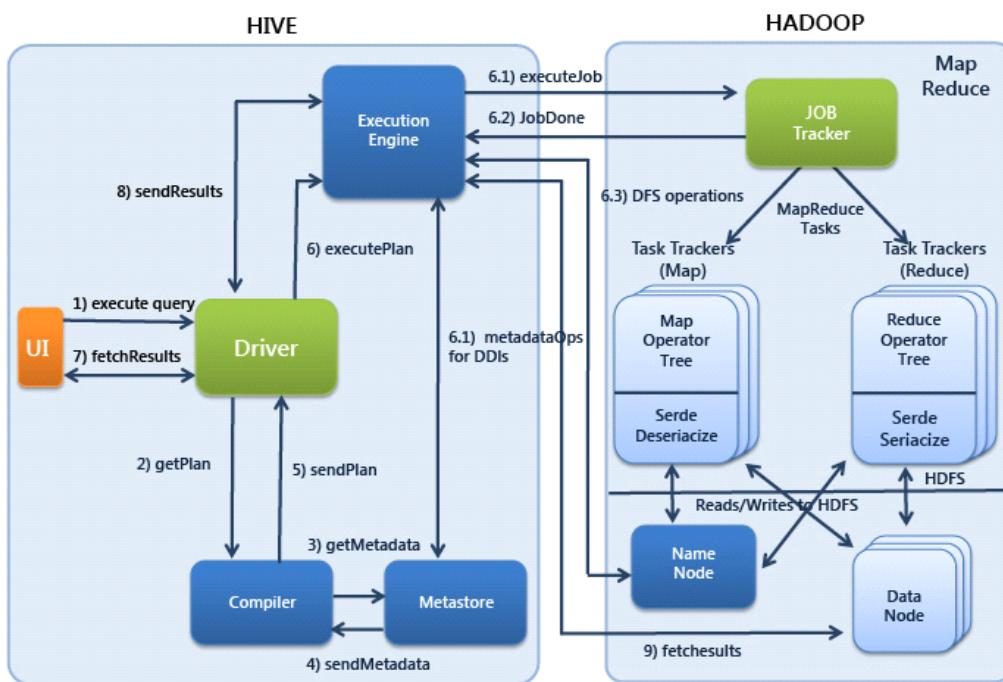


Fig: 8. Hive Job execution flow

Stage 1: The UI calls the execute interface to the Driver.

Stage 2: The Driver makes a session handle for the question and sends the inquiry to the compiler to create an execution plan.

Stage 3: The compiler gets the fundamental metadata from the metastore. This metadata is utilized to type check the articulations in the question tree just as to prune parcels dependent on inquiry predicates.

Stage 5: The arrangement produced by the compiler is a DAG (Direct non-cyclic chart) of stages with each stage being either a guide/decrease work, a metadata activity or a tasks on hdfs. For guide/diminish stages, the arrangement contains map administrator trees (administrator trees that are executed on the mappers) and a decrease administrator tree (for activities that need reducers).

Stage 6: The execution motors presents these phases to suitable segments (stages 6, 6.1, 6.2 and 6.3 strides in above figure). In each errand (mapper/reducer) the deserializer related with the table or middle of the road yields is utilized to peruse the columns from hdfs records and these are gone through the related administrator tree. When the yield is produced, it is kept in touch with a transitory hdfs document however the serializer (this occurs in the mapper in the event that the activity does not require a decrease. The brief records are utilized to give information to consequent guide/decrease phases of the arrangement. For DML activities the last brief document is moved to the table's area. This plan is utilized to guarantee that grimy information isn't perused (record rename being a nuclear activity in hdfs). For questions, the substance of the brief record are perused by the execution motor legitimately from hdfs as a major aspect of the get call from the Driver (stages 7, 8 and 9).

## Scene

In the present focused world associations are expecting an answer that can fulfill their data needs in only a single tick; settle business questions in a moment or two; with choice to redo according to hierarchical necessities. What Information Technology offers is a scope of Business Intelligence Tools that help associations in taking business choices all the more successfully and effectively.

Out of 100s of Business Intelligence Tools accessible in the market, I encountered scene and trust me it's simply magnificent. This BI Tool accompanies numerous rich highlights. You essentially need to move things and you will get what you really need, that too in simply part of seconds.

Scene offers four business insight items viz.

- Tableau Desktop
- Tableau Server
- Tableau Digital
- Tableau Public

## **Key differentiators**



Fig.9. Key Features of Tableau

## **Instrument for everybody**

Regularly, it has come to see that, there exists a hole between the report engineer and business chief, for example the person who devours the report. Scene connects this hole, as it were, by offering report building capacity which a layman can use to build up a Business Intelligence Solution. This aides in limiting the reliance on IT assets. The essential and a standout amongst the best highlights of Tableau is you need not be a specialized master. Business clients like chiefs, office heads, deals and promoting heads, deals agents, senior administration can without much of a stretch and viably use Tableau to create and produce reports to fulfill their data need while taking business choices.

## **Quick**

Scene is one of the quickest Business Intelligence Tool accessible in the market. Scene can make reports in minutes which may take hours in Microsoft Excel. One can interface with information source and in couple of moment's time one can have alluring graphical representations to look over.

## **Maps**

Scene's in-manufactured geocoding highlight makes Map a center piece of Tableau. At whatever point clients have information identified with regular regions like nations, states or post codes, they need not enter scope or longitude information to find that territory on the guide. Alongside solid information representation, scene gives us adaptability to recount stories with maps and to utilize maps to bore down into related data.

## **Interactive Data Visualization**

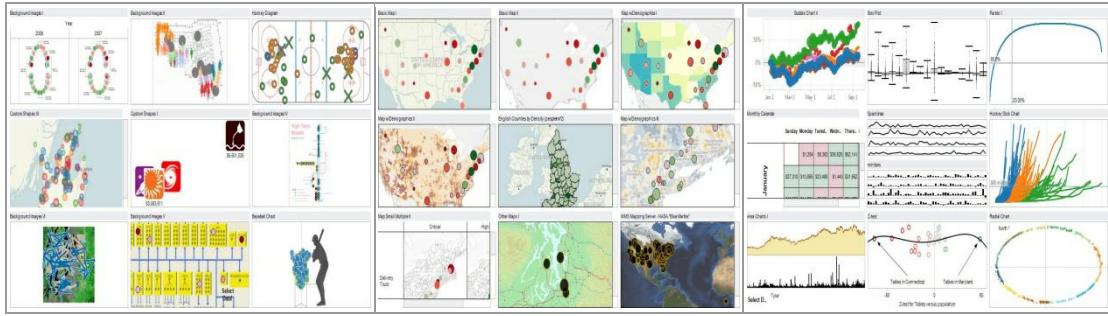


Fig.10. .Data Visualization Formats

Tableau is one of the intuitive Business Intelligence Reporting Tools accessible in the market. Scene comprehends the kind of information associated and recommends reasonable representation. Highlights like "Show Me" prescribes the appropriate information perception dependent on the chose information.

When it is associated with a database, Tableau consequently fragments information into measurements and measures.

Scene offers huge number of information representations by perusing the information which helps in finding huge business discoveries. Scene likewise offers boring through the information, subsequently influencing it conceivable to reach to the least and most natty gritty data conceivable. Adaptability

Scene gives adaptability to pick how you work with information. One can legitimately interface with database or can use in memory innovation. In memory can be utilized when database is moderate or one needs to work disconnected or when live database association is beyond the realm of imagination.

Scene has a capacity to associate straightforwardly with practically a wide range of databases accessible in the business. This fundamentally helps in decreasing time and endeavors for totaling the information and bringing into nearby information stockroom. Having direct network with different information sources, additional time can be spent in examination. Scene additionally comprehends and expands all the local capacities accessible in the associated database.

## **Prudent**

Scene is a standout amongst the most efficient Business Intelligence Tools accessible in the market. Associations can likewise set aside extra cash as there is less reliance on extra IT assets.

## **Sharing of data**

Scene Server offers disconnected and internet distributing of the information representation which makes the data trade quick, basic and viable. Scene Server shares data alongside fundamental outline information through messages alongside different information trades groups.

## **Looks after Security**

Scene regards security conventions. Each client has an exceptional id and secret key that limits the entrance to business data.

## IMPLEMENTATION

The main period of our task is introducing hadoop bunch. The best possible methodology would be a 2 stage methodology. Initial step is introduce single-hub Hadoop machines, arrange and test them as nearby Hadoop frameworks. Second step is combine that solitary hub frameworks into a multi-hub group. So first given us a chance to perceive how to setup the single hub machine.

### SINGLE NODE SETUP

Hadoop is bolstered by GNU/Linux stage and its flavors. In this manner, we need to introduce a Linux working framework for setting up Hadoop condition. On the off chance that you have an OS other than Linux, you can introduce a Virtual box programming in it and have Linux inside the Virtual box. So first we introduced VM and did the setup of Ubuntu OS.

#### Pre-installation Setup

Before installing Hadoop into the Linux environment, we need to set up Linux using **ssh** (Secure Shell). Follow the steps given below for setting up the Linux environment.

##### Creating a User

At the beginning, it is recommended to create a separate user for Hadoop to isolate Hadoop file system from UNIX file system. Follow the steps given below to create a user –

- Open the root using the command “su”.
- Create a user from the root account using the command “useradd username”.
- Now you can open an existing user account using the command “su username”.

Open the Linux terminal and type the following commands to create a user.

```
$ su  
password:  
# useradd hadoop  
# passwd hadoop  
New password:  
Retype new password
```

#### SSH Setup and Key Generation

SSH setup is required to do different operations on a cluster such as starting, stopping, distributed daemon shell operations. To authenticate different users of Hadoop, it is required to provide public/private key pair for a Hadoop user and share it with different users.

The following commands are used for generating a key value pair using SSH. Copy the public keys from id\_rsa.pub to authorized\_keys, and provide the owner with read and write permissions to authorized\_keys file respectively.

```
$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

```
hdpuser@hdpuser-VirtualBox:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
/home/hdpuser/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/hdpuser/.ssh/id_rsa.
Your public key has been saved in /home/hdpuser/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:HW6Comi7eD3SuWEnFM2vdjsgHlVeunQNdTiqnotvoe8 hdpuser@hdpuser-VirtualBox
The key's randomart image is:
+---[RSA 2048]---+
|      .... |
|      o . o o. |
|      . oo o.+ . |
|      ..o+oo.. |
|      o...S++ |
|      . oo...+o |
|      ...+++++.o |
|      o +.*=.o= |
|      .+...o.=Eo |
+---[SHA256]---+
hdpuser@hdpuser-VirtualBox:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Let's verify key based login. Below command should not ask for the password but the first time it will prompt for adding RSA to the list of known hosts.

```
$ ssh localhost
$ exit
```

## Setting up Java

Download jdk-8u201-linux-x64.tar from oracle jdk download website.

After downloading the tar file, move into the downloaded folder and extract the tar file by using below command

```
$tar -xvf jdk-8u201-linux-x64.tar
```

Next, we need to set environment variables used by Hadoop. Edit ~/.bashrc file and append following values at end of file as shown below to setup java.

```

hdpuser@hdpuser-VirtualBox:~$ 
hdpuser@hdpuser-VirtualBox:~$ sudo gedit ~/.bashrc
[sudo] password for hdpuser:

** (gedit:11921): WARNING **: 21:33:14.725: Set document metadata failed: Setting attribu
hdpuser@hdpuser-VirtualBox:~$ sudo gedit ~/.bashrc
[Open] [Save] *.bashrc ~/
# ~./.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/home/hdpuser/installations/jdk-8/jdk1.8.0_201

```

Now you can verify java setup by executin the below command

\$java –version

```

hdpuser@hdpuser-VirtualBox:~$ java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
hdpuser@hdpuser-VirtualBox:~$
```

## Setting up Hadoop

**Step 1 )** [Download Hadoop](#) from the [Apache Download Mirrors](#) and extract the contents of the Hadoop package to a location of your choice. I have picked **/home/hdpuser/installations/**.

Extract the hadoop tar file by using below command and move into the selected directory

```

$tar -xvf Downloads/hadoop-2.7.7.tar
$cp -R Downloads/hadoop-2.7.7 /home/hdpuser/installations
```

Now we need to edit **.basrc** file and set the hadoop variable as shown in below screen

```

hdpuser@hdpuser-VirtualBox:~$ gedit ~/.bashrc
hdpuser@hdpuser-VirtualBox:~$ ./.bashrc
.bashrc
~/
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/._.bash_aliases ]; then
    . ~/._.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/home/hdpuser/installations/jdk-8/jdk1.8.0_201
export HADOOP_HOME=/home/hdpuser/installations/hadoop-2.7.7
export HIVE_HOME=/home/hdpuser/installations/apache-hive-1.2.2
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin

sh ▾ Tab Width: 8 ▾ Ln 122, Col 83 ▾ INS

```

## Edit Configuration Files

Hadoop has many of configuration files, which need to configure as per requirements to setup Hadoop infrastructure. Let's start with the configuration with basic Hadoop single node cluster setup. first, navigate to below location

```
$cd $HADOOP_HOME/etc/hadoop
```

```

hdpuser@hdpuser-VirtualBox:~/installations/hadoop-2.7.7$ cd etc/hadoop/
hdpuser@hdpuser-VirtualBox:~/installations/hadoop-2.7.7/etc/hadoop$ ls
capacity-scheduler.xml      hadoop-metrics2.properties   httpfs-signature.secret  log4j.properties          ssl-client.xml.example
configuration.xml           hadoop-metrics.properties   httpfs-site.xml            mapred-env.cmd          ssl-server.xml.example
container-executor.cfg       hadoop-policy.xml        kms-acls.xml             mapred-env.sh          yarn-env.cmd
core-site.xml                hdfs-site.xml          kms-env.sh              mapred-queues.xml.template  yarn-env.sh
hadoop-env.cmd               httpfs-env.sh         kms-log4j.properties     mapred-site.xml        yarn-env.sh
hadoop-env.sh                httpfs-log4j.properties   kms-site.xml            slaves
hdpuser@hdpuser-VirtualBox:~/installations/hadoop-2.7.7/etc/hadoop$ 

```

The following files will have to be modified to complete the Hadoop setup

**/home/hdpuser/installations/hadoop-2.7.7/etc/hadoop/hadoop-env.sh**

**/home/hdpuser/installations/hadoop-2.7.7/etc/hadoop/core-site.xml**

**/home/hdpuser/installations/hadoop-2.7.7/etc/hadoop/mapred-site.xml.template**

**/home/hdpuser/installations/hadoop-2.7.7/etc/hadoop/hdfs-site.xml**

**/home/hdpuser/installations/hadoop-2.7.7/etc/hadoop/yarn-site.xml**

Open **core-site.xml** and edit the property mentioned below inside configuration tag:  
**core-site.xml** informs Hadoop daemon where Name Node runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & Mapreduce. Edited and added properties as shown below

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:8020</value>
    </property>
</configuration>
```

Edit **hdfs-site.xml** and edit the property mentioned below inside configuration tag:  
**hdfs-site.xml** contains configuration settings of HDFS daemons (i.e. Name Node, Data Node, Secondary Name Node). It also includes the replication factor and block size of HDFS.

```
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/home/hdpuser/hadoop-data/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/home/hdpuser/hadoop-data/data</value>
  </property>
</configuration>
```

Edit the **mapred-site.xml** file and edit the property mentioned below inside configuration tag:  
**mapred-site.xml** contains configuration settings of Mapreduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the mapred-site.xml file using mapred-site.xml template.

```
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Edit **yarn-site.xml** and edit the property mentioned below inside configuration tag:  
**yarn-site.xml** contains configuration settings of Resource Manager and Node Manager like application memory management size, the operation needed on program & algorithm, etc.

```

    limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>

</configuration>

```

i.

Edit **hadoop-env.sh** and add the Java Path as mentioned below:

**hadoop-env.sh** contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

```

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
export JAVA_HOME=/home/hdpuser/installations/jdk-8/jdk1.8.0_201

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol. Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#----- JAVA_HOME & JSVCS HOME

```

Format the HDFS File system and Starting Hadoop server

The first step to starting your Hadoop installation is the formatting of the Hadoop file system (HDFS) implemented on top of your local file system of your cluster. This step is required the first time you set up a Hadoop cluster. Do not format a running Hadoop file system as you will lose all the data currently in the cluster (in HDFS)!

- To format the file system (which simply initializes the directory specified by the `dfs.namenode.name.dir` variable), execute the following command in the `$HADOOP_HOME/bin` directory

`$hdfs namenode -format`

```

hduser_@guru99-VirtualBox:~$ $HADOOP_HOME/bin/hdfs namenode -format
14/05/05 13:01:58 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = guru99-VirtualBox/127.0.1.1
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.2.0
STARTUP_MSG: classpath = /home/guru99/Downloads/hadoop/etc/hadoop:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/activation-1.1.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/netty-3.6.2.Final.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/protobuf-java-2.5.0.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/xmlenc-0.52.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/jsp-api-2.1.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/commons-collections-3.2.1.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/avro-1.7.4.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/jackson-core-asl-1.8.8.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/commons-io-2.1.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/jersey-core-1.9.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/commons-codec-1.4.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/mockito-all-1.8.5.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/jets3t-0.6.1.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/guava-11.0.2.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/commons-net-3.1.jar:/home/guru99/Downloads/hadoop/share/hadoop/common/lib/commons-httpclient-3.1.jar:/home/guru99/Download

```

Start Hadoop single node cluster using below command

\$start-all.sh

```

hduser@hdpuser-VirtualBox:~$ jps
13083 Jps
hduser@hdpuser-VirtualBox:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hdpuser/installations/hadoop-2.7.7/logs/hadoop-hdpuser-namenode-hdpuser-VirtualBox.out

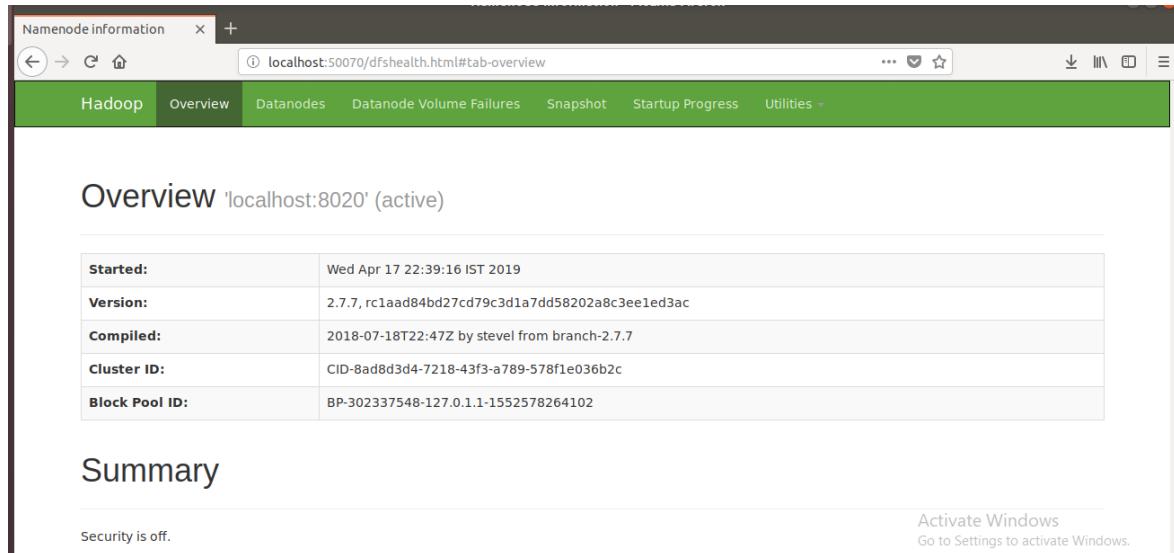
```

```

hduser@hdpuser-VirtualBox:~$ jps
13083 Jps
hduser@hdpuser-VirtualBox:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hdpuser/installations/hadoop-2.7.7/logs/hadoop-hdpuser-namenode-hdpuser-VirtualBox.out
localhost: starting datanode, logging to /home/hdpuser/installations/hadoop-2.7.7/logs/hadoop-hdpuser-datanode-hdpuser-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/hdpuser/installations/hadoop-2.7.7/logs/hadoop-hdpuser-secondarynamenode-hdpuser-VirtualBox.out
starting yarn daemons
starting resourcemanager, logging to /home/hdpuser/installations/hadoop-2.7.7/logs/yarn-hdpuser-resourcemanager-hdpuser-VirtualBox.out
localhost: starting nodemanager, logging to /home/hdpuser/installations/hadoop-2.7.7/logs/yarn-hdpuser-nodemanager-hdpuser-VirtualBox.out
hduser@hdpuser-VirtualBox:~$ 
hduser@hdpuser-VirtualBox:~$ jps
14002 NodeManager
13426 DataNode
13844 ResourceManager
14040 Jps
13625 SecondaryNameNode
13273 NameNode
hduser@hdpuser-VirtualBox:~$ 

```

Now we can browse the hdfs using our web browser with URL  
<http://localhost:50070>



Namenode information +

localhost:50070/dfshealth.html#tab-overview

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Overview 'localhost:8020' (active)

Started:	Wed Apr 17 22:39:16 IST 2019
Version:	2.7.7, rc1aad84bd27cd79c3d1a7dd58202a8c3ee1ed3ac
Compiled:	2018-07-18T22:47Z by stevel from branch-2.7.7
Cluster ID:	CID-8ad8d3d4-7218-43f3-a789-578f1e036b2c
Block Pool ID:	BP-302337548-127.0.1.1-1552578264102

Summary

Security is off.

Activate Windows  
Go to Settings to activate Windows.

If you want to stop the hadoop services we can do that with the below command

```
$stop-all.sh
```

```
hdpuuser@hdpuuser-VirtualBox:~$ jps
3585 NodeManager
2995 DataNode
3427 ResourceManager
2836 NameNode
12331 Jps
3197 SecondaryNameNode
5086 org.eclipse.equinox.launcher_1.3.201.v20161025-1711.jar
hdpuuser@hdpuuser-VirtualBox:~$ stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
stopping yarn daemons
stopping resourcemanager
localhost: stopping nodemanager
no proxyserver to stop
```

Now the hadoop single node cluster is ready and running. Next step is to configure Hive.

# HIVE CONFIGURATION

## Introduction

Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. Apache Hive supports analysis of large datasets stored in Hadoop's HDFS and compatible file systems such as Amazon S3 file system. It provides an SQL-like language called HiveQL(Hive Query Language) while maintaining full support for map/reduce.

### Installing HIVE:

- Browse to the link: <http://apache.claz.org/hive/stable/> to download the hive
- Click the apache-hive-1.2.2-bin.tar.gz
- Save and Extract it

## Commands

```
hdpuser@hdpuser-VirtualBox:~$tar -xvf Downloads/apache-hive-1.2.2-bin.tar.gz
```

```
hdpuser@hdpuser-VirtualBox:~$ sudo mkdir /home/hdpuser/installations/apache-hive-1.2.2  
hdpuser@hdpuser-VirtualBox:~$ cp -R Downloads/ apache-hive-1.2.2/*  
/home/hdpuser/installations/apache-hive-1.2.2/
```

### Setting Hive environment variable:

## Commands

```
hdpuser@hdpuser-VirtualBox:~$ cd  
hdpuser@hdpuser-VirtualBox:~$ sudo gedit ~/.bashrc  
Copy and paste the following lines at end of the file
```

```
# Set HIVE_HOME  
export HIVE_HOME=/home/hdpuser/installations/apache-hive-1.2.2  
PATH=$PATH:$HIVE_HOME/bin  
export PATH
```

### Setting HADOOP\_PATH in HIVE config.sh

## Commands

```
hdpuser@hdpuser-VirtualBox:~$ cd /usr/lib/hive/apache-hive-0.13.0-bin/bin  
hdpuser@hdpuser-VirtualBox:~$ sudo gedit hive-config.sh  
Go to the line where the following statements are written
```

```
# Allow alternate conf dir location.  
HIVE_CONF_DIR="${HIVE_CONF_DIR:-$HIVE_HOME/conf}"  
export HIVE_CONF_DIR=$HIVE_CONF_DIR  
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH  
Below this write the following
```

```
export HADOOP_HOME=/usr/local/hadoop (write the path where hadoop file is there)
```

## Create Hive directories within HDFS

Command

```
hdpuser@hdpuser-VirtualBox:~$ hadoop fs -mkdir /usr/hive/warehouse
```

## Setting READ/WRITE permission for table

Command

```
hdpuser@hdpuser-VirtualBox:~$ hadoop fs -chmod g+w /usr/hive/warehouse
```

## HIVE launch

Command

```
hdpuser@hdpuser-VirtualBox:~$ hive
```

Hive shell will prompt:

## OUTPUT

Shell will look like

```
hdpuser@hdpuser-VirtualBox:~$  
hdpuser@hdpuser-VirtualBox:~$ hive  
Logging initialized using configuration in jar:file:/home/hdpuser/installations/apache-hive-1.2.2/lib/hive-common-1.2.2.jar!/hive-log4j.properties  
hive> ■
```

## Creating a database

Command

```
hive> create database mydb;
```

OUTPUT

OK

Time taken: 0.369 seconds

```
hive>
```

If we get the above output we can confirm that the have has been configured successfully.

## DATA PROCESSING TASK

Up to now we have configured the hadoop and its eco systems/tools required to get that data into hadoop and process that data. So now the first thing to do here is getting the data into HDFS and then analyze using hive. By using **load data local inpath** query of hive we directly load data from local file system to HDFS which will be directly pointed to hive table that we load data into.

Below are the scripts to load data into Hive and Analyze the data.

### Analyzing the data with Hive

Now we have the data in hadoop distributed file system (HDFS), so we have to load and process this data in hive now. For this we have to execute the following script in Hive Shell

First enter into Hive shell using the following command

```
$Hive
```

Shell will look like

```
hdpuser@hdpuser-VirtualBox:~$ hdpuser@hdpuser-VirtualBox:~$ hive  
Logging initialized using configuration in jar:file:/home/hdpuser/installations/apache-hive-1.2.2/lib/hive-common-1.2.2.jar!/hive-log4j.properties  
hive> ■
```

In this Hive shell we have to run our following scripts

### SCRIPTS TO RUN

```
create table customers(customer_id bigint,first_name string,last_name string,user_id  
string,password string,email string,gender string) row format delimited fields terminated  
by ',' tblproperties ("skip.header.line.count"="1");
```

```
load data local inpath '/home/hdpuser/Desktop/ecommerce-data/Customers.csv' into table  
customers;
```

```
create table customer_address(customer_id bigint,street_address string,city string,state  
string,country string,zipcode bigint,contact_number string)row format delimited fields  
terminated by ',' tblproperties ("skip.header.line.count"="1");
```

```
load data local inpath '/home/hdpuser/Desktop/ecommerce-data/Address.csv' into table  
customer_address;
```

```
create table orders(order_id bigint,customer_id bigint,order_date timestamp,order_status  
string)row format delimited fields terminated by ',' ;
```

```
load data local inpath '/home/hdpuser/Desktop/ecommerce-data/orders.csv' into table  
orders;
```

```
create table orders_details(order_details_id bigint,order_id bigint,product_id bigint)row  
format delimited fields terminated by ',' ;
```

```
load data local inpath '/home/hdpuser/Desktop/ecommerce-data/order_details.csv' into  
table orders_details;
```

```
create table products(product_id bigint,product_name string,product_category_tree  
string,retail_price int,discounted_price int,product_rating double,brand string) ROW  
FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' tblproperties  
("skip.header.line.count"="1");
```

```
load data local inpath '/home/hdpuser/Desktop/ecommerce-data/ProductsData.csv' into  
table products;
```

---below 2 queries are for preprocessing

```
create or replace view products_cat as select product_id,  
product_name,split(substr(product_category_tree,3,length(product_category_tree)-4)," >>  
") as category_tree,retail_price ,discounted_price, product_rating, brand from products;
```

```
create or replace view products_sub_cat as select cast(product_id as bigint) product_id,  
product_name,category_tree[0] category_name,category_tree[1]  
sub_category1,category_tree[2] sub_category2,cast(retail_price as int)  
retail_price,cast(discounted_price as int) discounted_price, cast (product_rating as float)  
product_rating, brand from products_cat;
```

```
set hive.groupby.orderby.position.alias=true;
```

```
--  
select a.category_name,count(*) total from products_sub_cat a,orders o,orders_details od  
where o.order_id=od.order_id and od.product_id=a.product_id group by  
a.category_name;
```

```
select a.category_name,count(*) total from products_sub_cat a,orders o,orders_details od  
where o.order_id=od.order_id and od.product_id=a.product_id group by a.category_name  
order by total desc limit 10;
```

```
*select a.category_name,sum(discounted_price) total from products_sub_cat a,orders  
o,orders_details od where o.order_id=od.order_id and od.product_id=a.product_id group  
by a.category_name order by total limit 10;
```

```
*select a.product_name,sum(discounted_price) total from products_sub_cat a,orders  
o,orders_details od where category_name ='Mobiles & Accessories' and  
o.order_id=od.order_id and od.product_id=a.product_id group by a.product_name order  
by total limit 10;
```

```
*select a.product_name,count(*) total from products_sub_cat a,orders o,orders_details od where category_name ='Mobiles & Accessories' and o.order_id=od.order_id and od.product_id=a.product_id group by a.product_name;
```

```
select city,count(*) from customers c,customer_address ca where ca.customer_id=c.customer_id group by city;
```

```
select state,count(*) from customers c,customer_address ca where ca.customer_id=c.customer_id group by state;
```

```
select city,sum(discounted_price) from products_sub_cat a,customer_address ca,orders o,orders_details od where o.order_id=od.order_id and od.product_id=a.product_id and o.customer_id = ca.customer_id group by city;
```

```
select state,sum(discounted_price) from products_sub_cat a,customer_address ca,orders o,orders_details od where o.order_id=od.order_id and od.product_id=a.product_id and o.customer_id = ca.customer_id group by state;
```

```
select if(month(order_date)<=3,'Q1',if(month(order_date)<=6,'Q2',if(month(order_date)<=9,'Q3','Q4'))) as Quarter,count(*) from orders group by 1;
```

```
select if(month(order_date)<=3,'Q1',if(month(order_date)<=6,'Q2',if(month(order_date)<=9,'Q3','Q4'))) as Quarter,count(*) from orders group by if(month(order_date)<=3,'Q1',if(month(order_date)<=6,'Q2',if(month(order_date)<=9,'Q3','Q4')));
```

```
*select state,sum(discounted_price) total_sum from products_sub_cat a,customer_address ca,orders o,orders_details od where o.order_id=od.order_id and od.product_id=a.product_id and o.customer_id = ca.customer_id group by state order by total_sum desc limit 10;
```

```
*select city,sum(discounted_price) total_sum from products_sub_cat a,customer_address ca,orders o,orders_details od where o.order_id=od.order_id and od.product_id=a.product_id and o.customer_id = ca.customer_id group by city order by total_sum desc limit 10;
```

```
select date_format(order_date,'MMMM') month,count(*) from orders group by date_format(order_date,'MMMM');
```

```
*select state,sum(discounted_price) from products_sub_cat a,customer_address ca,orders o,orders_details od where o.order_id=od.order_id and od.product_id=a.product_id and o.customer_id = ca.customer_id and a.category_name='clothing' group by state limit 10;
```

```
[  
select sub_category1,sum(discounted_price) from products_sub_cat a, customer_address  
ca,orders o,orders_details od where o.order_id=od.order_id and  
od.product_id=a.product_id and o.customer_id = ca.customer_id and  
a.category_name='Clothing' group by sub_category1;
```

```
select product_name,product_rating from products_sub_cat where  
category_name='Clothing' and product_rating > 4
```

---to export the query results to local file which is used in tableau reports

```
insert overwrite local directory '/home/hdpuser/Desktop/hiveoutput' row format delimited  
fields terminated by ',' select if(month(order_date)<=3,'Q1',if(month(order_date)<=6,'Q2',  
if(month(order_date)<=9,'Q3','Q4'))) as Quarter,count(*) from orders group by  
if(month(order_date)<=3,'Q1',if(month(order_date)<=6,'Q2',  
if(month(order_date)<=9,'Q3','Q4')));
```

```
insert overwrite local directory '/home/hdpuser/Desktop/hiveoutput/statewise_customers'  
row format delimited fields terminated by ',' select state,count(*) from customers  
c, customer_address ca where ca.customer_id=c.customer_id group by state;
```

```
insert overwrite local directory '/home/hdpuser/Desktop/hiveoutput/categorywise_sale/'  
row format delimited fields terminated by ',' select a.category_name,count(*) total from  
products_sub_cat a,orders o,orders_details od where o.order_id=od.order_id and  
od.product_id=a.product_id group by a.category_name order by total desc limit 10;
```

```
insert overwrite local directory '/home/hdpuser/Desktop/hiveoutput/categorywise_sale/'  
row format delimited fields terminated by ' @@ ' select  
a.category_name,sum(discounted_price) total from products_sub_cat a,orders  
o,orders_details od where o.order_id=od.order_id and od.product_id=a.product_id group  
by a.category_name order by total limit 10;
```

## 5. OUTPUT SCREENS

### TABLES

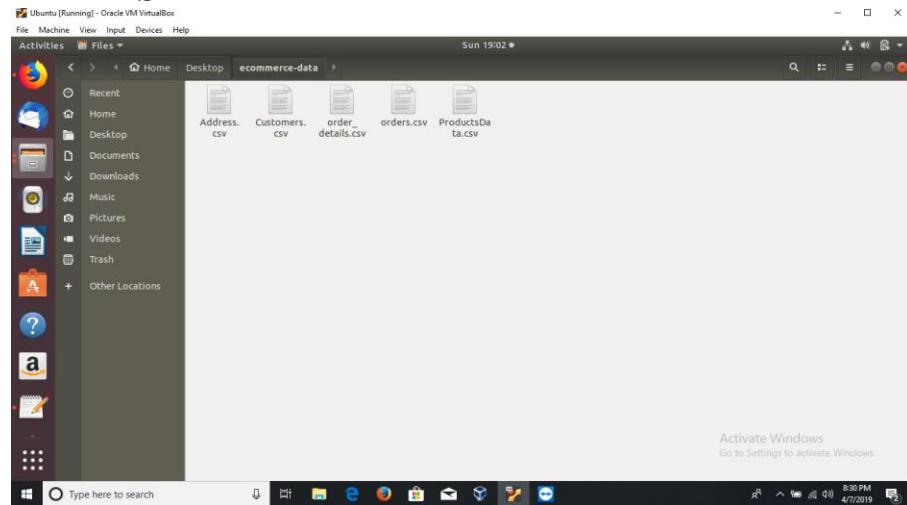


Fig. 11. Tables

### STRUCTURE OF TABLES

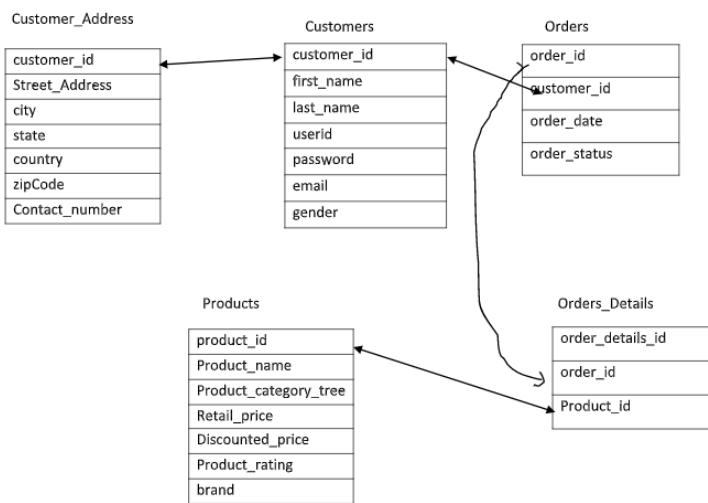


Fig.12. structure of tables

## STATEWISE ORDERS

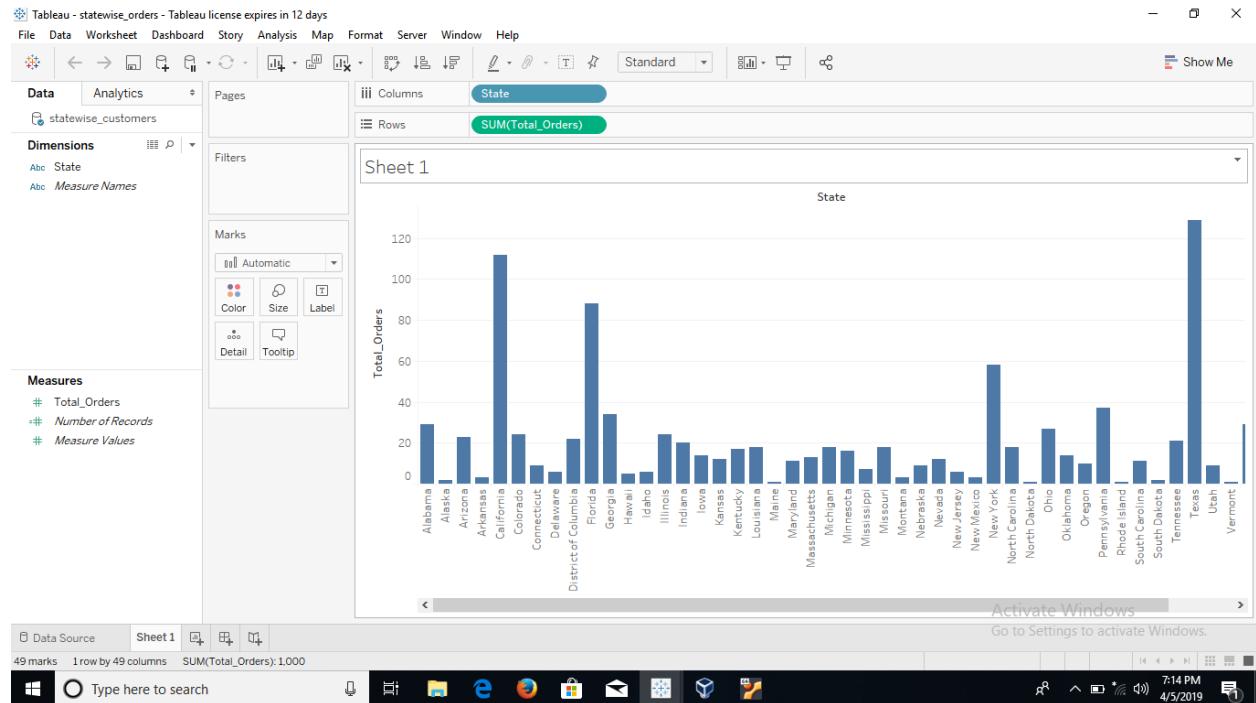


Fig. 13. State wise orders

## TOTAL AMOUNT COLLECTED FOR TOP TEN CATEGORIES

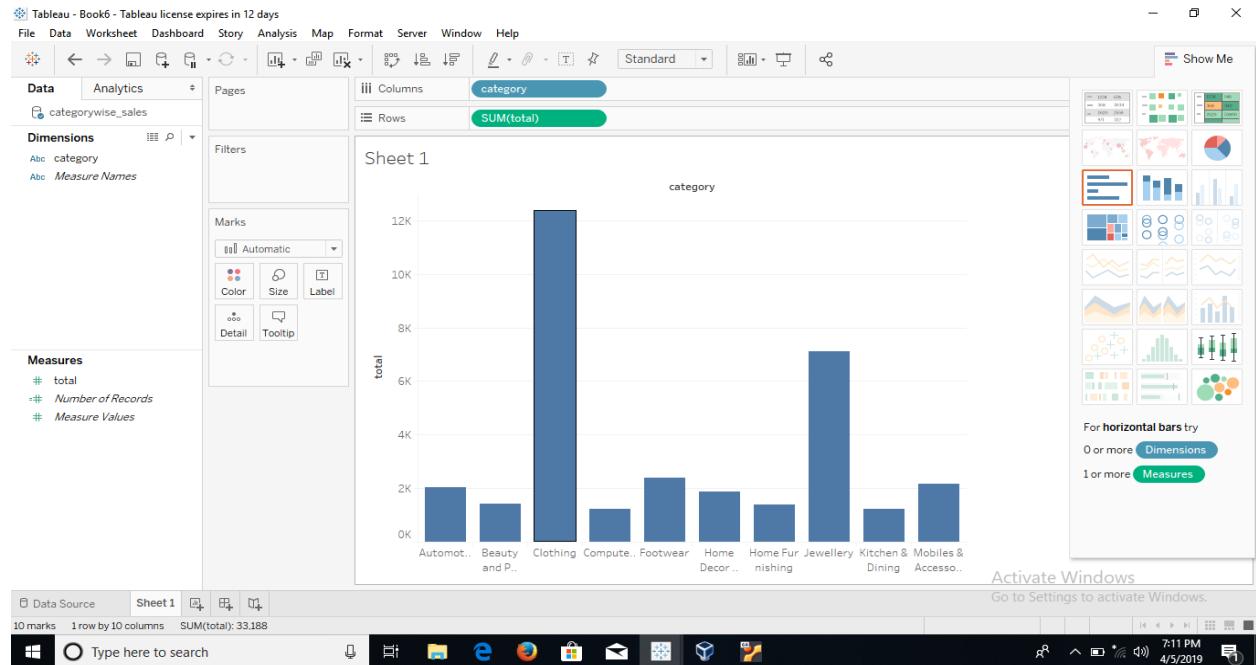


Fig. 14. Total amount collected for top ten categories

## QUARTER WISE SALES

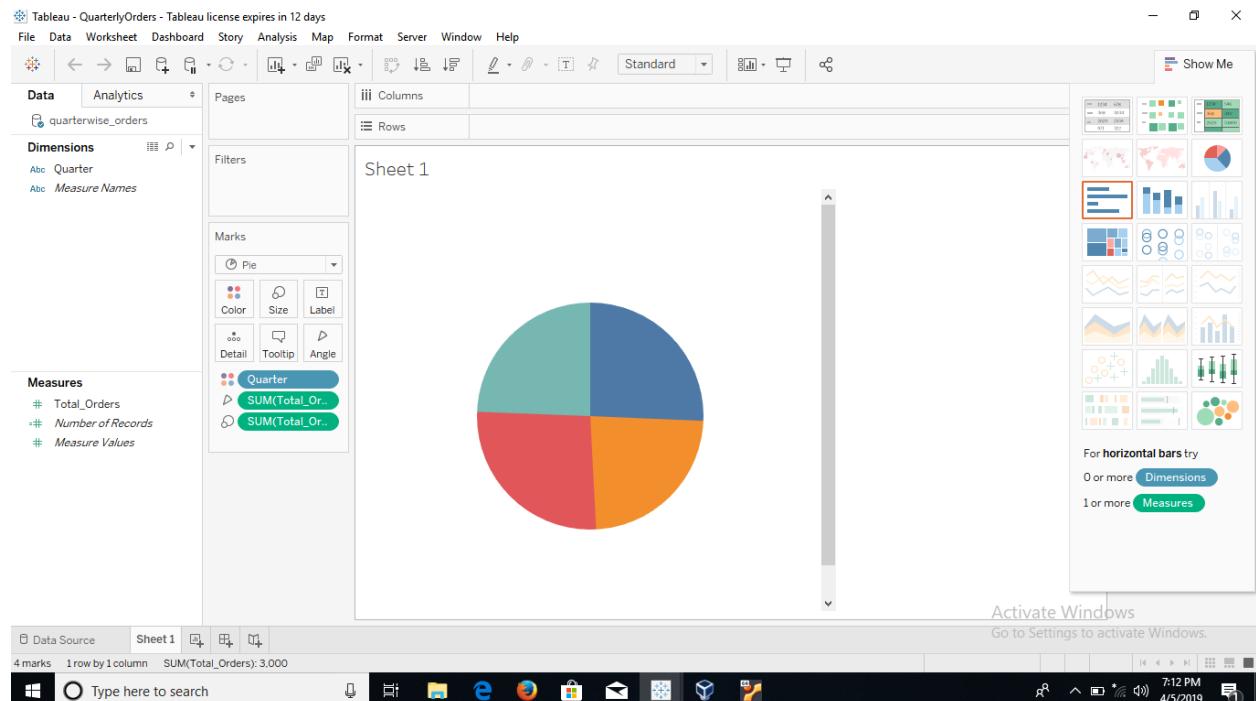


Fig. 15. Quarter wise sales

## TOP TEN CITIES WITH HIGHEST NUMBER OF ORDERS

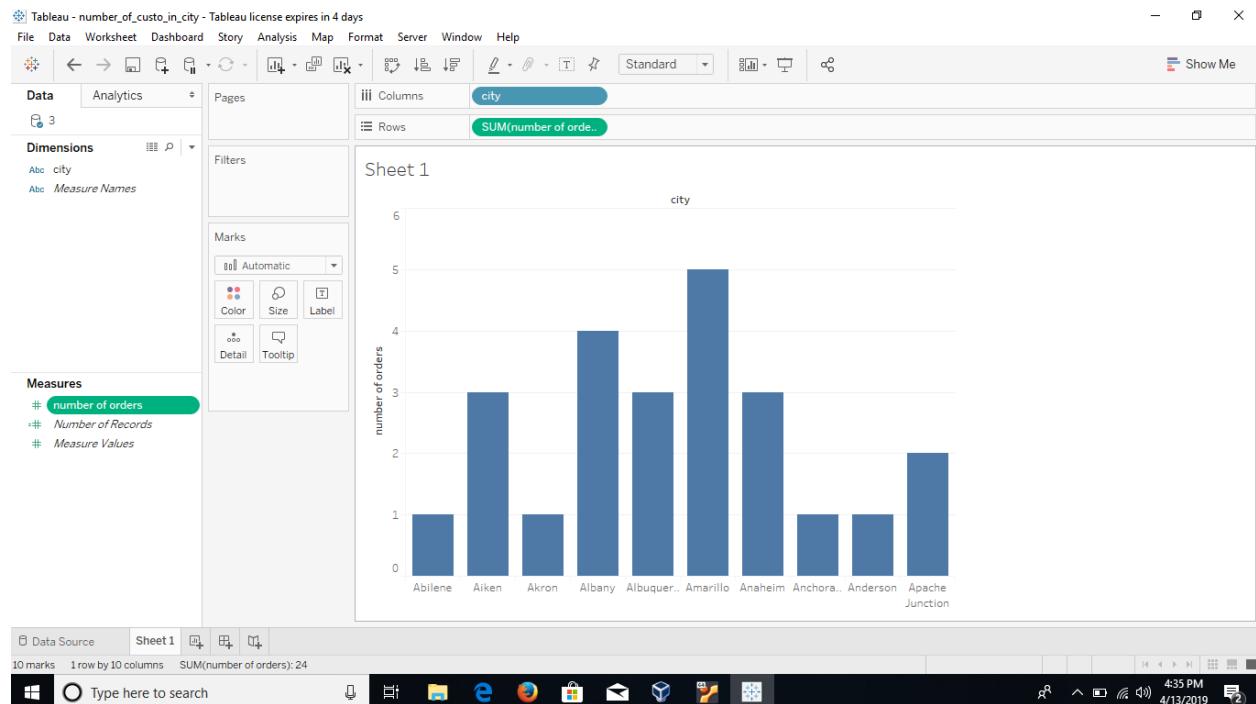


Fig. 16. Top ten cities with highest number of orders

## TOP TEN STATES WITH HIGHEST NUMBER OF ORDERS

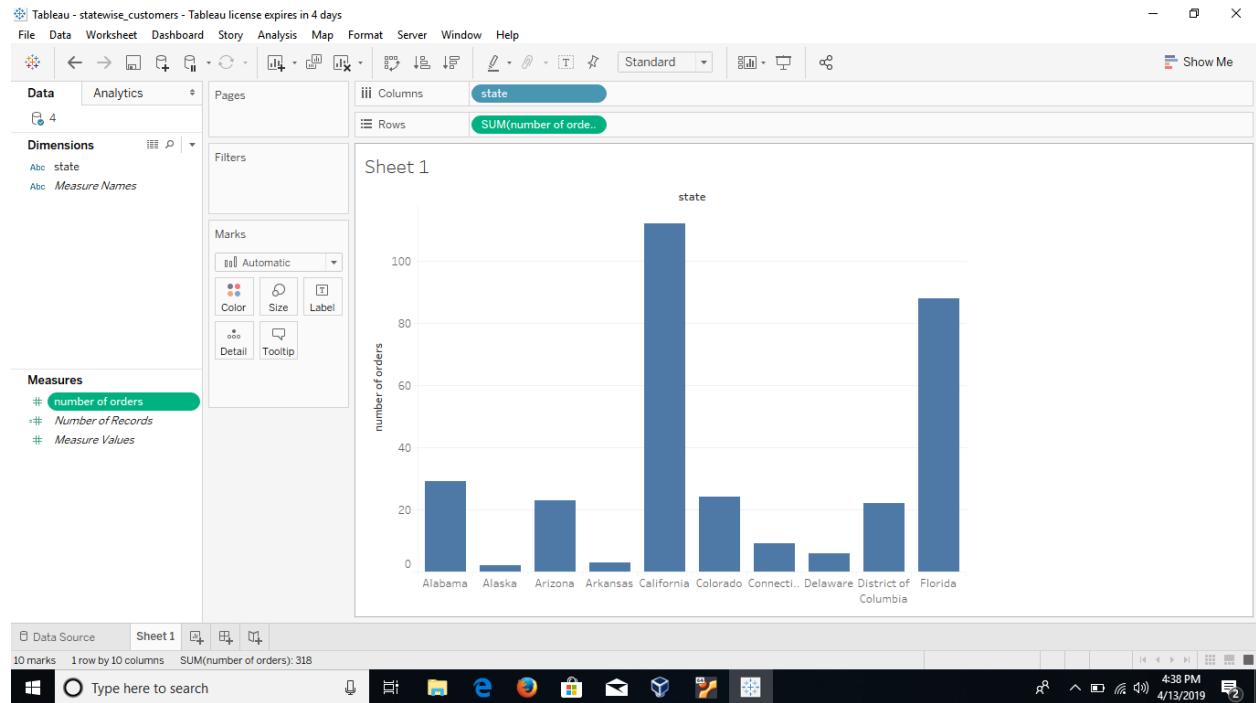
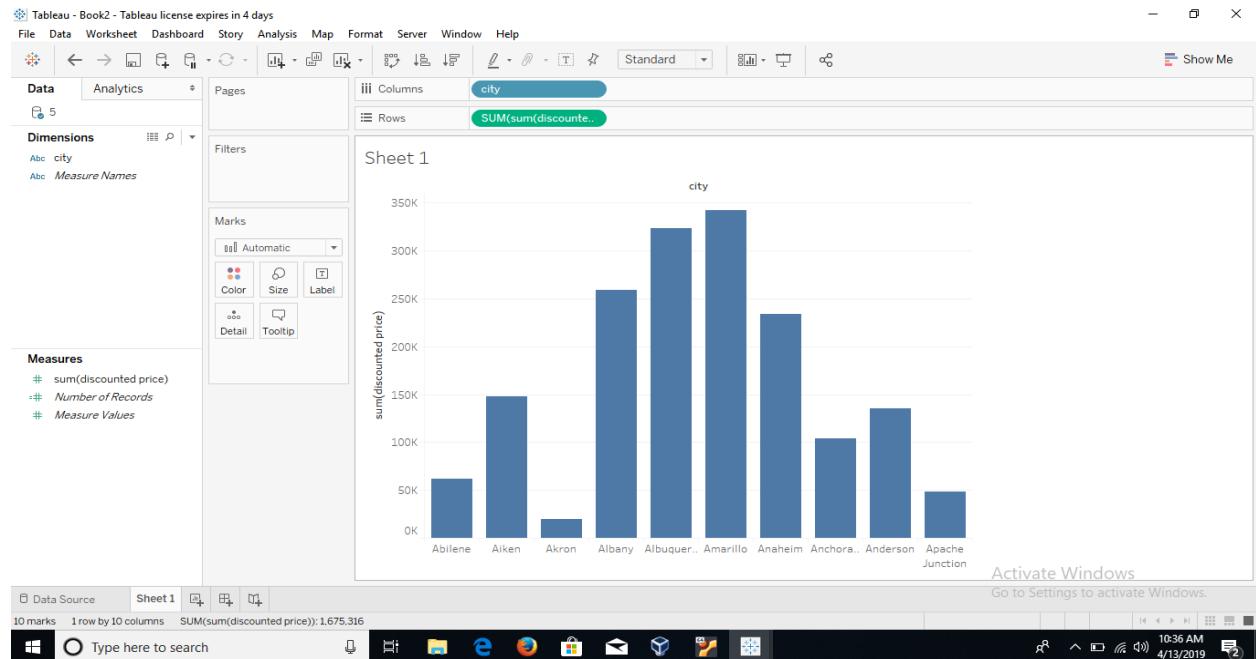


Fig. 17. Top ten states with highest number of orders

## TOP TEN CITIES WITH HIGHEST DISCOUNTED PRICE



## TOP TEN STATES WITH HIGHEST DISCOUNTED PRICE

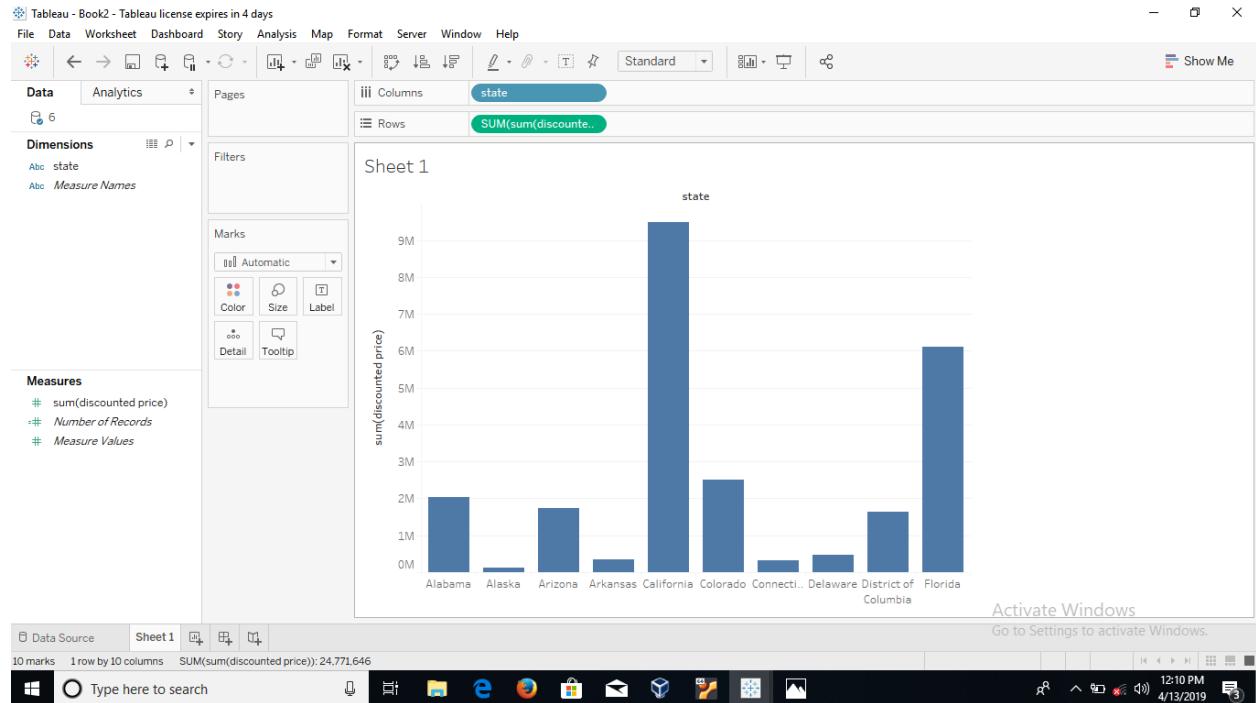


Fig. 19. Top ten states with highest discounted price

## TOTAL AMOUNT COLLECTED ON SALES EACH MONTH

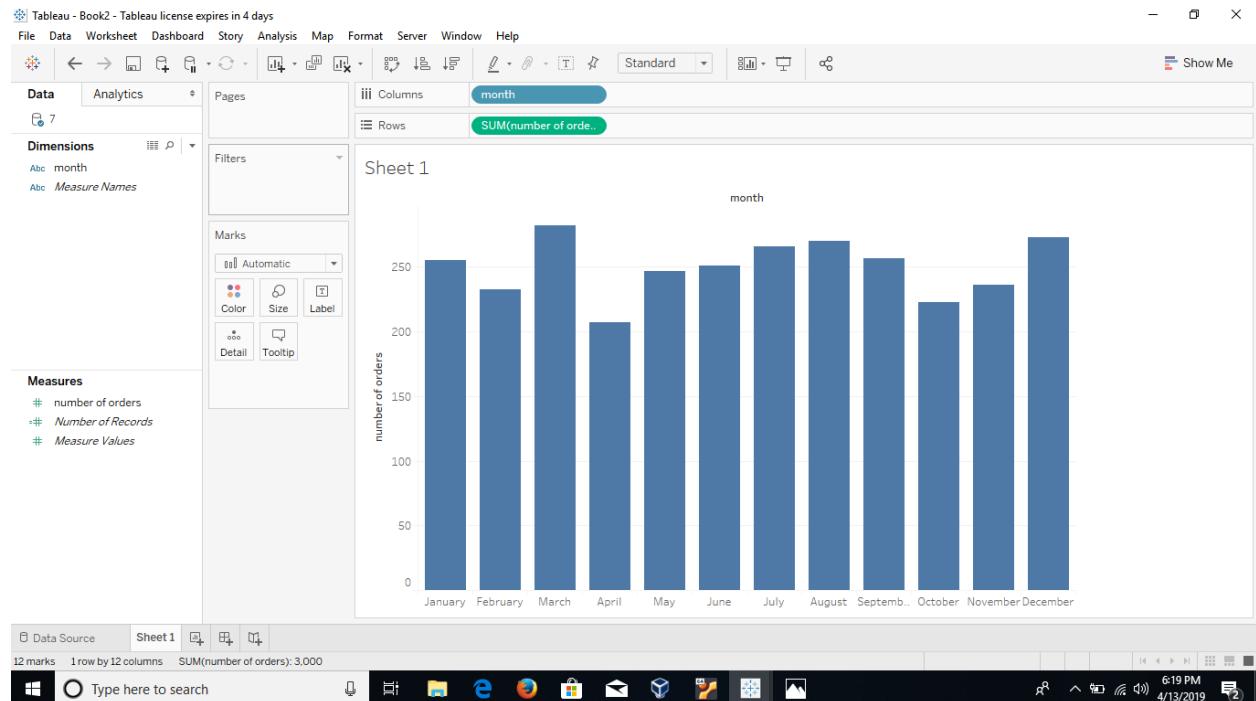


Fig. 20. Total amount collected on sales each month

## TOP TEN HIGHEST CATEGORIES IN SUB CATEGORY

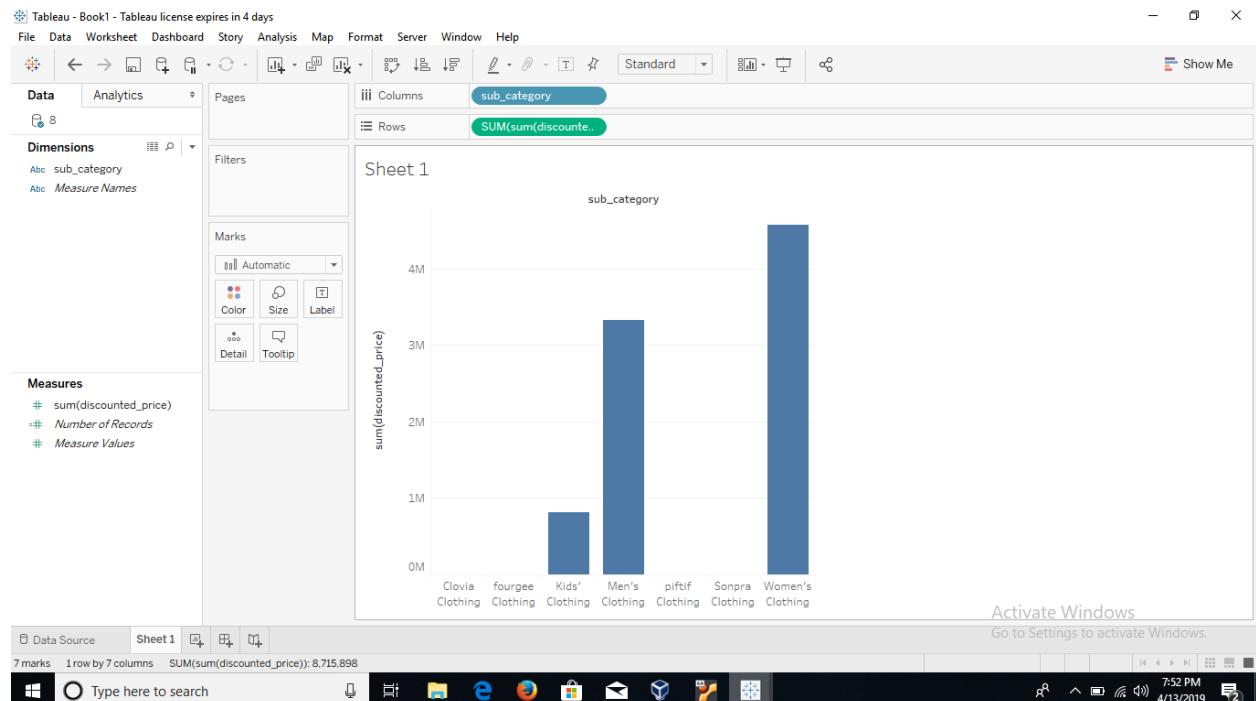


Fig. 21. Top ten highest categories in sub category

## TOP TEN HIGHEST RATING PRODUCTS

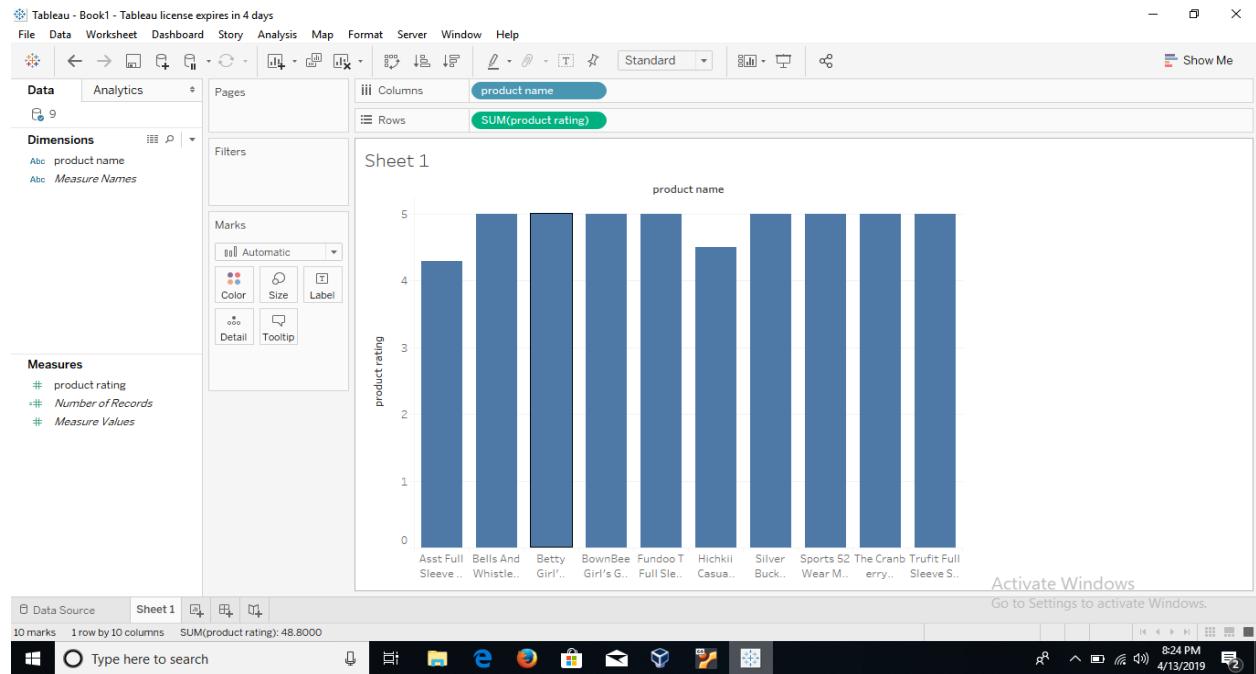


Fig. 22. Top ten highest rating products

## TOP TEN PRODUCTS WITH HIGHEST DISCOUNTED PRICE

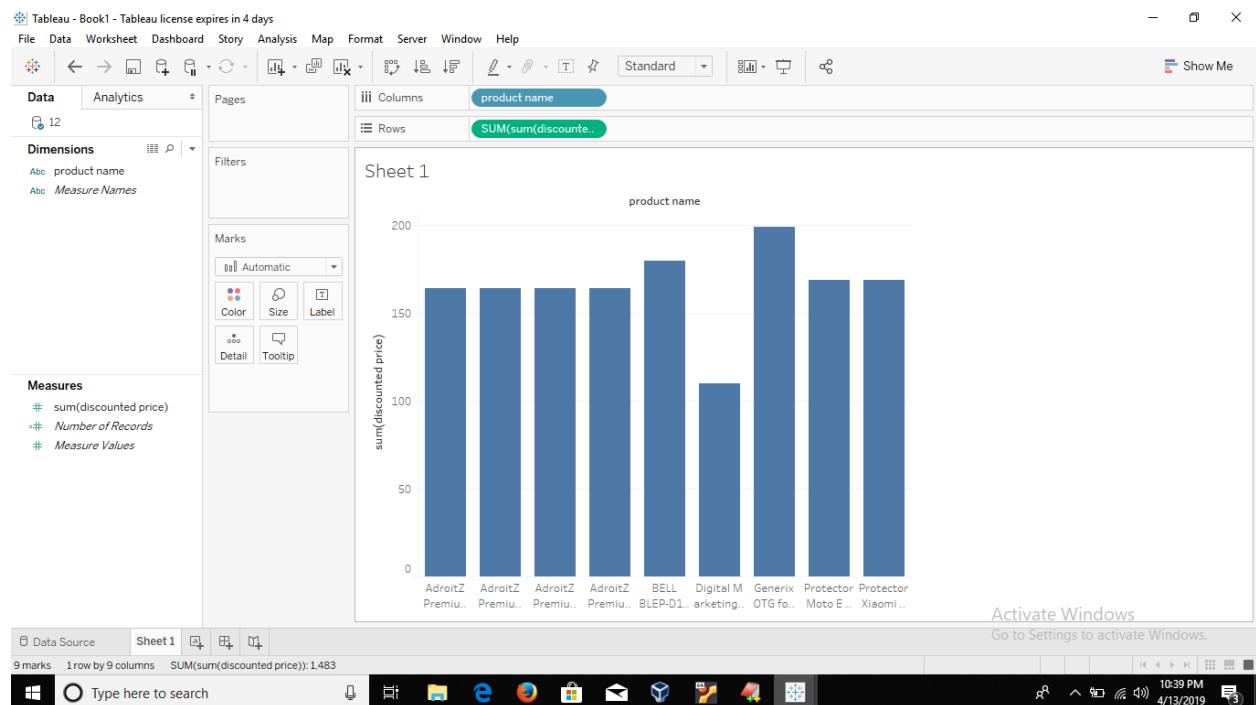


Fig. 23. Top ten products with highest discounted price

## TOP TEN CATEGORIES HAVING HIGHEST RECORDS

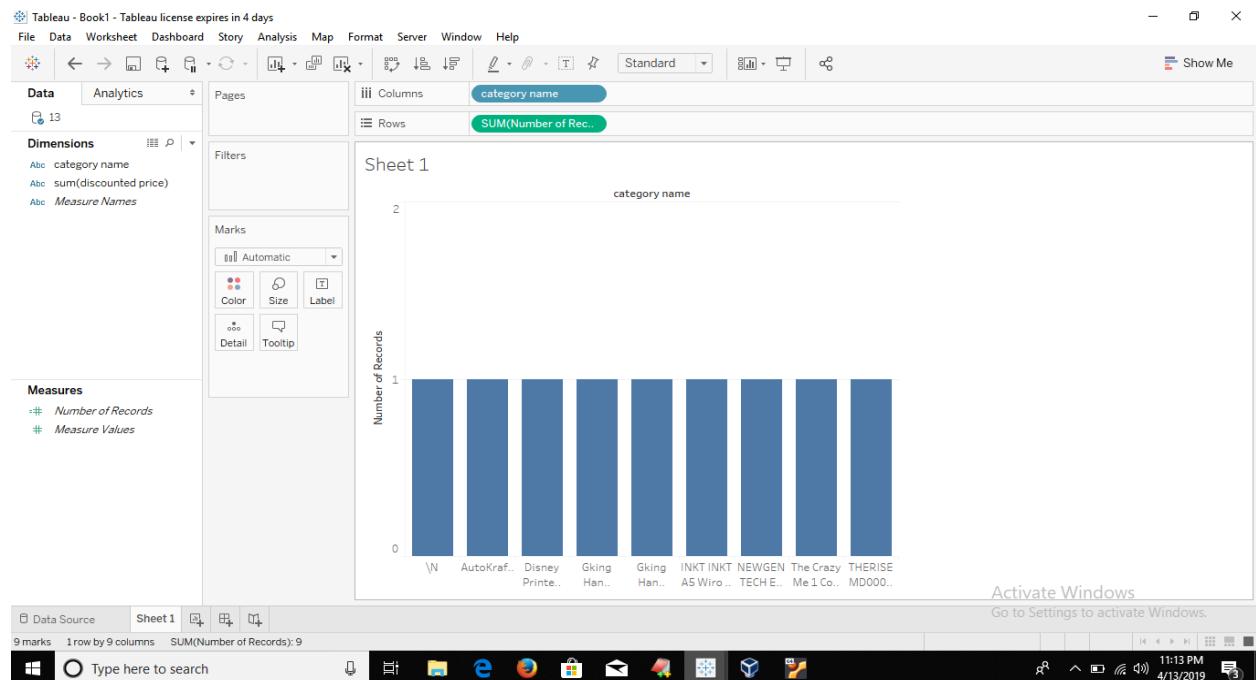


Fig. 24. Top ten categories having highest records

## TOP TEN PRODUCTS WITH HIGHEST DISCOUNTED PRICE

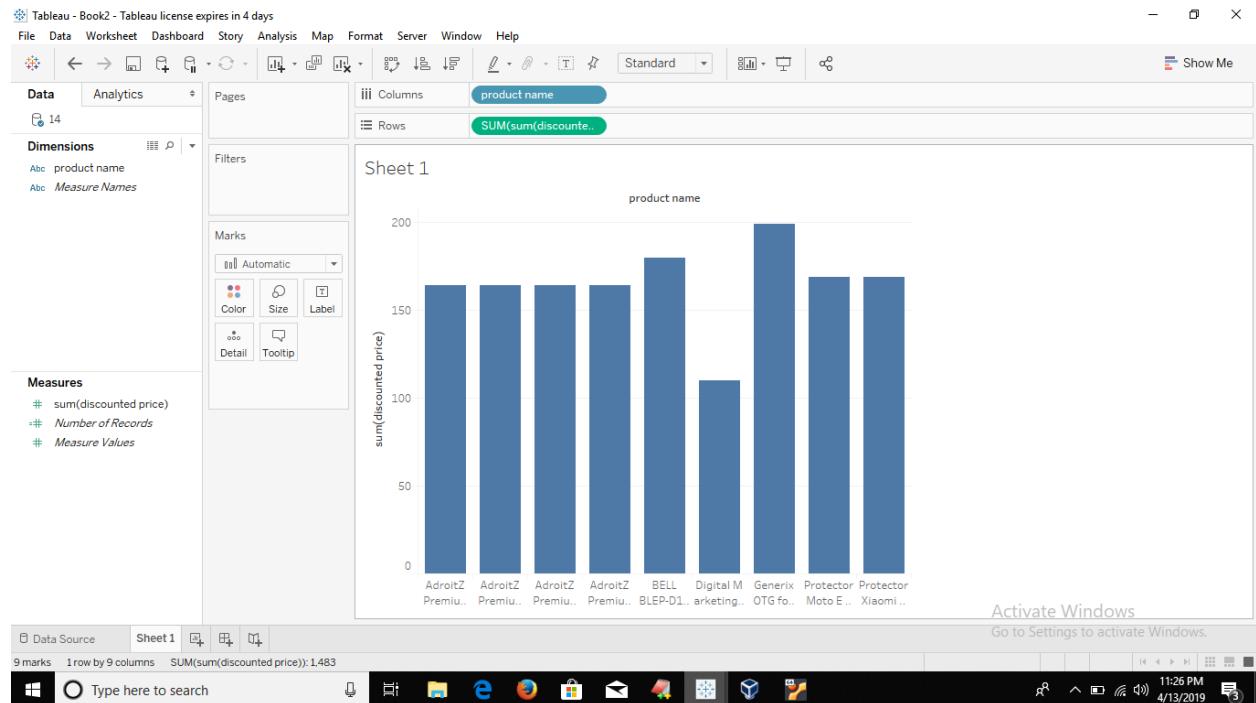


Fig. 25. Top ten products with highest discounted price

## TOP TEN PRODUCTS HAVING HIGHEST RECORDS

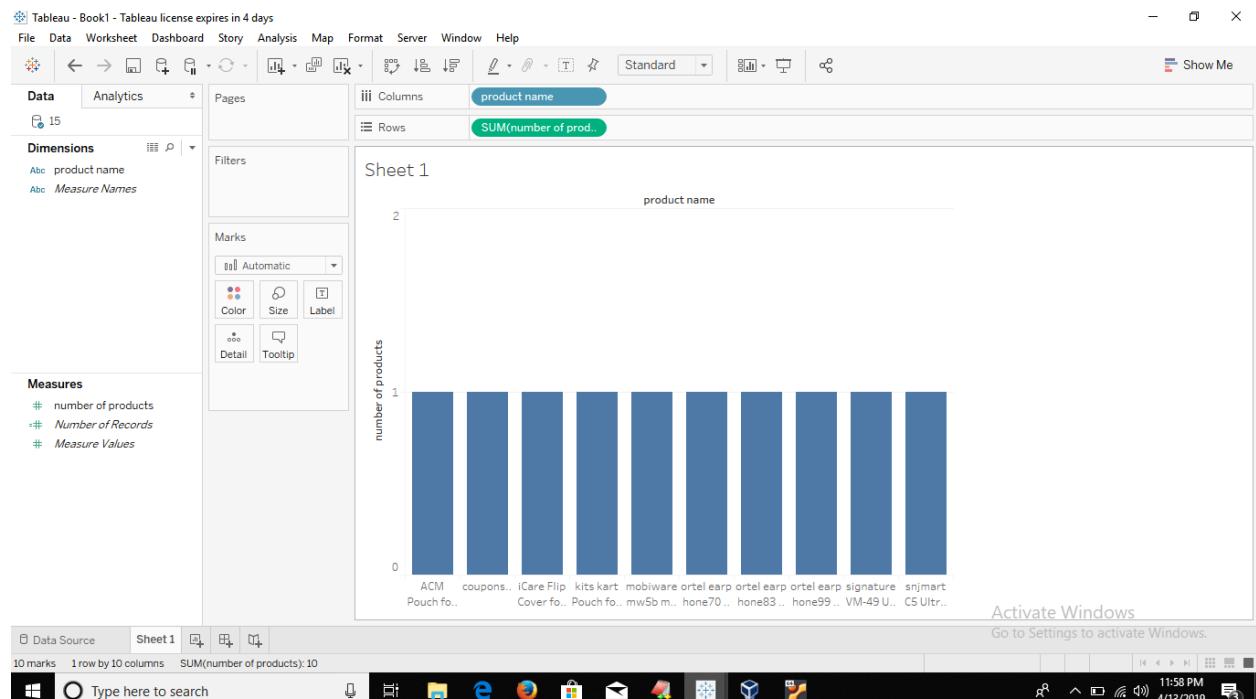


Fig. 26. Top ten products having highest records

## TOP TEN STATES WITH HIGHEST TOTAL DISCOUNTED PRICE

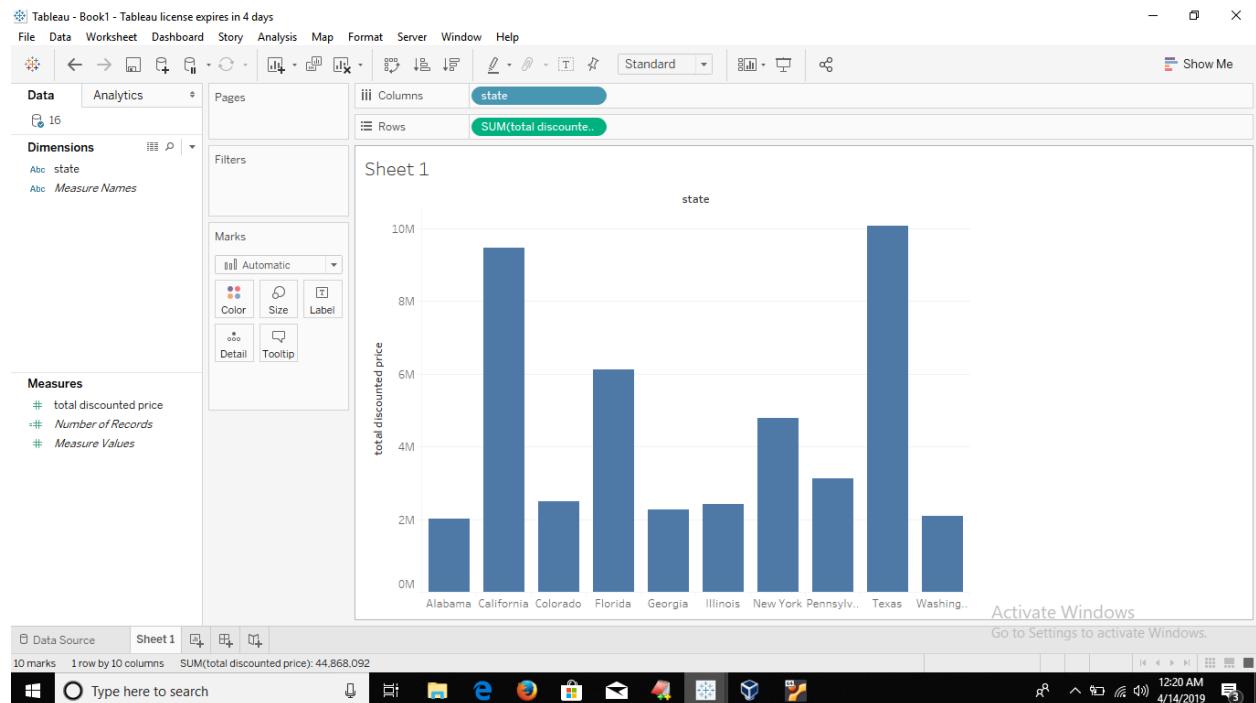


Fig. 27. Top ten states with highest total discounted price

## TOP TEN CITIES WITH HIGHEST TOTAL DISCOUNTED PRICE

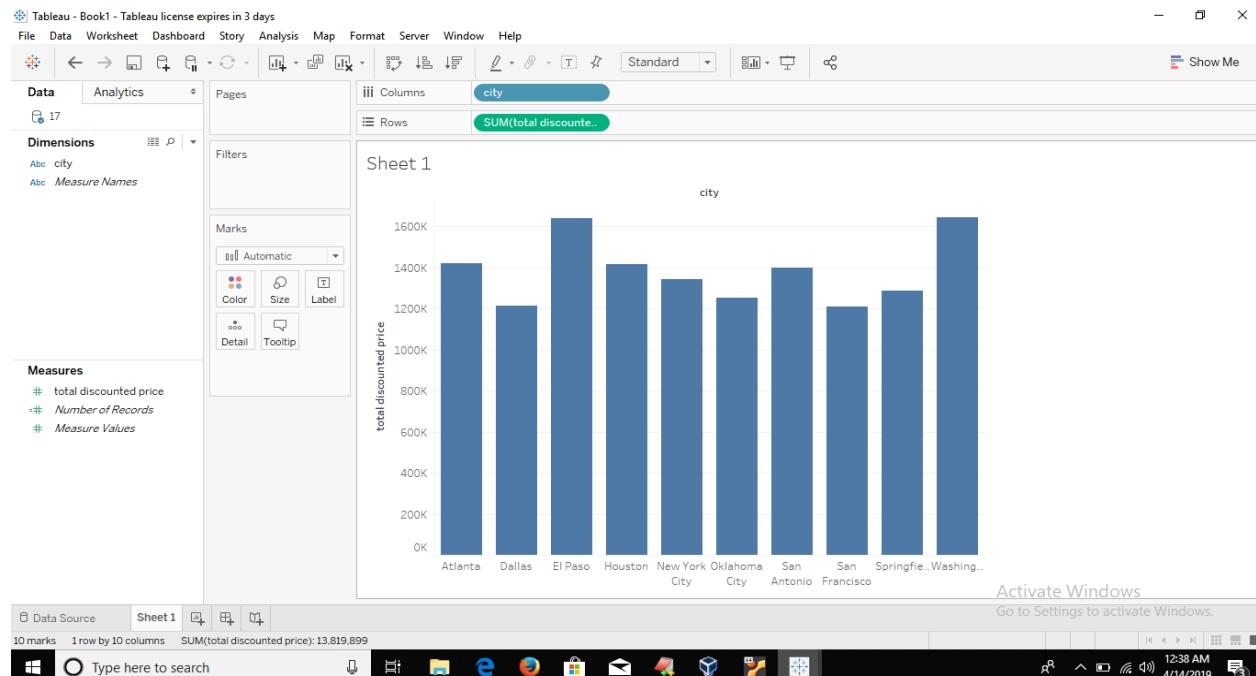


Fig. 28. Top ten cities with highest total discounted price

## USING MAPREDUCE PROGRAMS

### QUARTERWISE SALES

- DRIVER CLASS

```
StrTest.java  LowestStateRedu  LowestStateDriv  LowestStateMapp  TopCitiesDriver  TopCitiesMapper  EComDrive.java  %>
1 package com.retail;
2
3 import org.apache.hadoop.conf.Configuration;
4
5
6 public class EComDrive {
7     public static void main(String[] args) {
8
9         try {
10             BasicConfigurator.configure();
11             Configuration conf = new Configuration();
12             conf.set("fs.defaultFS", "hdfs://localhost:8020");
13
14             Job job = Job.getInstance(conf, "Retail Data analysis");
15
16             job.setMapperClass(EComMapper.class);
17             job.setReducerClass(EComReducer.class);
18             job.setOutputKeyClass(Text.class);
19             job.setOutputValueClass(IntWritable.class);
20
21             FileInputFormat.addInputPath(job, new Path("/user/hdpuuser/ecomm/input/orders.csv"));
22             FileOutputFormat.setOutputPath(job, new Path("/user/hdpuuser/ecomm/output/quarterwise_orders"));
23
24             job.waitForCompletion(true);
25             System.out.println("Status:::: "+job.getStatus().toString());
26             System.out.println(job.getStatus().toString());
27
28         } catch (Exception e) {
29             e.printStackTrace();
30         }
31     }
32 }
```

Type here to search      9:56 AM 4/26/2019

- MAPPER CLASS

```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities  Eclipse  Fri 09:59
workspace - Java EE - Ecomm/src/com/retail/EComMapper.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
StrTest.java  LowestStateRedu  LowestStateDriv  LowestStateMapp  TopCitiesDriver  TopCitiesMapper  EComDrive.java  EComMapper.java  %
1 package com.retail;
2
3 import java.io.*;
4
5 public class EComMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
6
7     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
8         String line = value.toString();
9         String fields[] = line.split(",");
10        //System.out.println("fields[2] :: "+fields[2]);
11        String monthStr = fields[2].substring(fields[2].indexOf("-")+1,fields[2].indexOf("-"+3));
12        //System.out.println("month :: "+monthStr);
13        int month = Integer.parseInt(monthStr);
14
15        String quarter = null;
16
17        if(month <=3){
18            quarter = "Q1";
19        }else if(month >6){
20            quarter = "Q2";
21        }else if(month >9){
22            quarter = "Q3";
23        }else{
24            quarter = "Q4";
25        }
26
27        context.write(new Text(quarter), new IntWritable(1));
28    }
29 }
```

Type here to search      9:56 AM 4/26/2019

- REDUCER CLASS

The screenshot shows the Eclipse IDE interface on an Ubuntu desktop. The title bar reads "Ubuntu [Running] - Oracle VM VirtualBox" and "workspace - Java EE - Ecomm/src/com/retail/EComReducer.java - Eclipse". The code editor displays the following Java code:

```
StrTest.java LowestStateRedu LowestStateDriv LowestStateMapp EComDrive.java EComMapper.java EComReducer.java
1 package com.retail;
2
3
4
5 import java.io.IOException;
6
7 public class EComReducer
8     extends Reducer<Text, IntWritable, Text, IntWritable> {
9     int total;
10
11     @Override
12     protected void reduce(Text key, Iterable<IntWritable> values,
13         Context context)
14         throws IOException, InterruptedException {
15         int sum = 0;
16         for(IntWritable val:values){
17             sum+=val.get();
18         }
19         context.write(key, new IntWritable(sum));
20     }
21 }
```

The code implements a reducer that sums up integer values for each key. The Java code is displayed in a light blue background, and the Eclipse interface includes toolbars, a sidebar with icons, and a status bar at the bottom.

## OUTPUT:

The screenshot shows a "Text Editor" window on the Ubuntu desktop. The title bar reads "Ubuntu [Running] - Oracle VM VirtualBox" and "part-r-00000[13] - Text Editor". The editor displays the following text:

```
Q1    778
Q2    705
Q3    793
Q4    732
```

The text represents the output of the reducer, showing four key-value pairs where the key is a question mark and the value is an integer. The text editor interface includes a toolbar, a status bar at the bottom, and a system tray icon.

## LOWEST STATES

- DRIVER CLASS

The screenshot shows the Eclipse IDE interface with the file `LowestStateDriver.java` open. The code implements a driver class for a Hadoop job. It sets up the configuration, specifies the mapper and reducer classes, and defines the input and output paths. The code also includes a try-catch block for handling exceptions.

```
12 public class LowestStateDriver {  
13     public static void main(String[] args) {  
14         try {  
15             BasicConfigurator.configure();  
16             Configuration conf = new Configuration();  
17             conf.set("fs.defaultFS", "hdfs://localhost:8020");  
18             Job job = Job.getInstance(conf, "Retail Data analysis");  
19             job.setMapperClass(LowestStateMapper.class);  
20             job.setReducerClass(LowestStateReducer.class);  
21             job.setOutputKeyClass(Text.class);  
22             job.setOutputValueClass(IntWritable.class);  
23             FileInputFormat.addInputPath(job, new Path("/user/hduser/ecommerce/input/Address.csv"));  
24             FileOutputFormat.setOutputPath(job, new Path("/user/hduser/ecommerce/output/lowest_customer_states"));  
25             job.waitForCompletion(true);  
26             System.out.println("Status:::: "+job.getStatus().toString());  
27             System.out.println(job.getStatus().toString());  
28         } catch (Exception e) {  
29             e.printStackTrace();  
30         }  
31     }  
32 }
```

- MAPPER CLASS

The screenshot shows the Eclipse IDE interface with the file `LowestStateMapper.java` open. The code defines a mapper class that extends `Mapper<LongWritable, Text, Text, IntWritable>`. It overrides the `map` method to process each line of input, split it by commas, and write the state and month values as key-value pairs.

```
1 package com.retail;  
2  
3 import java.io.*;  
4  
5 public class LowestStateMapper extends Mapper<LongWritable, Text, Text, IntWritable> {  
6     int counter = 0;  
7  
8     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {  
9         String line = value.toString();  
10        String fields[] = line.split(",");  
11        //System.out.println("fields[2] :: "+fields[2]);  
12        //String cityName = fields[2];  
13        String state = fields[3];  
14        //System.out.println("month :: "+monthStr);  
15  
16        context.write(new Text(state), new IntWritable(1));  
17    }  
18 }
```

- REDUCER CLASS

```

1 package com.retail;
2
3
4
5 import java.io.IOException;
6
7 public class LowestStateReducer
8     extends Reducer<Text, IntWritable, Text, IntWritable>{
9
10    int total;
11
12    Map<String, Integer> allcitiesMap = new HashMap<String, Integer>();
13    MyValueComparator bvc = new MyValueComparator(allcitiesMap);
14    TreeMap<String, Integer> sorted_map = new TreeMap<String, Integer>(bvc);
15
16
17    @Override
18    protected void reduce(Text key, Iterable<IntWritable> values,
19        Context context)
20        throws IOException, InterruptedException {
21
22        int sum = 0;
23        for(IntWritable val:values){
24            sum+=val.get();
25        }
26        allcitiesMap.put(key.toString(), sum);
27
28    }
29
30    @Override
31    protected void cleanup(Context context){
32
33        sorted_map.putAll(allcitiesMap);
34        System.out.println("allcitiesMap :: "+allcitiesMap);
35        Set<String> keyset = sorted_map.descendingKeySet();
36        System.out.println("sorted_map :: "+sorted_map);
37        System.out.println("keyset :: "+keyset);
38
39        int count =0;
40        try {
41            for(String cityName : keyset){
42                System.out.println("cityName :: "+cityName);
43                System.out.println("value :: "+sorted_map.get(cityName));
44                COUNT++;
45                context.write(new Text(cityName), new IntWritable(allcitiesMap.get(cityName)));
46                if(count==5){
47                    break;
48                }
49            }
50        } catch (IOException | InterruptedException e) {
51            // TODO Auto-generated catch block
52            e.printStackTrace();
53        }
54
55    }
56
57
58}
59
60
61
62
63

```

The code implements a Reducer class named LowestStateReducer. It extends the Reducer class from the org.apache.hadoop.mapreduce package. The class has a private variable total for storing the sum of values. It uses a HashMap named allcitiesMap to store the key-value pairs. A MyValueComparator named bvc is used to sort the map. A TreeMap named sorted\_map is created using the bvc comparator. The reduce method takes a Text key and an Iterable of IntWritable values, and calculates the sum of values for each key and stores it in the allcitiesMap. The cleanup method prints the allcitiesMap, gets its keyset, and then iterates over the keyset to write each key-value pair to the context. The code also includes a COUNT variable and a break statement after 5 iterations.

The screenshot shows the Eclipse IDE interface running on an Ubuntu desktop. The title bar indicates the workspace is 'Java EE - Ecomm/src/com/retail/LowestStateReducer.java - Eclipse'. The code editor displays Java code for a reducer, specifically a MyValueComparator class and its implementation of the Comparator interface. The code includes logic for reading from a file and comparing string values based on their integer representation.

```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities & Eclipse *
File Edit Source Refactor Navigate Search Project Run Window Help
StrTest.java LowestStateRedu LowestStateDrive LowestSt...
53     if(count==5){
54         break;
55     }
56 } catch (IOException | InterruptedException e) {
57     // TODO Auto-generated catch block
58     e.printStackTrace();
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66
class MyValueComparator implements Comparator<String> {
67
68     Map<String, Integer> base;
69     public MyValueComparator(Map<String, Integer> base) {
70         this.base = base;
71     }
72
73     public int compare(String a, String b) {
74         if (base.get(a) > base.get(b)) {
75             return -1;
76         } else {
77             return 1;
78         }
79     }
80 }
```

## OUTPUT

A screenshot of a Linux desktop environment, likely Ubuntu, running in Oracle VM VirtualBox. The desktop has a dark theme. On the left is a vertical dock with icons for various applications like a web browser, file manager, and system settings. A terminal window titled 'Text Editor' is open in the center, displaying the following text:

```
Vermont 1
Maine 1
North Dakota 1
Rhode Island 1
Alaska 2
```

The terminal window has a standard title bar with 'File', 'Machine', 'View', 'Input', 'Devices', and 'Help' options. The status bar at the bottom shows 'Plain Text' and 'Tab Width: 8'. The bottom of the screen features a dock with icons for system functions like volume, brightness, and network. The bottom right corner shows the date and time as '4/25/2019 4:54 PM'.

## MONTHLY ORDERS

- DRIVER CLASS

The screenshot shows the Eclipse IDE interface on a Windows desktop. The title bar reads "Ubuntu (Running) - Oracle VM VirtualBox" and "workspace - Java EE - Ecomm/src/com/retail/MonthlyOrdersDriver.java - Eclipse". The code editor displays the following Java code:

```
1 package com.retail;
2
3 import org.apache.hadoop.conf.Configuration;
4
5 public class MonthlyOrdersDriver {
6     public static void main(String[] args) {
7         try {
8             BasicConfigurator.configure();
9             Configuration conf = new Configuration();
10            conf.set("fs.defaultFS", "hdfs://localhost:8020");
11
12            Job job = Job.getInstance(conf, "monthwise_orders");
13
14            job.setMapperClass(MonthlyOrdersMapper.class);
15            job.setReducerClass(MonthlyOrdersReducer.class);
16            job.setOutputKeyClass(Text.class);
17            job.setOutputValueClass(IntWritable.class);
18
19            FileInputFormat.addInputPath(job, new Path("/user/hdpuuser/ecomm/input/orders.csv"));
20            FileOutputFormat.setOutputPath(job, new Path("/user/hdpuuser/ecomm/output/monthwise_orders"));
21
22            job.waitForCompletion(true);
23            System.out.println("Status:::: "+job.getStatus().getState().toString());
24            System.out.println(job.getStatus().toString());
25
26        } catch (Exception e) {
27            e.printStackTrace();
28        }
29    }
30}
```

The code implements a Hadoop job to process monthly orders. It sets up the configuration, defines the mapper and reducer classes, and specifies the input and output paths. The job is run with a completion check.

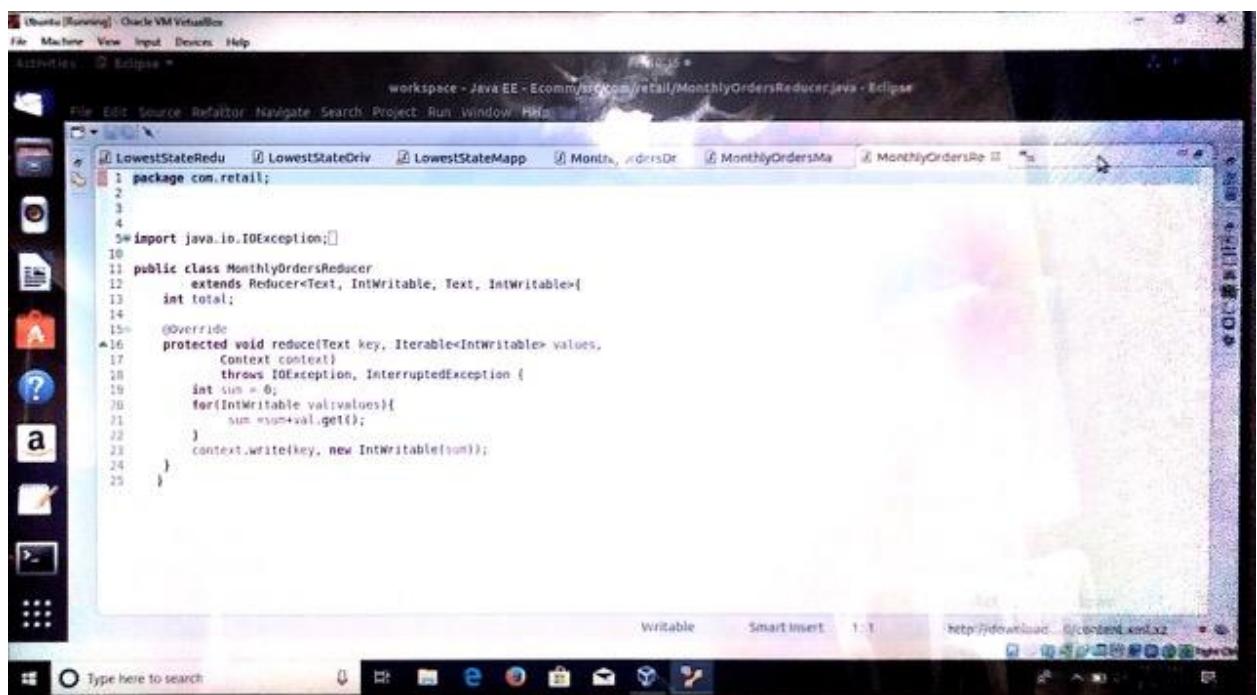
- MAPPER CLASS

The screenshot shows the Eclipse IDE interface on a Windows desktop. The title bar reads "Ubuntu (Running) - Oracle VM VirtualBox" and "workspace - Java EE - Ecomm/src/com/retail/MonthlyOrdersMapper.java - Eclipse". The code editor displays the following Java code:

```
1 package com.retail;
2
3 import java.io.*;
4
5 public class MonthlyOrdersMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
6
7     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
8         String line = value.toString();
9         String fields[] = line.split(",");
10        //System.out.println(fields[2] + ":" + fields[2]);
11        String monthStr = fields[2].substring(fields[2].indexOf("-") + 1, fields[2].indexOf("-") + 3);
12        //System.out.println("month :: " + monthStr);
13        int month = Integer.parseInt(monthStr);
14
15        String monthName = Month.of(month).name();
16        context.write(new Text(monthName), new IntWritable(1));
17
18    }
19
20}
```

The code defines a Mapper class that takes a LongWritable key and a Text value, and emits a Text key and an IntWritable value. It splits the input line by commas and extracts the month from the second field. The month is then converted to its name and emitted along with a count of 1.

- REDUCER CLASS



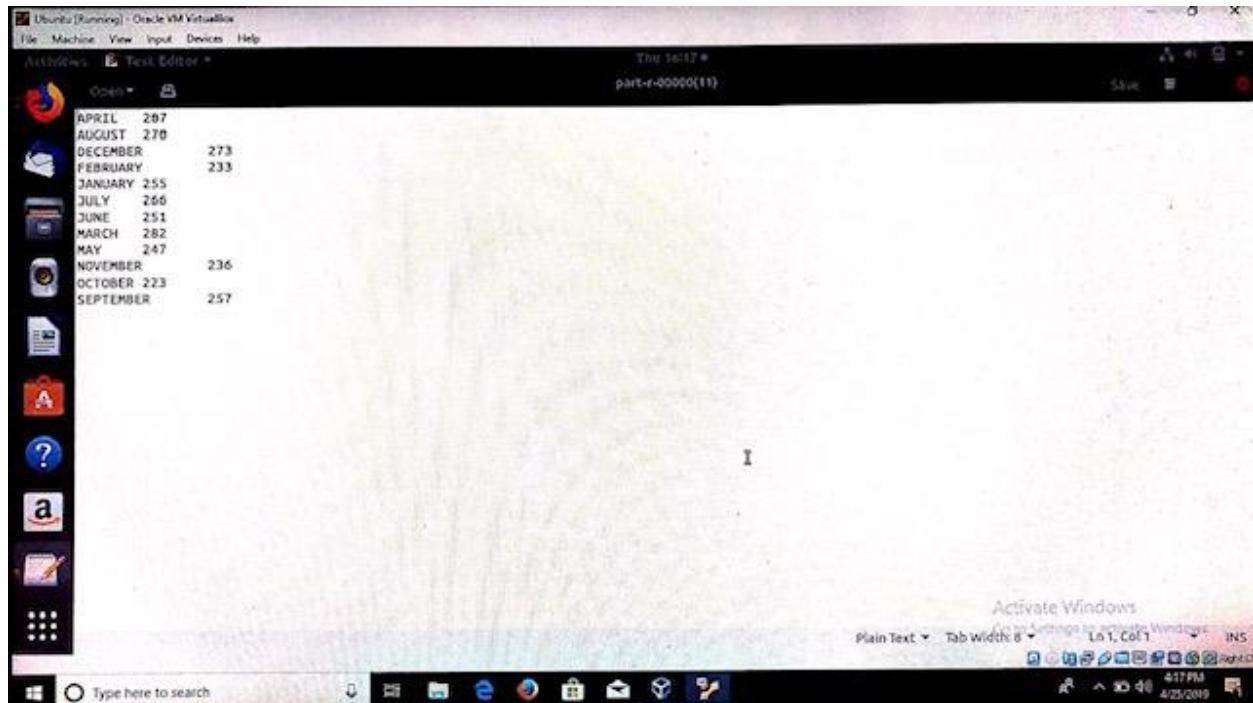
```

Ubuntu (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities  Eclipse *
File Edit Source Refactor Navigate Search Project Run Window Help
workspace - Java EE - Ecommerce/ecommerce/retail/MonthlyOrdersReducer.java - Eclipse
1 package com.retail;
2
3
4
5 import java.io.IOException;
6
7 public class MonthlyOrdersReducer
8     extends Reducer<Text, IntWritable, Text, IntWritable>{
9     int total;
10
11     @Override
12     protected void reduce(Text key, Iterable<IntWritable> values,
13             Context context)
14         throws IOException, InterruptedException {
15         int sum = 0;
16         for(IntWritable val:values){
17             sum+=val.get();
18         }
19         context.write(key, new IntWritable(sum));
20     }
21 }

```

The screenshot shows the Eclipse IDE interface with the code for `MonthlyOrdersReducer.java` in the editor. The code defines a reducer that takes a `Text` key and an iterable of `IntWritable` values, and outputs a `Text` key with a `IntWritable` value representing the sum of the input values.

## OUTPUT



```

Ubuntu (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities  Text Editor *
Type here to search
APRIL 287
AUGUST 270
DECEMBER 273
FEBRUARY 233
JANUARY 255
JULY 266
JUNE 251
MARCH 282
MAY 247
NOVEMBER 236
OCTOBER 223
SEPTEMBER 257

```

The screenshot shows a terminal window displaying the output of the reducer. The output consists of 12 lines, each containing a month name followed by a space and a numerical value. The months listed are APRIL, AUGUST, DECEMBER, FEBRUARY, JANUARY, JULY, JUNE, MARCH, MAY, NOVEMBER, OCTOBER, and SEPTEMBER, with their corresponding values being 287, 270, 273, 233, 255, 266, 251, 282, 247, 236, 223, and 257 respectively.

## QUARTER WISE AVERAGE SALES

- DRIVER CLASS

The screenshot shows the Eclipse IDE interface with the code editor open. The code is a Java driver class named `QwiseAvgOrdersDriver`. It imports `org.apache.hadoop.conf.Configuration` and defines a main method. The main method sets up a job configuration, specifies mapper and reducer classes, and defines input and output paths. It then waits for the job to complete and prints its status.

```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Eclipse *
workspace - Java EE - Ecomm/src/com/retail/QwiseAvgOrdersDriver.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
LowestStateRedu LowestStateMapp QwiseAvgOrdersD Mon' ersDr MonthlyOrdersMa MonthlyOrdersRe
1 package com.retail;
2
3@import org.apache.hadoop.conf.Configuration;
4
5 public class QwiseAvgOrdersDriver {
6     public static void main(String[] args) {
7         try {
8             BasicConfigurator.configure();
9             Configuration conf = new Configuration();
10            conf.set("fs.defaultFS", "hdfs://localhost:8020");
11
12            Job job = Job.getInstance(conf, "Retail Data analysis");
13
14            job.setMapperClass(QwiseAvgOrdersMapper.class);
15            job.setReducerClass(QwiseAvgOrdersReducer.class);
16            job.setOutputKeyClass(Text.class);
17            job.setOutputValueClass(IntWritable.class);
18
19            FileInputFormat.addInputPath(job, new Path("/user/hdpuuser/ecomm/input/orders.csv"));
20            FileOutputFormat.setOutputPath(job, new Path("/user/hdpuuser/ecomm/output/quarterwise_average_orders"));
21
22            job.waitForCompletion(true);
23            System.out.println("Status:::: "+job.getStatus().toString());
24            System.out.println(job.getJobID().toString());
25
26        } catch (Exception e) {
27            e.printStackTrace();
28        }
29    }
}

```

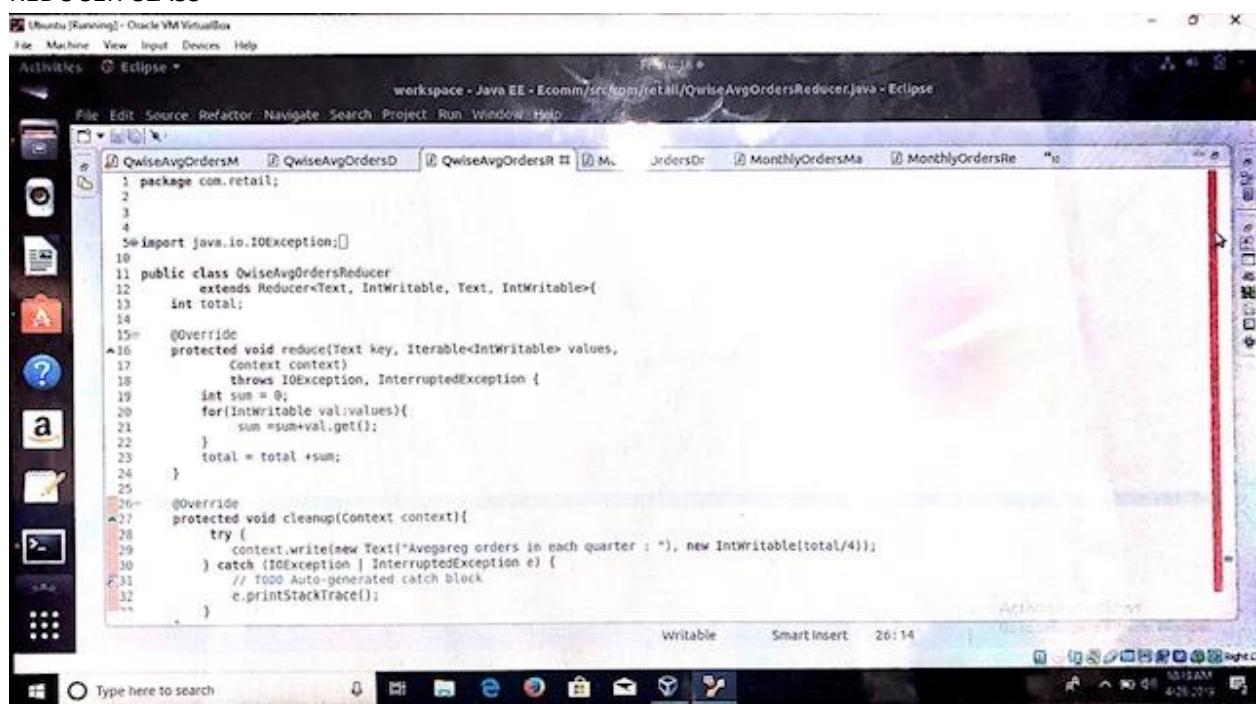
- MAPPER CLASS

The screenshot shows the Eclipse IDE interface with the code editor open. The code is a Java mapper class named `QwiseAvgOrdersMapper`. It extends `Mapper<LongWritable, Text, Text, IntWritable>`. The `map` method reads a line of text, splits it into fields, extracts the month, and calculates the quarter based on the month index. It then writes the quarter and a value of 1 to the context.

```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Eclipse *
workspace - Java EE - Ecomm/src/com/retail/QwiseAvgOrdersMapper.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
LowestStateRedu QwiseAvgOrdersM QwiseAvgOrdersD MonthlyOrdersDr MonthlyOrdersMa MonthlyOrdersRe
1 package com.retail;
2
3@import java.io.*;
4
5 public class QwiseAvgOrdersMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
6
7     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
8         String line = value.toString();
9         String[] fields = line.split(",");
10        //System.out.println(fields[2] + " " + fields[2]);
11        String monthStr = fields[2].substring(fields[2].indexOf("-") + 1, fields[2].indexOf("-") + 3);
12        //System.out.println("month :: " + monthStr);
13        int month = Integer.parseInt(monthStr);
14
15        String quarter = null;
16
17        if(month <=3){
18            quarter = "Q1";
19        }else if(month <=6){
20            quarter = "Q2";
21        }else if(month <=9){
22            quarter = "Q3";
23        }else{
24            quarter = "Q4";
25        }
26
27        context.write(new Text(quarter), new IntWritable(1));
28    }
}

```

- REDUCER CLASS



```

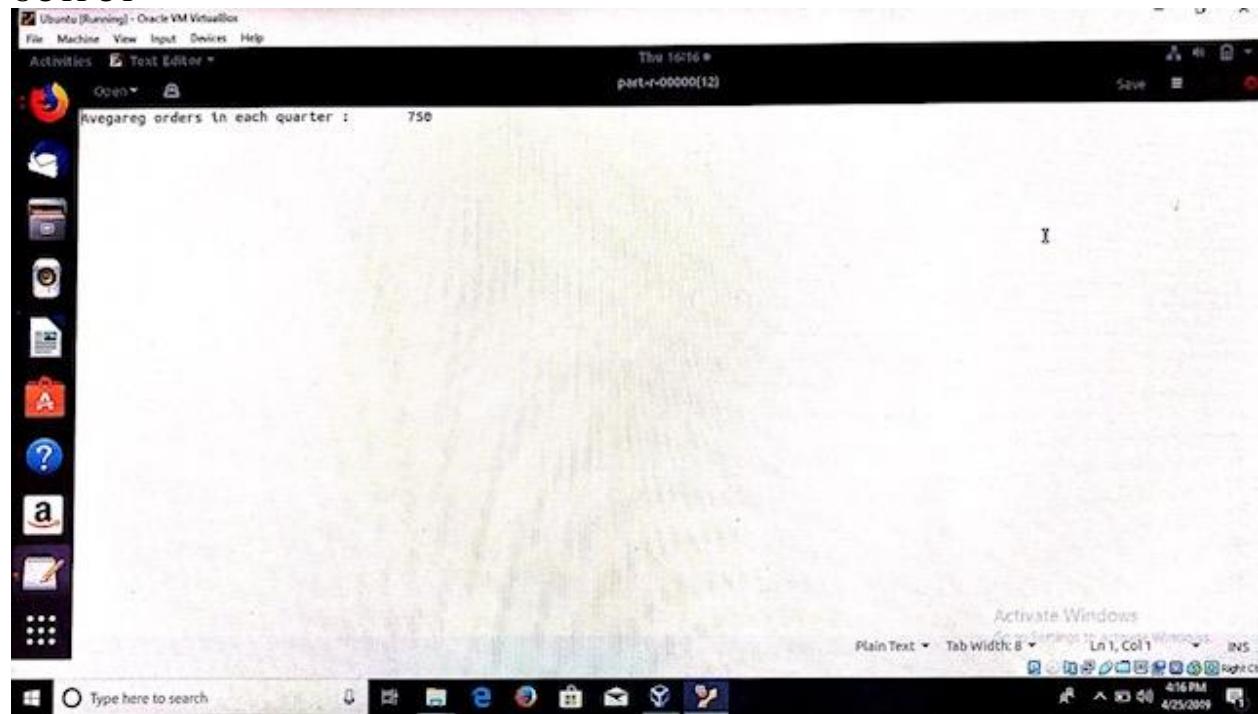
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Eclipse +
File Edit Source Refactor Navigate Search Project Run Window Help
workspace - Java EE - Ecomm/src/main/java/QwiceAvgOrdersReducer.java - Eclipse
QwiceAvgOrdersM QwiceAvgOrdersD QwiceAvgOrdersR # OrdersDr MonthlyOrdersMa MonthlyOrdersRe "is
1 package com.retail;
2
3
4
5 import java.io.IOException;
6
7 public class QwiceAvgOrdersReducer
8     extends Reducer<Text, IntWritable, Text, IntWritable>{
9     int total;
10
11     @Override
12     protected void reduce(Text key, Iterable<IntWritable> values,
13             Context context)
14         throws IOException, InterruptedException {
15         int sum = 0;
16         for(IntWritable val:values){
17             sum =sum+val.get();
18         }
19         total = total +sum;
20     }
21
22     @Override
23     protected void cleanup(Context context){
24         try {
25             context.write(new Text("Avegareg orders in each quarter : "), new IntWritable(total/4));
26         } catch (IOException | InterruptedException e) {
27             // TODO Auto-generated catch block
28             e.printStackTrace();
29         }
30     }
31
32 }

```

Writable SmartInsert 26:14

Type here to search

## OUTPUT



```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Text Editor +
Open Save
Avegareg orders in each quarter :    750
part-r-00000(12)
Plain Text Tab Width: 8 Ln 1, Col 1 INS
Activate Windows
Plain Text Tab Width: 8 Ln 1, Col 1 INS
4:16 PM 4/25/2019

```

Type here to search

## STATEWISE CUSTOMERS

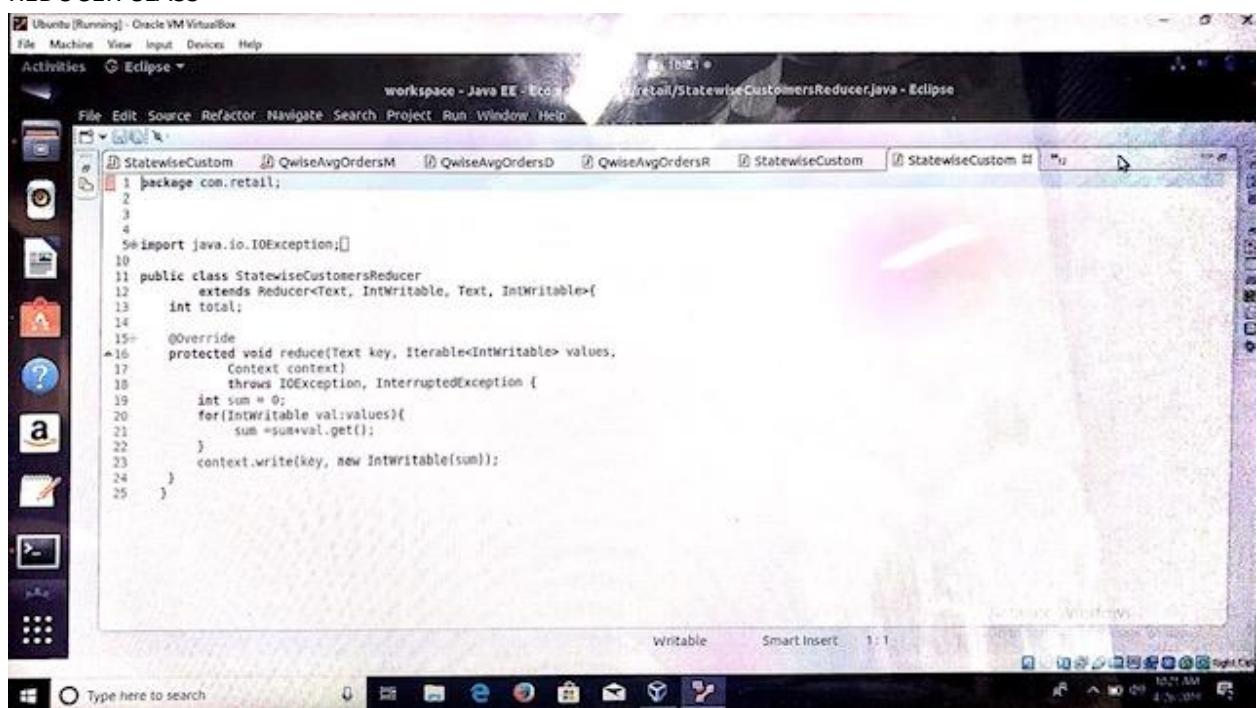
- DRIVER CLASS

```
Ubuntu [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities Eclipse workspace - Java EE - Ecomm/src/main/java/StatewiseCustomersDriver.java - Eclipse  
File Edit Source Refactor Navigate Search Project Run Window Help  
QwiseAvgOrdersM QwiseAvgOrdersD QwiseAvgOrdersR MonthlyOrdersMa MonthlyOrdersRe StatewiseCustom  
1 package com.retail;  
2  
3 import org.apache.hadoop.conf.Configuration;  
4  
5 public class StatewiseCustomersDriver {  
6     public static void main(String[] args) {  
7         try {  
8             BasicConfigurator.configure();  
9             Configuration conf = new Configuration();  
10            conf.set("fs.defaultFS", "hdfs://localhost:8020");  
11  
12            Job job = Job.getInstance(conf, "Statewise customer count");  
13  
14            job.setMapperClass(StatewiseCustomersMapper.class);  
15            job.setReducerClass(StatewiseCustomersReducer.class);  
16            job.setOutputKeyClass(Text.class);  
17            job.setOutputValueClass(IntWritable.class);  
18  
19            FileInputFormat.addInputPath(job, new Path("/user/hdpuuser/ecom/input/Address.csv"));  
20            FileOutputFormat.setOutputPath(job, new Path("/user/hdpuuser/ecom/output/statewise_customers"));  
21  
22            job.waitForCompletion(true);  
23            System.out.println("Status:::: "+job.getStatus().toString());  
24            System.out.println(job.getConfiguration().toString());  
25  
26        } catch (Exception e) {  
27            e.printStackTrace();  
28        }  
29    }  
30}
```

- MAPPER CLASS

```
Ubuntu [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities Eclipse workspace - Java EE - Ecomm/src/main/java/com/retail/statewiseCustomersMapper.java - Eclipse  
File Edit Source Refactor Navigate Search Project Run Window Help  
StatewiseCustom QwiseAvgOrdersM QwiseAvgOrdersD QwiseAvgOrdersR MonthlyOrdersRe StatewiseCustom  
1 package com.retail;  
2  
3 import java.io.*;  
4  
5 public class StatewiseCustomersMapper extends Mapper<LongWritable, Text, Text, IntWritable> {  
6     int counter;  
7  
8     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {  
9         if(counter == 0){  
10             counter++;  
11         }  
12         String line = value.toString();  
13         String fields[] = line.split(",");  
14         //System.out.println("fields[2] :: "+fields[2]);  
15         String state = fields[3];  
16         context.write(new Text(state), new IntWritable(1));  
17     }  
18 }  
19
```

- REDUCER CLASS



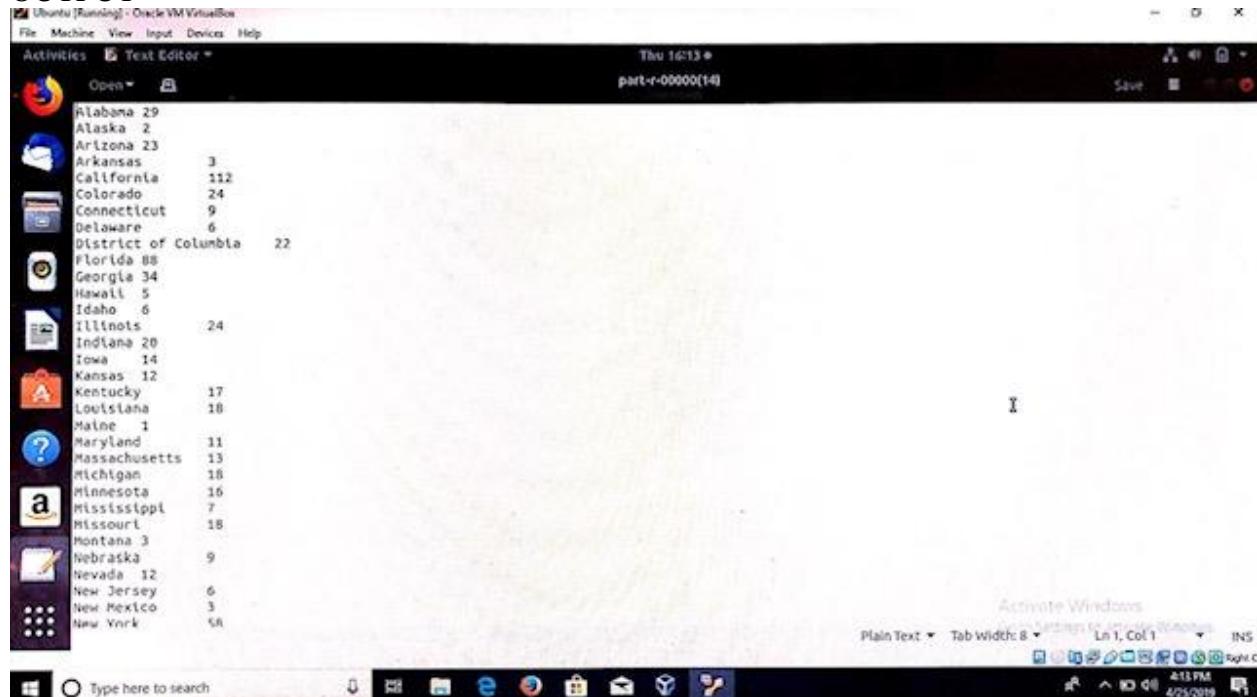
```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Eclipse
workspace - Java EE - Eclipse
retail/StatewiseCustomersReducer.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
1 package com.retail;
2
3
4
5 import java.io.IOException;
6
7 public class StatewiseCustomersReducer
8     extends Reducer<Text, IntWritable, Text, IntWritable>{
9     int total;
10
11     @Override
12     protected void reduce(Text key, Iterable<IntWritable> values,
13             Context context)
14         throws IOException, InterruptedException {
15         int sum = 0;
16         for(IntWritable val:values){
17             sum =sum+val.get();
18         }
19         context.write(key, new IntWritable(sum));
20     }
21 }

```

Writable Smart Insert 1:1

## OUTPUT



```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Text Editor
Open part-r-00000(14)
Thu 16:13
Alabama 29
Alaska 2
Arizona 23
Arkansas 3
California 112
Colorado 24
Connecticut 9
Delaware 6
District of Columbia 22
Florida 88
Georgia 34
Hawaii 5
Idaho 6
Illinois 24
Indiana 20
Iowa 14
Kansas 12
Kentucky 17
Louisiana 18
Maine 1
Maryland 11
Massachusetts 13
Michigan 18
Minnesota 16
Mississippi 7
Missouri 18
Montana 3
Nebraska 9
Nevada 12
New Jersey 6
New Mexico 3
New York 58

```

Activate Windows  
Plain Text Tab Width: 8 Ln 1, Col 1 INS

```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Text Editor
Open part-r-00000(14)
Thu 16:13 #
part-r-00000(14)
Save
Louisiana 18
Maine 1
Maryland 11
Massachusetts 13
Michigan 18
Minnesota 16
Mississippi 7
Missouri 18
Montana 3
Nebraska 9
Nevada 12
New Jersey 6
New Mexico 3
New York 58
North Carolina 18
North Dakota 1
Ohio 27
Oklahoma 14
Oregon 19
Pennsylvania 37
Rhode Island 1
South Carolina 11
South Dakota 2
Tennessee 21
Texas 129
Utah 9
Vermont 1
Virginia 29
Washington 27
West Virginia 9
Wisconsin 11
state 1

```

Activate Windows  
Plain Text Tab Width: 8 Ln 1, Col 1 INS

## TOP CITIES

- DRIVER CLASS

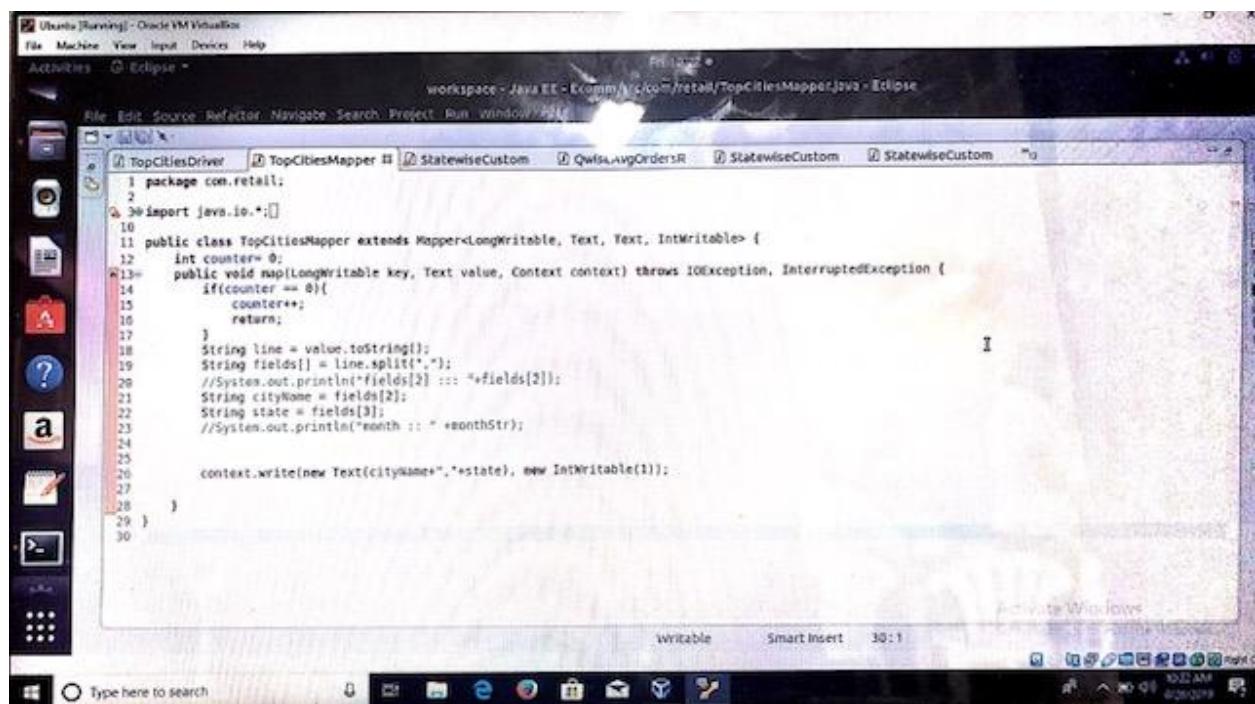
```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Eclipse
workspace - Java EE - Ecommerce - TopCitiesDriver.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
TopCitiesDriver.java
1 package com.retail;
2 import org.apache.hadoop.conf.Configuration;
3 public class TopCitiesDriver {
4     public static void main(String[] args) {
5         try {
6             BasicConfigurator.configure();
7             Configuration conf = new Configuration();
8             conf.set("fs.defaultFS", "hdfs://localhost:8020");
9
10            Job job = Job.getInstance(conf, "Retail Data analysis");
11
12            job.setMapperClass(TopCitiesMapper.class);
13            job.setReducerClass(TopCitiesReducer.class);
14            job.setOutputKeyClass(Text.class);
15            job.setOutputValueClass(IntWritable.class);
16
17            FileInputFormat.addInputPath(job, new Path("/user/hduser/ecom/input/Address.csv"));
18            FileOutputFormat.setOutputPath(job, new Path("/user/hduser/ecom/output/top5_cities"));
19
20            job.waitForCompletion(true);
21            System.out.println("Status:::: "+job.getStatus().toString());
22            System.out.println(job.getStatus().toString());
23
24        } catch (Exception e) {
25            e.printStackTrace();
26        }
27    }
}

```

writable Smart Insert 1:1

- MAPPER CLASS



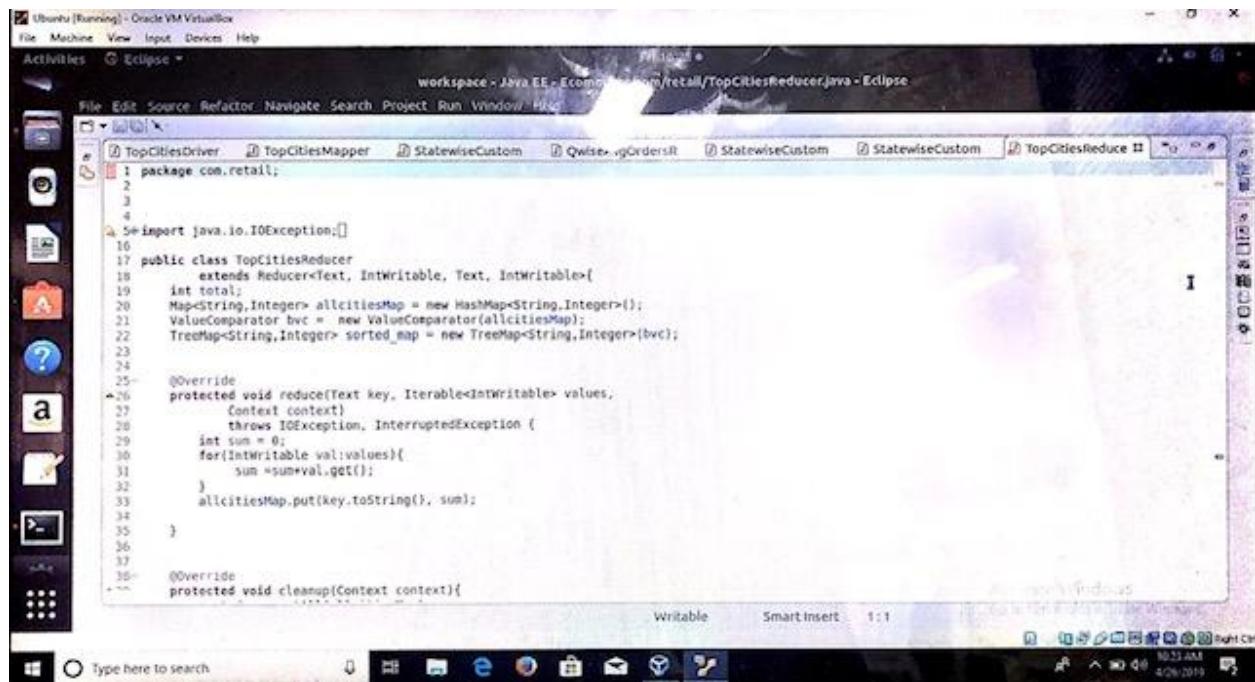
```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
workspace - Java EE - Ecommerce - com.retail/TopCitiesMapper.java - Eclipse
TopCitiesDriver TopCitiesMapper StatewiseCustom QwfsAvgOrdersR StatewiseCustom StatewiseCustom TopCitiesReduce
1 package com.retail;
2
3 import java.io.*;
4
5 public class TopCitiesMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
6     int counter = 0;
7     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
8         if(counter == 0){
9             counter++;
10            return;
11        }
12        String line = value.toString();
13        String fields[] = line.split(",");
14        //System.out.println("fields[2] :: " + fields[2]);
15        String cityName = fields[2];
16        String state = fields[3];
17        //System.out.println("month :: " + monthStr);
18
19        context.write(new Text(cityName), new IntWritable(1));
20    }
21}

```

The screenshot shows the Eclipse IDE interface with the code editor open to the `TopCitiesMapper.java` file. The code implements a `Mapper` class that processes input lines, extracts city and state names, and emits them as key-value pairs where the key is the city name and the value is the count of 1.

- REDUCER CLASS



```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
workspace - Java EE - Ecommerce - com.retail/TopCitiesReducer.java - Eclipse
TopCitiesDriver TopCitiesMapper StatewiseCustom QwfsAvgOrdersR StatewiseCustom StatewiseCustom TopCitiesReduce
1 package com.retail;
2
3
4
5 import java.io.IOException;
6
7 public class TopCitiesReducer extends Reducer<Text, IntWritable, Text, IntWritable>{
8     int total;
9     Map<String, Integer> allcitiesMap = new HashMap<String, Integer>();
10    ValueComparator bvc = new ValueComparator(allcitiesMap);
11    TreeMap<String, Integer> sorted_map = new TreeMap<String, Integer>(bvc);
12
13
14    @Override
15    protected void reduce(Text key, Iterable<IntWritable> values,
16                          Context context)
17        throws IOException, InterruptedException {
18        int sum = 0;
19        for(IntWritable val:values){
20            sum += val.get();
21        }
22        allcitiesMap.put(key.toString(), sum);
23    }
24
25    @Override
26    protected void cleanup(Context context){
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
311
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
139
```

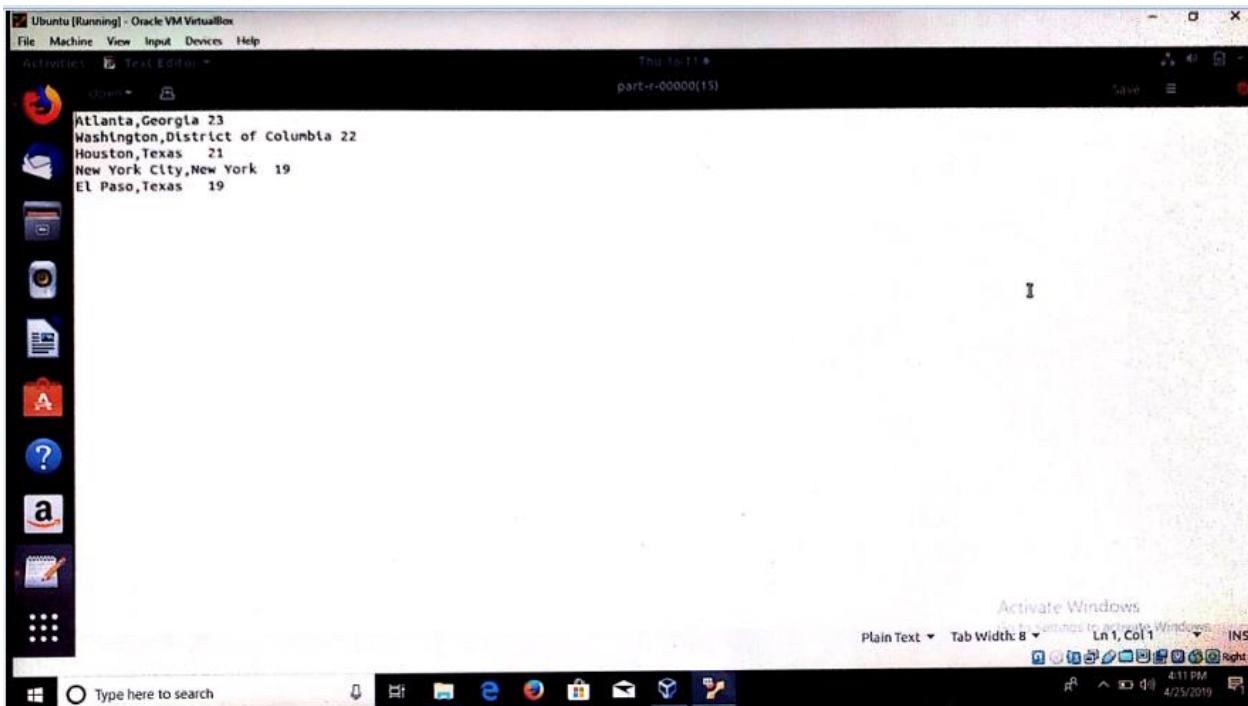
Ubuntu [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities Eclipse \* workspace - Java EE - com/retail/TopCitiesReducer.java - Eclipse

```
37+     @Override
38+     protected void cleanup(Context context){
39+         sortedMap.putAll(allcitiesMap);
40+         System.out.println("allcitiesMap :: "+allcitiesMap);
41+         Set<String> keyset = sortedMap.keySet();
42+         System.out.println("sorted map :: "+sortedMap);
43+         System.out.println("keyset :: "+keyset);
44+
45+         int count =0;
46+         try {
47+             for(String cityName : keyset){
48+                 System.out.println("cityName :: "+cityName);
49+                 System.out.println("value :: "+sortedMap.get(cityName));
50+                 count++;
51+                 context.write(new Text(cityName), new IntWritable(allcitiesMap.get(cityName)));
52+                 if(count==5){
53+                     break;
54+                 }
55+             }
56+         } catch (IOException | InterruptedException e) {
57+             // TODO Auto-generated catch block
58+             e.printStackTrace();
59+         }
60+     }
61+
62+
63+
64+
65+ }
```

Type here to search Ubuntu [Running] - Oracle VM VirtualBox Machine View Input Devices Help Activities Eclipse \* workspace - Java EE - Ecomm/src/com/retail/TopCitiesReducer.java - Eclipse

```
53         if(count==5){
54             break;
55         }
56     }
57 } catch (IOException | InterruptedException e) {
58     // TODO Auto-generated catch block
59     e.printStackTrace();
60 }
61
62
63
64
65
66 class ValueComparator implements Comparator<String> {
67
68     Map<String, Integer> base;
69     public ValueComparator(Map<String, Integer> base) {
70         this.base = base;
71     }
72
73     public int compare(String a, String b) {
74         if (base.get(a) >= base.get(b)) {
75             return -1;
76         } else {
77             return 1;
78         }
79     }
80 }
```

## OUTPUT



## 6. SYSTEM TESTING

The reason for testing is to find mistakes. Testing is the way toward endeavoring to find each possible issue or shortcoming in a work item. It gives an approach to check the usefulness of parts, sub-congregations, gatherings or potentially a completed item. It is the way toward practicing programming with the plan of guaranteeing that the Software framework lives up to its necessities and client desires and does not bomb in an unsatisfactory way. There are different kinds of test. Each test type tends to a particular testing prerequisite.

### 6.1. TYPES OF TESTING

#### 6.1.1. UNIT TESTING

Unit testing includes the plan of experiments that approve that the inward program rationale is working appropriately, and that program inputs produce legitimate yields. All choice branches and inner code stream ought to be approved. It is the trying of individual programming units of the application .it is done after the fruition of an individual unit before combination. This is a basic testing, that depends on information of its development and is obtrusive. Unit tests perform essential tests at segment level and test a particular business procedure, application, or potentially framework setup. Unit tests guarantee that every one of a kind way of a business procedure performs precisely to the archived details and contains unmistakably characterized sources of info and anticipated outcomes.

### **6.1.2. BLACK BOX TESTING**

Black Box Testing will be trying the product with no information of the inward activities, structure or language of the module being tried. Discovery tests, as most different sorts of tests, must be composed from an authoritative source record, for example, detail or necessities report, for example, particular or prerequisites archive. It is a trying in which the product under test is dealt with, as a discovery .you can't "see" into it. The test gives data sources and reacts to yields without thinking about how the product functions.

### **6.1.3. WHITE BOX TESTING**

White Box Testing is a trying in which in which the product analyzer knows about the inward operations, structure and language of the product, or if nothing else its motivation. It is reason. It is utilized to test territories that can't be come to from a discovery level.

### **6.1.4. TEST STRATEGY AND APPROACH**

Field testing will be performed manually and functional tests will be written in detail.

## **6.2. TEST OBJECTIVES**

- All field sections must work appropriately.
- Pages must be initiated from the recognized connection.
- The passage screen, messages and reactions must not be deferred.

## **6.3. FEATURES TO BE TESTED**

- Verify that the passages are of the right configuration
- No copy passages ought to be permitted
- All connections should take the client to the right page.

### **6.3.1. INTEGRATION TESTING**

Integration tests are intended to test coordinated programming segments to decide whether they really keep running as one program. Testing is occasion driven and is progressively worried about the fundamental result of screens or fields. Joining tests exhibit that despite the fact that the parts were independently fulfillment, as appeared by effectively unit testing, the mix of segments is right and steady. Coordination testing is explicitly gone for uncovering the issues that emerge from the mix of parts.

### **6.3.2. FUNCTIONAL TESTING**

Practical tests give deliberate showings that capacities tried are accessible as determined by the business and specialized prerequisites, framework documentation, and client manuals.

Practical testing is focused on the accompanying things:

Legitimate Input : recognized classes of substantial information must be acknowledged.

Invalid Input : distinguished classes of invalid info must be rejected.

Capacities : distinguished capacities must be worked out.

Yield : distinguished classes of use yields must be worked out.

Frameworks/Procedures : interfacing frameworks or techniques must be summoned.

Association and arrangement of practical tests is centered around prerequisites, key capacities, or unique experiments. What's more, orderly inclusion relating to recognize Business process streams; information fields, predefined forms, and progressive procedures must be considered for testing. Before practical testing is finished, extra tests are recognized and the successful estimation of current tests is resolved.

### **6.3.3. SYSTEM TESTING**

System testing guarantees that the whole incorporated programming framework meets prerequisites. It tests a design to guarantee known and unsurprising outcomes. A case of framework testing is the design situated framework incorporation test. Framework testing depends on procedure portrayals and streams, accentuating pre-driven procedure connections and incorporation focuses.

### **5.4. TESTS CONDUCTED**

Tests are led to watch that each hub in the bunch is effectively working or not and the physically some test are led to check the yield esteems.

#### **Test Results**

All the test cases mentioned above passed successfully. No defects encountered.

## **7. CONCLUSION**

I found the business bits of knowledge of current online business information. Also, get the advantages for business development. State shrewd check of client from this, we discovered which are the states have most noteworthy number of clients and which have least number of customers.so that, if the clients tally is low, they can do, more advancements in such territories and they can develop their business. City insightful number of requests, from this, we discovered which urban communities have most astounding number of requests and least number of requests. Quarter shrewd deals and the sky is the limit from there.

## 8. BIBLIOGRAPHY

- [1] Big Data and Cloud Computing: Current State and Future Opportunities 2013. [2] D. Agrawal, S. Das, and A. E. Abbadi. Big data and cloud computing: New wine or just new bottles? PVLDB, 3(2):1647–1648, 2010. [3] D. Agrawal, A. El Abbadi, S. Antony, and S. Das. Data Management Challenges in Cloud Computing Infrastructures. In DNIS, pages1–10, 2010. [4] P. Agrawal, A. Silberstein, B. F. Cooper, U. Srivastava, and. Ramakrishnan. Asynchronous view maintenance for vlsddatabases. In SIGMOD Conference, pages 179–192, 2009. [5] S. Aulbach, D. Jacobs, A. Kemper, and M. Seibold. A comparison offlexible schemas for software as a service. In SIGMOD, pages881–888, 2009. [6] “Understanding Hadoop Clusters and the Network.” Available at <http://bradhedlund.com>. Accessed on June 1, 2013. [7] Sammer, E. 2012. Hadoop Operations. Sebastopol, CA: O'Reilly Media. [8] “HDFS High Availability Using the Quorum Journal Manager.” Apache Software Foundation. Available at <http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/HDFSHighAvailabilityWithQJM.html>. Accessed on June 5, 2013. [9] “Hadoop HDFS over HTTP - Documentation Sets 2.0.4-alpha.” Apache Software Foundation. Available at <http://hadoop.apache.org/docs/r2.0.4-alpha/hadoop-hdfs-httpfs/index.html>. Accessed on June 5, 2013. [10]“Yahoo! Hadoop Tutorial.” Yahoo! Developer Network. Available at <http://developer.yahoo.com/hadoop/tutorial/>. Accessed on June 4, 2013. [11]“Configuring the Hive Metastore.” Cloudera, Inc. Available at [http://www.cloudera.com/content/cloudera-content/cloudera-docs/CDH4/4.2.0/CDH4-Installation-Guide/cdh4ig\\_topic\\_18\\_4.html](http://www.cloudera.com/content/cloudera-content/cloudera-docs/CDH4/4.2.0/CDH4-Installation-Guide/cdh4ig_topic_18_4.html). Accessed on June 18, 2013. [12] Kestelyn, J. “Introducing Parquet: Efficient Columnar Storage for Apache Hadoop.” Available at <http://blog.cloudera.com/blog/2013/03/introducing-parquet-columnar-storage-for-apache-hadoop/>. Accessed on August 2, 2013.