**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

It is broadly acknowledged that wellbeing data devices and methods can be misused effectively to help specialists in diagnosing and treating their patients all the more productively. Utilizing their experience and information, the doctors order patients and analyze their infections, however in doing as such, it is plausible that they submit a few errors, especially when they need sufficient experience or when their staff of judgment is poor. In such circumstances, Clinical Decision Support (CDS) frameworks, including frameworks that give finding, customized restorative estimation, treatment and important learning, would be useful to the doctors by method for giving them particular learning, patients' data and keen applications, which can enhance the proficiency of their basic leadership form CDS frameworks center on separating attributes of patients, in light of which they order patients and give relating clinical recommendations to the doctors.

Traditional methods usually treat all terms with same feature sets, such that performance can be damaged when noisy information is brought by wrong features for a given term. Here, we propose a term-based personalization approach to finding the best features for each term. First, features are given as the input so that we focus on selection strategies. Second, we present a feature searching method to generate feature candidate subsets for each term. The Big data deals with the analysis of the patient details, problems, reports and hospitals specialties. Finally, we obtain the personalized feature set for each term as a subset of all features.

**1.2 BIG DATA**

The data which is beyond to the storage capacity and beyond to the processing power such a data is called Big Data. Big data means really a big data; it is a collection of large data sets that cannot be processed using traditional computing techniques. Big data is not merely a data; rather it has become a complete subject, which involves various tools, techniques and frameworks. Data which are very large in size is called Big Data. Normally we work on data of size MB (Wordbook, Excel) or maximum GB (Movies, Codes) but data in Petabytes i.e. 10^15-byte size is called Big Data. It is stated that almost 90% of today's data has been generated in the past 6 years. However, there are certain basic tenets of Big Data that will make it even simpler to answer what is Big Data:

1. It refers to a massive amount of data that keeps on growing exponentially with time.
2. It is so voluminous that it cannot be processed or analyzed using conventional data processing techniques.
3. It includes data mining, data storage, data analysis, data sharing, and data visualization.
4. The term is an all-comprehensive one including data, data frameworks, along with the tools and techniques used to process and analyze the data

The same concept applies on Big Data. Big Data says, till today, we were okay with storing the data into our servers because the volume of the data was pretty limited, and the amount of time to process this data was also okay.  But now in this current technological world, the data is growing too fast and people are relying on the data a lot of times. Also the speed at which the data is growing, it is becoming impossible to store the data into any server.

* 1. **TYPES OF BIG DATA**

1. **STRUCTURED**

By structured data, we mean data that can be processed, stored, and retrieved in a fixed format. It refers to highly organized information that can be readily and seamlessly stored and accessed from a database by simple search engine algorithms. For instance, the employee table in a company database will be structured as the employee details, their job positions, their salaries, etc., will be present in an organized manner.

1. **UNSTRUCTURED**

Unstructured data refers to the data that lacks any specific form or structure whatsoever. This makes it very difficult and time-consuming to process and analyze unstructured data. Email is an example of unstructured data.

1. **SEMI-STRUCTURED**

Semi-structured data pertains to the data containing both the formats mentioned above, that is, structured and unstructured data. To be precise, it refers to the data that although has not been classified under a particular repository (database), yet contains vital information or tags that segregate individual elements within the data.

**1.4 CHALLENGES WITH BIG DATA**

1. **DATA QUALITY**

The problem here is the 4th V i.e. Veracity. The data here is very messy, inconsistent and incomplete. Dirty data cost $600 billion to the companies every year in the United States.

1. **DISCOVERY**

Finding insights on Big Data is like finding a needle in a haystack. Analyzing petabytes of data using extremely powerful algorithms to find patterns and insights are very difficult.

1. **STORAGE**

The more data an organization has, the more complex the problems of managing it can become. The question that arises here is “Where to store it?”. We need a storage system which can easily scale up or down on-demand.

1. **Analytics**

In the case of Big Data, most of the time we are unaware of the kind of data we are dealing with, so analyzing that data is even more difficult.

1. **Security**

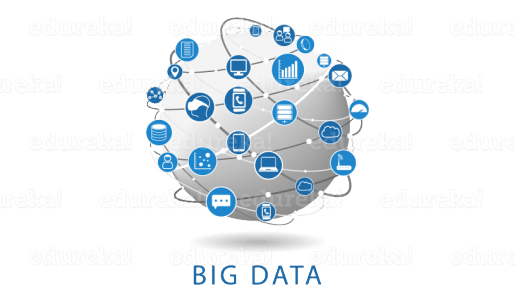
Since the data is huge in size, keeping it secure is another challenge. It includes user authentication, restricting access based on a user, recording data access histories, proper use of data encryption etc.

1. **Lack of Talent**

There are a lot of Big Data projects in major organizations, but a sophisticated team of developers, data scientists and analysts who also have sufficient amount of domain knowledge is still a challenge.

## 1.5 APPLICATIONS OF BIG DATA

Cannot talk about data without talking about the people, people who are getting benefited by Big Data applications. Almost all the industries today are leveraging Big Data applications in one or the other way.



1. **SMARTER HEALTHCARE**

Making use of the petabytes of patient’s data, the organization can extract meaningful information and then build applications that can predict the patient’s deteriorating condition in advance.

1. **TELECOM**

Telecom sectors collects information, analyzes it and provide solutions to different problems. By using Big Data applications, telecom companies have been able to significantly reduce data packet loss, which occurs when networks are overloaded, and thus, providing a seamless connection to their customers.

1. **RETAIL**

Retail has some of the tightest margins, and is one of the greatest beneficiaries of big data. The beauty of using big data in retail is to understand consumer behavior. Amazon’s recommendation engine provides suggestion based on the browsing history of the consumer.

1. **TRAFFIC CONTROL**

Traffic congestion is a major challenge for many cities globally. Effective use of data and sensors will be key to managing traffic better as cities become increasingly densely populated.

1. **MANUFACTURING**

Analyzing big data in the manufacturing industry can reduce component defects, improve product quality, increase efficiency, and save time and money.

1. **SEARCH QUALITY**

Every time we are extracting information from google, we are simultaneously generating data for it. Google stores this data and uses it to improve its search quality.

* 1. **FEATURE SELECTION**

Feature selection has been an important research area in data mining, which chooses a subset of relevant features for use in the model building. This paper aims to provide an overview of feature selection methods for big data mining. First, it discusses the current challenges and difficulties faced when mining valuable information from big data. A comprehensive review of existing feature selection methods in big data is then presented. Herein, we approach the review from two aspects: methods specific to a particular kind of big data with certain characteristics and applications of methods in classification analysis, which are significantly different to the existing review work. This paper also highlights the current issues of feature selection in big data and suggests the future research directions.

**1.8 CHALLENGES OF FEATURE SELECTION**

Compared to traditional data, some influential points need to be highlighted on extracting valuable information from big data. Taking the 3V characteristics into consideration, traditional feature selection methods face the following threefold challenges with respect to the case of big data. Traditional feature selection methods usually require large amounts of learning time, so it is hard for processing speed to catch up with the change of big data. Generally, big data not only include an immense amount of irrelevant and/or redundant features, but also have possible noises of different degrees and different types, which greatly increases the difficulty of selecting feature. Some data are unreliable/forged, due to different means of acquisition, or even loss, which further enhances the complexity of feature selection.

**1.9 OBJECTIVE OF FEATURE SELECTION**

1. Feature learning and selection have been widely applied in many research areas because of their good performance and lower complexity.
2. To propose a term-based personalization approach to finding the best features for each term.
3. First, features are given as the input so that we focus on selection strategies.
4. Second, we present a feature searching method to generate feature candidate subsets for each term.
5. Here, the big data only deals with the analysis of the patient details, problems, reports and hospitals specialties.
6. Finally, we obtain the personalized feature set for each term as a subset of all features so they can used to integrate onto searching method.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 TITLE**

A Machine Learning Approach to Classifying Self-Reported Health Status in a cohort of Patients with Heart Disease using Activity Tracker Data.

**Author:**Yiwen Meng, William Speier, Member, Chrisandra Shufelt, Sandy Joung, Jennifer E Van Eyk, C. Noel Bairey Merz, Mayra Lopez, Brennan Spiegel and Corey W. Arnold

**Description:**Constructing statistical models using personal sensor data could allow for tracking health status over time, thereby enabling the possibility of early intervention. The goal of this study was to use machine learning algorithms to classify patient-reported outcomes (PROs) using activity tracker data in a cohort of patients with stable ischemic heart disease (SIHD). A population of 182 patients with SIHD was monitored over a period of 12 weeks. Each subject received a Fit bit Charge 2 device to record daily activity data, and each subject completed eight Patient-Reported Outcomes Measurement Information Systems (PROMIS®) short form at the end of each week as a self-assessment of their health status. Two models were built to classify PRO scores using activity tracker data. The first model treated each week independently, while the second used a Hidden Markov model (HMM) to take advantage of correlations between successive weeks. Retrospective analysis compared the classification accuracy of the two models and the importance of each feature. In the independent model, a random forest classifier achieved a mean area under curve (AUC) of 0.76 for classifying the Physical Function PRO.

**2.2 TITLE**

Dynamic Prediction in Clinical Survival Analysis using Temporal Convolutional Networks

**Author:** Daniel Jarrett, Jinsung Yoon, Member, IEEE, and Mihaela van der Schaar, Fellow, IEEE

**Description:** Accurate prediction of disease trajectories is critical for early identification and timely treatment of patients at risk. Conventional methods in survival analysis are often constrained by strong parametric assumptions and limited in their ability to learn from high-dimensional data. This paper develops a novel convolutional approach that addresses the drawbacks of both traditional statistical approaches as well as recent neural network models for survival. We present MATCH-Net: A Missing Awareness Temporal Convolutional Hitting-time Network, designed to capture temporal dependencies and heterogeneous interactions in covariate trajectories and patterns of missingness. To the best of our knowledge, this is the first investigation of temporal convolutions in the context of dynamic prediction for personalized risk prognosis

**2.3 TITLE**

Guest-Editorial Biomedical Informatics in Clinical Environments

**Author:** M. AKAY and D. I. FOTIADIS

**Description:** The point of this uncommon segment is to give an outline of the rising biomedical informatics advancements and their application in inquire about and clinical conditions. Later advancements in biomedical informatics have made strategies, strategies and apparatuses, which depend on the investigation of heterogeneous information, information mining, choice emotionally supportive networks, multiscale demonstrating, and so on. The separation from the improvement of such frameworks what's more, the genuine clinical situations is still sufficiently long, and just some of them have been utilized in a clinical scale. Physiological signals, such as Electrocardiograms and Electroencephalograms, are used for the diagnosis of the patient status and detection of events. In, an unsupervised, robust, and computationally fast algorithm that uses Modified Multiscale Sample Entropy and Kurtosis to automatically identify the independent eye blink components is presented.

**2.4 TITLE**

Learning multi-label scene classification

**Author:** Matthew R. Boutella, Jiebo Luob

**Description:** In great example acknowledgment issues, classes are fundamentally unrelated by dentition.Classication mistakes happen when the classes cover in the component space. We look at a di5erent circumstance, happening when the classes are, by dentition, not totally unrelated. Such issues emerge in semantic scene and archive classication and in restorative analysis. We present a system to deal with such issues and apply it to the issue of semantic scene classication where a characteristic scene may contain various protests to such an extent that the scene can be depicted by different class names (e.g., an eld scene with a mountain in the foundation). Such an issue presents difficulties to the exemplary example acknowledgment worldview and requests a di5erent treatment. We talk about methodologies for preparing and testing in this situation and present new measurements for assessing singular precedents, class review and exactness, and by and large precision. Investigations demonstrate that our strategies are appropriate for scene classication; besides, our work seems to sum up to other classication issues of a similar sort.

**2.5 TITLE**

The Progression of Hypertensive Heart Disease

**Author:** Mark H. Drazner

**Description:** Hypertension remains a noteworthy general medical issue related with significant grimness and mortality. Hypertensive coronary illness is a group of stars of variations from the norm that incorporates left ventricular hypertrophy (LVH), systolic and diastolic brokenness, and their clinical appearances including arrhythmias and symptomatic heart disappointment. The work of art worldview of hypertensive coronary illness is that the left ventricular (LV) divider thickens in light of hoisted blood weight as a compensatory instrument to limit divider push. In this way, after a progression of ineffectively portrayed occasions ("change to disappointment"), the left ventricle expands, and the LV discharge portion (EF) decreases (characterized thus as "widened heart failure").1 The reason for this survey is to centre around the key strides in the movement of hypertensive coronary illness, featuring ongoing advances too as some uncertain discussions.

**CHAPTER 3**

**AIM AND SCOPE OF THE PROJECT**

**3.1 AIM**

To propose a term-based personalization approach to finding the best features for each term. The big data only deals with the analysis of the patient details, problems, reports and hospitals specialties. Finally, we obtain the personalized feature set for each term as a subset of all features so they can used to integrate onto searching method. Analyzing clinical data by using Hadoop tool along with some Hadoop ecosystems like hdfs, MapReduce, sqoop, hive and pig. By using these tools we can process no limitation of data, no data lost problem, we can get high throughput, maintenance cost also very less and it is an open source software, it is compatible on all the platforms since it is Java based. In clinical data details, this is based on whenever patient enters their id in hospital database within no time complex their data will present.

**3.2 SCOPE OF THE PROJECT**

Analyzing Clinical Data by using Hadoop tool along with some Hadoop ecosystems like HDFS, map reduce, Sqoop, hive and pig. By using these tools, we can process no limitation of data, no data lost problem, we can get high throughput, maintenance cost also very less and it is an open source software, it is compatible on all the platforms since it is Java based. In clinical, we will know which hospital is curing better for particular health problems.

**3.3 EXISTING SYSTEM**

Existing concept deals with providing backend by using MySQL which contains lot of drawbacks i.e. data limitation is that processing time is high when the data is huge and once data is lost, we cannot recover so thus we proposing concept by using Hadoop tool.

MySQL is a fast, easy to use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

1. MySQL is released under an open-source license. So you have nothing to pay to use it.
2. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
3. MySQL uses a standard form of the well-known SQL data language.
4. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
5. MySQL works very quickly and works well even with large data sets.
6. MySQL is very friendly to PHP, the most appreciated language for web development.
7. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
8. MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.
   * 1. **DRAWBACKS OF EXISTING SYSTEM**
9. We can process limitation of data.
10. We get results with take more time and maintenance cost is very high**.**

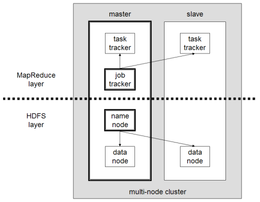
**3.4 PROPOSED SYSTEM**

Proposed concept deals with providing database by using Hadoop tool we can analyze no limitation of data and simply add number of machines to the cluster and we get results with less time, high throughput and maintenance cost is very less and we are using joins, partitions and bucketing techniques in Hadoop.

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Hadoop consists of the Hadoop Common package, which provides file system and operating system level abstractions, a MapReduce either MapReduce and the [Hadoop Distributed File System](https://en.wikipedia.org/wiki/Apache_Hadoop#Hadoop_distributed_file_system) (HDFS). The Hadoop Common package contains  files and scripts needed to start Hadoop.

For effective scheduling of work, every Hadoop-compatible file system should provide location awareness, which is the name of the rack, specifically the network switch where a worker node is. Hadoop applications can use this information to execute code on the node where the data is, and, failing that, on the same rack/switch to reduce backbone traffic. HDFS uses this method when replicating data for data redundancy across multiple racks. This approach reduces the impact of a rack power outage or switch failure if any of these hardware failures occurs, the data will remain available.

[](https://en.wikipedia.org/wiki/File:Hadoop_1.png)

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a Job Tracker, Task Tracker, Name Node, and Data Node. A slave acts as both a Data Node and Task Tracker, though it is possible to have data-only and compute-only worker nodes. These are normally used only in nonstandard applications.

Hadoop requires [Java Runtime Environment](https://en.wikipedia.org/wiki/JRE) (JRE) 1.6 or higher. The standard startup and shutdown scripts require that [Secure Shell](https://en.wikipedia.org/wiki/Secure_Shell) (SSH) be set up between nodes in the cluster.

In a larger cluster, HDFS nodes are managed through a dedicated Name Node server to host the file system index, and a secondary Name Node that can generate snapshots of the name node memory structures, thereby preventing file-system corruption and loss of data. Similarly, a standalone Job Tracker server can manage job scheduling across nodes. When Hadoop MapReduce is used with an alternate file system, the Name Node, secondary Name Node, and Data Node architecture of HDFS are replaced by the file-system-specific equivalents.

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware. HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

It is quite expensive to build bigger servers with heavy configurations that handle large scale processing, but as an alternative, you can tie together many commodity computers with single-CPU, as a single functional distributed system and practically, the clustered machines can read the dataset in parallel and provide a much higher throughput. Moreover, it is cheaper than one high-end server. So this is the first motivational factor behind using Hadoop that it runs across clustered and low-cost machines.

Hadoop runs code across a cluster of computers. This process includes the following core tasks that Hadoop performs

1. Data is initially divided into directories and files. Files are divided into uniform sized blocks of 128M and 64M (preferably 128M).
2. These files are then distributed across various cluster nodes for further processing.
3. HDFS, being on top of the local file system, supervises the processing.
4. Blocks are replicated for handling hardware failure.
5. Checking that the code was executed successfully.
6. Performing the sort that takes place between the map and reduce stages.
7. Sending the sorted data to a certain computer.
8. Writing the debugging logs for each job.
   * 1. **ADVANTAGES**
9. No data loss problem.
10. Less time consumption.
11. Accuracy is improved.
12. Efficient data processing.
13. Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
14. Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.
15. Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.
16. Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

**CHAPTER 4**

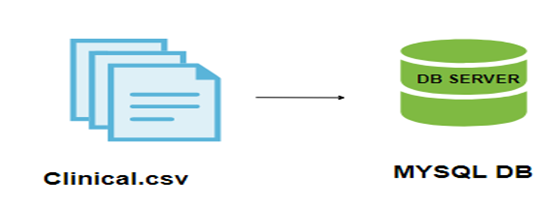
**METHODOLOGY**

Analyzing clinical data by using Hadoop framework along with some Hadoop ecosystems like hdfs, MapReduce, sqoop, hive and pig. By using these tools we can process huge amount of data, no data lost problem, we can get high throughput, maintenance cost also very less and it is a open source software it is compatible on all the platforms since it is Java based.

**4.1 MODULES**

1. PREPROCESSING CLINICAL DATABASE
2. STORAGE
3. ANALYSE QUERY
4. ANALYSIS LATIN SCRIPT (PIG)
5. PROCESSING (MAPREDUCE)
   * 1. **Pre-processing Clinical Database:**

In this module, Analyzing the data with different kinds of fields in Microsoft Excel then it is converted into comma delimited format which is said to be csv (comma separator value) file and moved to MySQL backup through Database**.**



***Fig 4.1 Pre-processing.***

Here by getting historical data we have to convert those historical batch processing data from (.xlsc) format to (.csv) format and by taking backup of all those data in MYSQL Database to avoid loss of data.

Pre-processing is a very important step in data analysis and can heavily influence the modelling results. In the following, we will show the main problems associated with clinical databases. According to Brause et al (2001), a large amount of missing data and uneven sampling times are typical for medical data and should be taken into account in all approaches for medical data diagnosis. To get a good clean data from a medical database, a systematic pre-processing procedure is proposed and is described in more detail.

* **Define features**

Through a systematic approach, identifying all clinical concepts contained in EHR data and defining features conveying each concept, including its type (numerical or categorical) and mechanism to determine feature value; features must account for different data.

* **Process data**

Manipulating the feature set to improve homogeneity and avoid data dispersion by mitigating redundancy (concepts represented with different designations) and granularity (clinical concept is expressed with different levels of detail), which are tackled by combining different features referring to same clinical concept into a single feature.

* **Assess feature values**

Determining the value of each clinical feature (variable) for each dataset instance, by querying the extracted database entries according to the feature types and recording mechanisms.

* + 1. **Storage**

In this module we are getting all those backup data which we have stored in MYSQL and importing all those data by use of Sqoop commands to HDFS (Hadoop Distributed File System). Now all the data are stored in HDFS were it is ready to get processed by use of hive. By storing the huge stock data into HDFS, the solution provided is much more robust, reliable, economical, and scalable. Whenever data size is increasing, you can just add some more nodes, configure into Hadoop and that’s all. If sometime any node is down, then even other nodes are ready to handle the responsibility due to data replication.By managing the Hive schema into embedded database or any other standard SQL database, we are able to utilize the power of SQL as well.

Sqoop (SQL-to-Hadoop) is a big data tool that offers the capability to extract data from non-Hadoop data stores, transform the data into a form usable by Hadoop, and then load the data into HDFS. This process is called ETL, for Extract, Transform, and Load.

While getting data into Hadoop is critical for processing using MapReduce, it is also critical to get data out of Hadoop and into an external data source for use in other kinds of application. Sqoop is able to do this as well.

While it is sometimes necessary to move the data in real time, it is most often necessary to load or unload data in bulk. Like Pig, Sqoop is a command-line interpreter. You type Sqoop commands into the interpreter and they are executed one at a time. Four key features are found in Sqoop:

* **Bulk import:** Sqoop can import individual tables or entire databases into HDFS. The data is stored in the native directories and files in the HDFS file system.
* **Direct input:** Sqoop can import and map SQL (relational) databases directly into Hive and HBase.

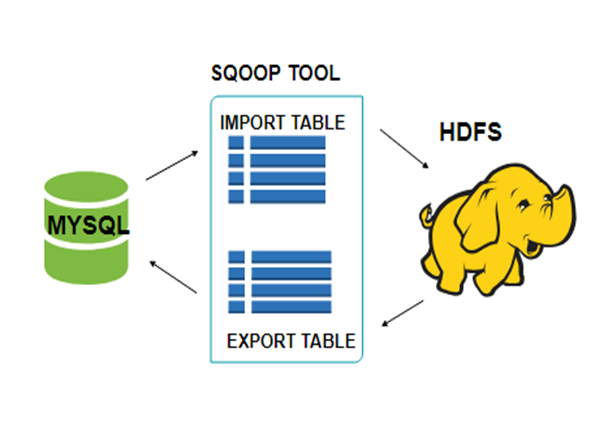
**Data interaction:** Sqoop can generate Java classes so that you can interact with the data programmatically.

* **Data export:** Sqoop can export data directly from HDFS into a relational database using a target table definition based on the specifics of the target database.

Sqoop works by looking at the database you want to import and selecting an appropriate import function for the source data. After it recognizes the input, it then reads the metadata for the table (or database) and creates a class definition of your input requirements.

You can force Sqoop to be very selective so that you get just the columns you are looking for before input rather than doing an entire input and then looking for your data. This can save considerable time. The actual import from the external database to HDFS is performed by a MapReduce job created behind the scenes by Sqoop.

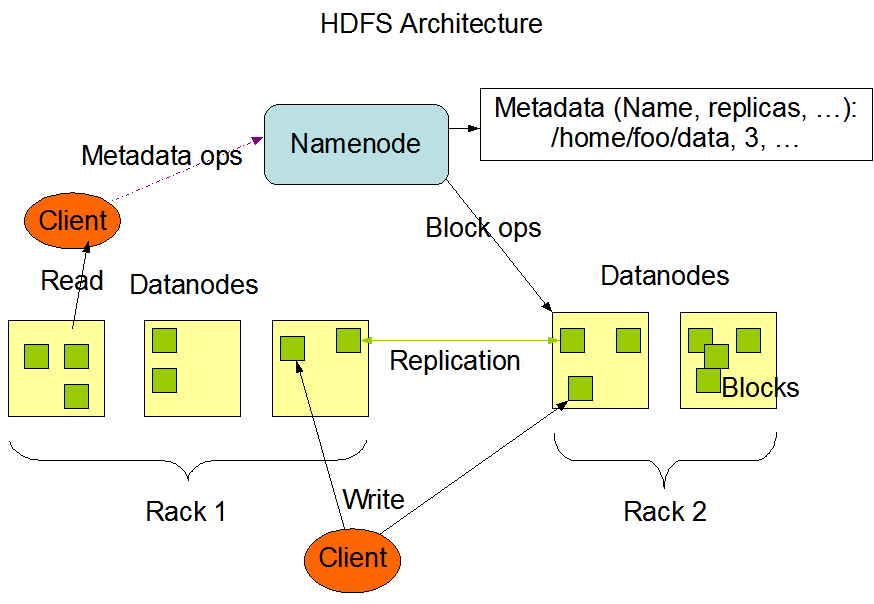
Sqoop is an effective tool for nonprogrammers. The other important item to note is the reliance on underlying technologies like HDFS and MapReduce. You see this repeatedly throughout the element of the Hadoop ecosystem.

****

***Fig 4.2 Storage.***

All the existing **Database Management Systems**are designed with[SQL](https://www.guru99.com/sql.html)standard in mind. However, each DBMS differs with respect to dialect to some extent. So, this difference poses challenges when it comes to data transfers across the systems. Sqoop Connectors are components which help overcome these challenges.

Data transfer between Sqoop and external storage system is made possible with the help of Sqoop's connectors.



***Fig 4.3 HDFS Architecture***

Sqoop has connectors for working with a range of popular relational databases, including MySQL, PostgreSQL, Oracle, SQL Server, and DB2. Each of these connectors knows how to interact with its associated DBMS. There is also a generic JDBC connector for connecting to any database that supports Java's JDBC protocol. In addition, Sqoop provides optimized MySQL and PostgreSQL connectors that use database-specific APIs to perform bulk transfers efficiently.

Analytical processing using Hadoop requires loading of huge amounts of data from diverse sources into Hadoop clusters. This process of bulk data load into Hadoop, from heterogeneous sources and then processing it, comes with a certain set of challenges. Maintaining and ensuring data consistency and ensuring efficient utilization of resources, are some factors to consider before selecting the right approach for data load.

* **DATA LOAD USING SCRIPTS**

The traditional approach of using scripts to load data is not suitable for bulk data load into Hadoop; this approach is inefficient and very time-consuming.

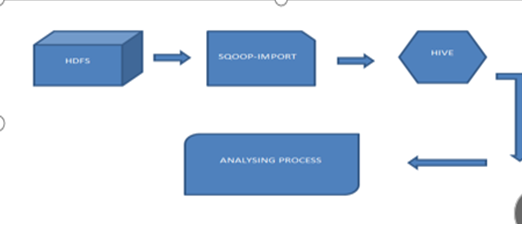
* **DIRECT ACCESS TO EXTERNAL DATA VIA MAP-REDUCE APPLICATION**

Providing direct access to the data residing at external systems (without loading into Hadoop) for map-reduce applications complicates these applications. So, this approach is not feasible.

In addition to having the ability to work with enormous data, Hadoop can work with data in several different forms. So, to load such heterogeneous data into Hadoop, different tools have been developed.

* + 1. **ANALYZE QUERY**

In this module we are getting all those data from HDFS to HIVE by use of Sqoop import command were hive is ready to analyze. Here in HIVE we can process only structured data to analyze. By extracting only, the meaningful data and neglecting unclenched data we can analyze the data in more effective manner by use of hive. The queries are executed by MapReduce, ApacheTez or Apache Spark. Hive queries run on MapReduce take long to run because of batch processing. Spark provides a way to run low latency queries. Spark provides better performance than MapReduce without requiring any changes in queries. Hive is able to access data stored in HDFS. Hive is a data warehouse system for data summarization and analysis and for querying of large data systems in the open-source Hadoop platform. It converts SQL-like queries into MapReduce jobs for easy execution and processing of extremely large volumes of data



***Fig 4.4 Analysing.***

In this module we are getting those data from HDFS to HIVE by use of Sqoop import request. Were hive is set up to separate. Here in HIVE we can process in a general sense regulated data to look at. By restricting only, the gigantic data and blameable unclenched data we can separate the data in magnificently reasonable course by utilization of hive.to process the information that begins from the mapper. In the wake of setting it up, passes on another system of yield, which will be affirmed. The three important functionalities for which Hive is deployed are data summarization, data analysis, and data query. The query language, exclusively supported by Hive, is HiveQL. This language translates SQL-like queries into MapReduce jobs for deploying them on Hadoop. HiveQL also supports MapReduce scripts that can be plugged into the queries. Hive increases schema design flexibility and also data serialization and deserialization.

Hive is best suited for batch jobs, rather than working with web log data and append-only data. It cannot work for online transaction processing (OLTP) systems since it does not provide real-time querying for row-level update

Hive services such as Meta store, File system, and Job Client in turn communicates with Hive storage and performs the following actions

* 1. Metadata information of tables created in Hive is stored in Hive "Meta storage database".
  2. Query results and data loaded in the tables are going to be stored in Hadoop cluster on HDFS.

**JOB EXECTUTION FLOW**

The data flow in Hive behaves in the following pattern:

1. Executing Query from the UI( User Interface)
2. The driver is interacting with Compiler for getting the plan. (Here plan refers to query execution) process and its related metadata information gathering
3. The compiler creates the plan for a job to be executed. Compiler communicating with Meta store for getting metadata request
4. Meta store sends metadata information back to compiler
5. Compiler communicating with Driver with the proposed plan to execute the query
6. Driver Sending execution plans to Execution engine
7. Execution Engine (EE) acts as a bridge between Hive and Hadoop to process the query. For DFS operations.
8. EE should first contacts Name Node and then to Data nodes to get the values stored in tables.
9. EE is going to fetch desired records from Data Nodes. The actual data of tables resides in data node only. While from Name Node it only fetches the metadata information for the query.
10. It collects actual data from data nodes related to mentioned query
11. Execution Engine (EE) communicates bi-directionally with Meta store present in Hive to perform DDL (Data Definition Language) operations. Here DDL operations like CREATE, DROP and ALTERING tables and databases are done. Meta store will store information about database name, table names and column names only. It will fetch data related to query mentioned.
12. Execution Engine (EE) in turn communicates with Hadoop daemons such as Name node, Data nodes, and job tracker to execute the query on top of Hadoop file system



***Fig 4.5 HIVE.***

**DIFFERENT MODES OF HIVE**

Hive can operate in two modes depending on the size of data nodes in Hadoop.

These modes are,

* **Local mode**
* **Map reduce mode**

**WHEN TO USE LOCAL MODE**

1. If the Hadoop installed under pseudo mode with having one data node we use Hive in this mode
2. If the data size is smaller in term of limited to single local machine, we can use this mode
3. Processing will be very fast on smaller data sets present in the local machine

**WHEN TO USE MAP REDUCE MODE**

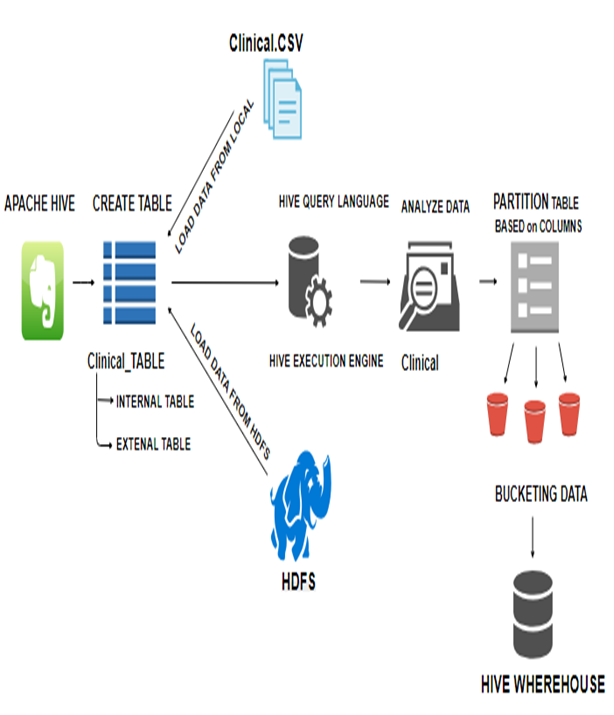
1. If Hadoop is having multiple data nodes and data is distributed across different node we use Hive in this mode
2. It will perform on large amount of data sets and query going to execute in parallel way
3. Processing of large data sets with better performance can be achieved through this mode

In Hive, we can set this property to mention which mode Hive can work? By default, it works on Map Reduce mode and for local mode you can have the following setting. From the Hive version 0.7 it supports a mode to run map reduce jobs in local mode automatically data which is available in hdfs so while processing the data first internally it will going to convert into the (key, value) pair that input will goes to mapper then sorting and shuffling of data will happened and this intermediate data will going to pass to the reducer and internally combiner will combine the data of key and value data and finally it will going to pass to the hdfs for storage purpose.



***Fig 4.6 Map Reduce.***

* 1. It stores schema in a database and processed data into HDFS.
  2. It is designed for OLAP.
  3. It provides SQL type language for querying called HiveQL or HQL.
  4. It is familiar, fast, scalable, and extensible.



***Fig 4.7 Analyzing Query.***

* + 1. **ANALYSIS LATIN SCRIPT (PIG)**

To analyze Clinical using Pig, programmers need to write scripts using Pig Latin language and execute them in interactive mode using the Grunt shell. All these scripts are internally converted to Map and Reduce tasks. After invoking the Grunt shell, you can run your Pig scripts in the shell. Except LOAD and STORE, while performing all other operations, Pig Latin statements take a relation as input and produce another relation as output. As soon as you enter a Load statement in the Grunt shell, its semantic checking will be carried out. To see the contents of the schema, you need to use the Dump operator. Only after performing the dump operation, the Map Reduce job for loading the data into the file system will be carried out. Pig provides many built-in operators to support data operations like grouping, filters, ordering, etc.

The major benefit of PIG is that it works with data that are obtained from various sources and store the results into HDFS (Hadoop Data File System). The programmers have to write the scripts in PIG Latin language which are then converted into Map and reduce tasks with the Pig Engine component (Apache Pig has a component called Pig Engine. It usually accepts the Pig Latin scripts to convert them into MapReduce jobs). Generally, the Apache Pig gives an abstraction to reduce the complexity of developing MapReduce Programming for the developers. The common reason for using the Pig is that it gives hand to write short programs.



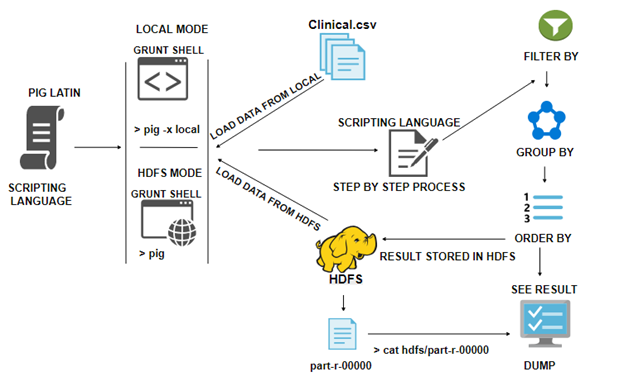
***Fig 4.8 Pig.***

The advanced features of Apache Pig enable the programmers to do more work than other frameworks. This also eases the life of a data engineer in maintaining various ad hoc queries on the data sets. In fact, Apache Pig is a boon for all the programmers and so it is most recommended to use in data management.

The Pig Latin Compiler is defined as the runtime environment which converts the Pig source code into executable code. Generally, most of the executable code exists in the form of MapReduce form.

First of all, the programmers have to write Pig scripts and analyze them. These pig scripts are processed with the help of Apache Pig components such as a parser, optimizer, compiler and finally to the execution engine.

Now, you will be able to get the executable code of Pig Latin which is to be converted into MapReduce tasks. The MapReduce tasks are then stored into the Hadoop Distributed File System (HDFS).

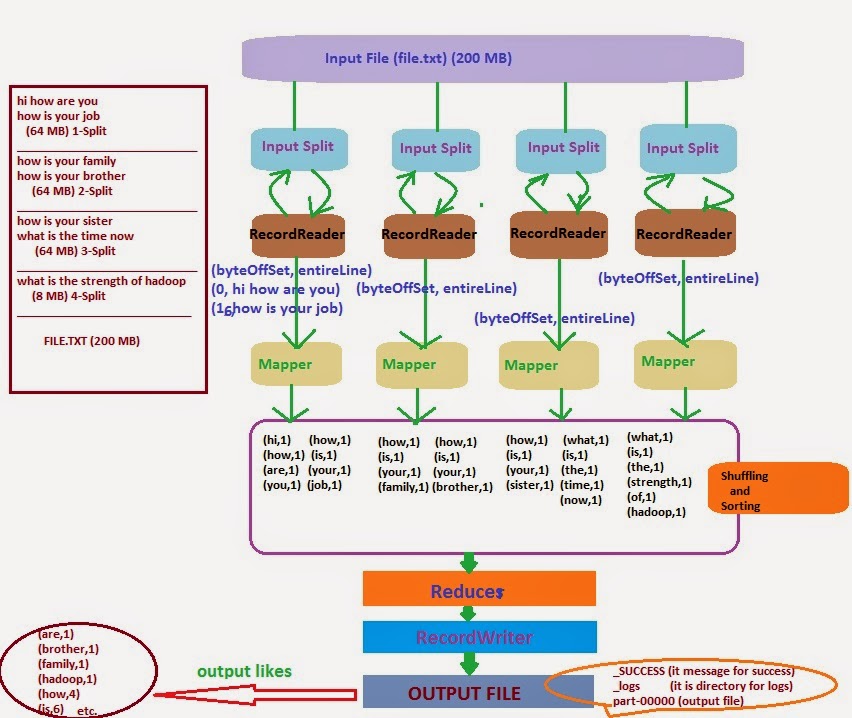


***Fig 4.9 Analysis Of PIG.***

1. **Processing (Map reduce):**

Map Reduce is a framework using which we can write applications to process huge amounts of Clinical, in parallel, on large clusters of commodity hardware in a reliable manner. Map Reduce is a processing technique and a program model for distributed computing based on java. The Map Reduce algorithm contains two important tasks, namely Map and Reduce. Map Reduce program executes in three stages, namely map stage, shuffle stage, and reduce stage. The map or mapper’s job is to process the input data. Generally, the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data. This stage is the combination of the Shuffle stage and the Reduce stage.

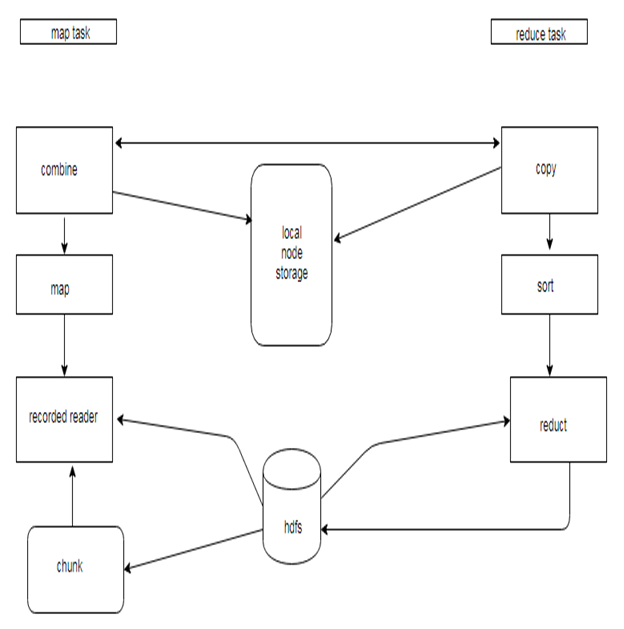
The Reducer’s job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.Since Hive schema is created on a standard SQL database, you get the advantage of running SQL queries on the huge dataset also and are able to process GBs or TBs of data with simple SQL queries. Since actual data resides into HDFS, so these Hive SQL queries are being converted into MapReduce jobs and these parallelized maps reduce jobs process these huge volumes of data and achieve scalable and fault tolerant solutions.



***Fig 4.10 Map Reduce Process.***

One map task is created for each split which then executes map function for each record in the split.

1. It is always beneficial to have multiple splits because the time taken to process a split is small as compared to the time taken for processing of the whole input. When the splits are smaller, the processing is better to load balanced since we are processing the splits in parallel.
2. However, it is also not desirable to have splits too small in size. When splits are too small, the overload of managing the splits and map task creation begins to dominate the total job execution time.
3. For most jobs, it is better to make a split size equal to the size of an HDFS block (which is 64 MB, by default).
4. Execution of map tasks results into writing output to a local disk on the respective node and not to HDFS.
5. Reason for choosing local disk over HDFS is, to avoid replication which takes place in case of HDFS store operation.
6. Map output is intermediate output which is processed by reduce tasks to produce the final output.
7. Once the job is complete, the map output can be thrown away. So, storing it in HDFS with replication becomes overkill.
8. In the event of node failure, before the map output is consumed by the reduce task, Hadoop reruns the map task on another node and re-creates the map output.
9. Reduce task doesn't work on the concept of data locality. An output of every map task is fed to the reduce task. Map output is transferred to the machine where reduce task is running.
10. On this machine, the output is merged and then passed to the user-defined reduce function.
11. Unlike the map output, reduce output is stored in HDFS (the first replica is stored on the local node and other replicas are stored on off-rack nodes). So, writing the reduce output



***Fig 4.11 Map Reduce Architecture***

* 1. **ALGORITHMS**

Introduces the important concept of local aggregation in Map Reduce and strategies for designing efficient algorithms that minimize the amount of partial results that need to be copied across the network. The proper use of combiners is discussed in detail, as well as the “in-mapper combining” design pattern.

Generally MapReduce paradigm is based on sending the computer to where the data resides!

MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

* 1. **MAP STAGE** The map or mapper’s job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
  2. **REDUCE STAGE**: This stage is the combination of the Shuffle stage and the Reduce stage. The Reducer’s job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

**Algorithm 4.2.1**

Word count

The mapper emits an intermediate key-value pair for each word in a document.

The reducer sums up all counts for each word.

1: class Mapper

2: method Map (doc id a, doc d)

3: for all term t ∈ doc d do

4: Emit (term t, count 1)

1: class Reducer

2: method Reduce (term t, counts [c1, c2, . . .])

3: sum ← 0

4: for all count c ∈ counts [c1, c2, . . .] do

5: sum ← sum + c

6: Emit (term t, count sum)

**Algorithm 4.2.2**

1: class Mapper

2: method Map (doc id a, doc d)

3: H ← new Associative Array

4: for all term t ∈ doc d do

5: H{t} ← H{t} + 1. Tally counts for entire document

6: for all term t ∈ H do

7: Emit (term t, count H{t})

**Algorithm 4.2.3**

1: class Mapper

2: method Initialize

3: H ← new Associative Array

4: method Map (doc id a, doc d)

5: for all term t ∈ doc d do

6: H{t} ← H{t} + 1. Tally counts across documents

7: method Close

8: for all term t ∈ H do

9: Emit (term t, count H{t}

* 1. **REQUIREMENT ANALYSIS**

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

**4.3.1 HARDWARE ENVIRONMENT**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

* Hard disk : 40 GB
* Monitor : 15’ colour with VGI card support
* Ram : 4GB DD Ram
* Processor : Pentium iv and above (or) equivalent
* Processor speed : Minimum 500 MHZ
  + 1. **SOFTWARE ENVIRONMENT**

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team ’s and tracking the team’s progress throughout the development activity.

* Operating system : VM Ware
* Front end : MySQL
* Back end : Hadoop

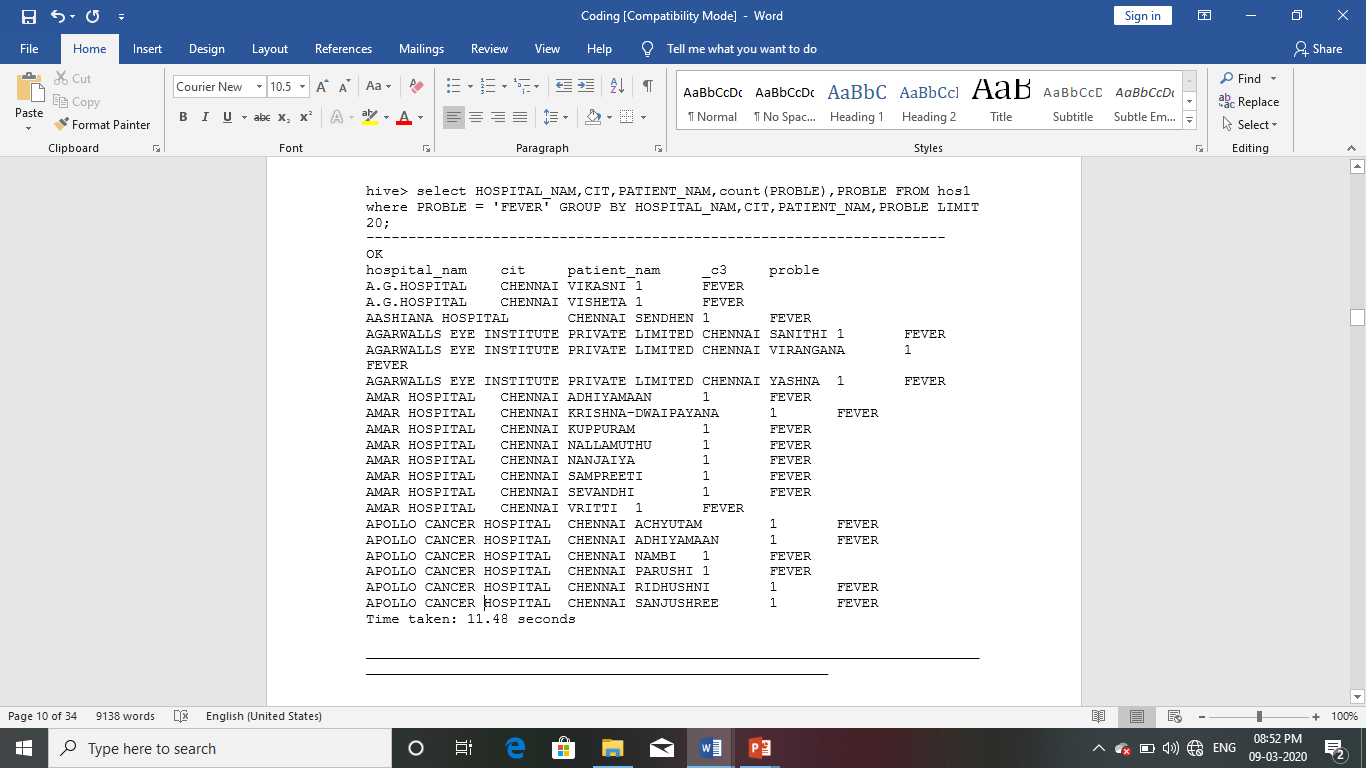
**CHAPTER 4**

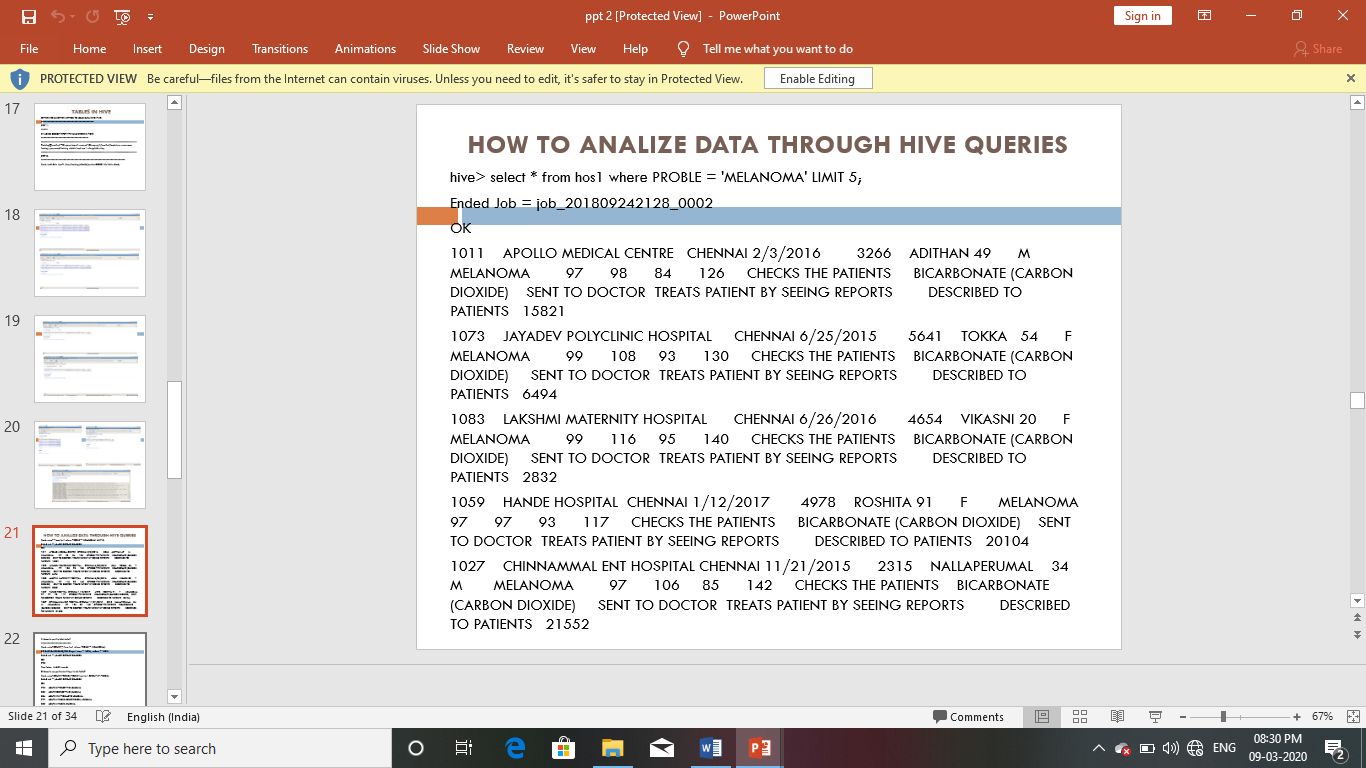
**RESULT**

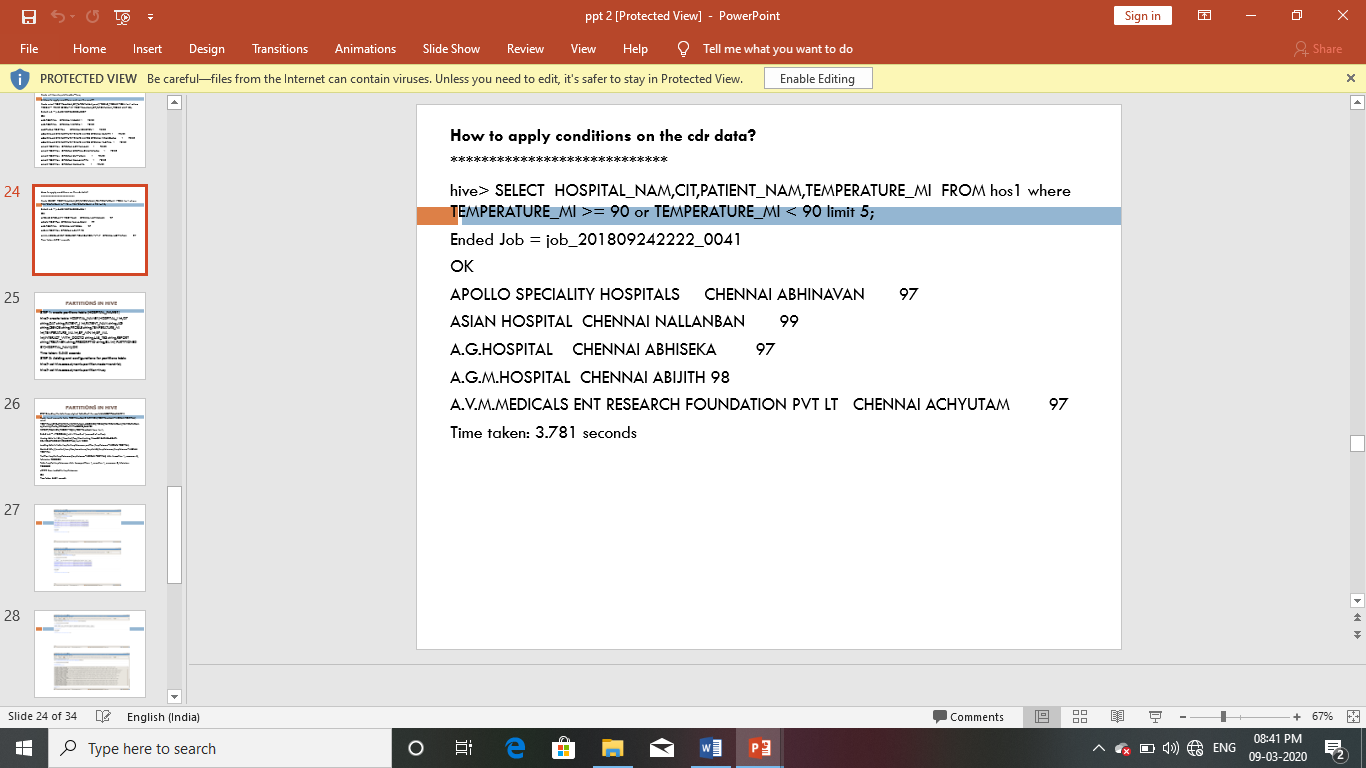
To analyze no limitation of data and simply add number of machines to the cluster and we get results with less time, high throughput and maintenance cost is very less and we are using joins, partitions and bucketing techniques in Hadoop.

Big Data are radically changing research. The unprecedented advances in automated collection of large-scale molecular and clinical data pose major challenges to data analysis and interpretation, calling for the development of new computational approaches. The creation of powerful systems for the effective use of Big Data in Personalized will require significant scientific and technical developments, including infrastructure, engineering, project and financial management. We review here how the evolution of data-driven methods offers the possibility to address many of these problems, guiding the formulation of hypotheses on systems functioning and the generation of mechanistic models, and facilitating the design of clinical procedures.

**OUTPUT**







**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

**CONCLUSION**

This model presented a study on Clinical is to help the doctors by getting the idea of what treatment has been given to that particular patient in the past. A patient can get an idea to see which hospital is treating better for particular problems. Analyze the Clinical data using Hadoop ecosystem where Hive and Pig used for analysis and MapReduce for processing.

**FUTURE ENHANCEMENT**

Apache Spark is an open source processing engine built around speed, case of use, and analytics. If you have large amounts of data that requires low latency processing that a typical Map Reduce program cannot provide, Spark is the alternative. Spark provides in-memory cluster computing for lightning fast speed and supports Java, Scala, and Python APIs for ease of development.

**REFERENCES**

[1] Bhatti U A, Huang M, Zhang Y, et al. Research on the Smartphone Based eHealth Systems for Strengthing Healthcare Organization[M]// Smart Health. 2017.

[2] Black A D, Car J, Pagliari C, et al. “The impact of eHealth on the quality and safety of health care: a systematic overview”, PLoS Medicine, vol. 8, no.1, 2011.

[3] Cao P, Xian Medical University. “The Impact of Big Data on the Development of Chinese Health Service Industry”, Journal of Electronic Test, 2014

[4] Daniel Jarrett, Jinsung Yoon, Member, IEEE, and Mihaela van der Schaar, Fellow, IEEE “Dynamic Prediction in Clinical Survival Analysis using Temporal Convolutional Networks”.

[5] Hermann M, Pentek T, Otto B. Design Principles for Industrie 4.0 Scenarios, pp. 3928-3937, 2016.

[6] Horsky J, Schiff G D, Johnston D, et al. “Methodological Review: Interface design principles for usable decision support: A targeted review of best practices for clinical prescribing interventions”, Journal of Biomedical Informatics, vol. 45, no. 6, pp. 1202-1216, 2012.

[7] L. Lakshmanan, k. VijayaKumar,Parthasarathy, S. Ramani,” Cluster based distributed architecture for prediction of students performance in higher education” Cluster Computing, vol16, pp.5-72019.

[8] M.Akay, D. I. Fotiadis, K. S. Nikita and R. W. Williams, “Guest Editorial: Biomedical Informatics in Clinical Environments,” Biomedical and Health Informatics, IEEE Journal of, vol. 19, no. 1, pp. 149-150, 2015.

[9] MatthewR. Boutella, JieboLuob “Learning multi-label scene classification”,2015.

[10] Mark H. Drazner “The Progression of Hypertensive Heart Disease”,2016.

[11] Thuemmler C, Bai C. Health 4.0: How Virtualization and Big Data are Revolutionizing Healthcare, Springer International Publishing, 2017.

[12] Vancampfort D, Mugisha J, Hallgren M, et al. “The prevalence of diabetes mellitus type 2 in people with alcohol use disorders: a systematic review and large-scale meta-analysis”, Psychiatry Research, vol. 246, pp. 394-400, 2016.

[13] Yu W D, Prakash C, Swati S, et al. “A Modelling Approach to Big Data Based Recommendation Engine in Modern Health Care Environment”, Computer Software and Applications Conference, IEEE Computer Society, pp.75-86, 2015.

**APPENDIX**

1. **SAMPLE CODE**

MySQL> select \* from treat limit 5;

-----------------------------------

+------------+-----------------------------------------------+---------+------------+-----------+-------------+------+-------+-------------+----------------+----------------+--------+-------+---------------------+------------------------------------+----------------+----------------------------------+-----------------------+-------+

| HOSPITAL\_I | HOSPITAL\_NAM | CIT | DAT | PATIENT\_I | PATIENT\_NAM | AG | GENDE | PROBLE | TEMPERATURE\_MI | TEMPERATURE\_MA | BP\_MIN | BP\_MA | INTERACT\_WITH\_DOCTO | LAB\_TES | REPORT | TREATMEN | PRESCRIPTIO | BIL |

+------------+-----------------------------------------------+---------+------------+-----------+-------------+------+-------+-------------+----------------+----------------+--------+-------+---------------------+------------------------------------+----------------+----------------------------------+-----------------------+-------+

| 1000 | APOLLO SPECIALITY HOSPITALS | CHENNAI | 10/29/2015 | 1243 | ABHINAVAN | 26 | M | FEVER | 99 | 100 | 87 | 100 | CHECKS THE PATIENTS | ACTH SUPPRESSION | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 23004 |

| 1001 | ASIAN HOSPITAL | CHENNAI | 12/31/2017 | 1565 | NALLANBAN | 15 | M | BACKPAIN | 100 | 96 | 94 | 108 | CHECKS THE PATIENTS | ADRENOCORTICOTROPIC HORMONE (ACTH) | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 4059 |

| 1002 | A.G. HOSPITAL | CHENNAI | 11/27/2015 | 5564 | ABHISEKA | 45 | M | NEURAL PAIN | 97 | 109 | 91 | 127 | CHECKS THE PATIENTS | ALANINE AMINOTRANSFERASE (ALT) | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 18508 |

| 1003 | A.G.M. HOSPITAL | CHENNAI | 1/19/2015 | 1147 | ABIJITH | 68 | M | APPENDIX | 97 | 96 | 94 | 136 | CHECKS THE PATIENTS | ALBUMIN | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 20115 |

| 1004 | A.V.M. MEDICALS ENT RESEARCH FOUNDATION PVT LT | CHENNAI | 4/11/2017 | 2013 | ACHYUTAM | 10 | M | CANCER | 97 | 107 | 80 | 120 | CHECKS THE PATIENTS | ALKALINE PHOSPHATASE | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 22761 |

+------------+-----------------------------------------------+---------+------------+-----------+-------------+------+-------+-------------+----------------+----------------+--------+-------+---------------------+------------------------------------+----------------+----------------------------------+-----------------------+-------+

5 rows in set (0.01 sec)

=====================================================================================================================================

SQOOP: -

\*\*\*\*\*

[training localhost~] $Sqoop import --connect jdbc: mysql://localhost/movie lens --username training --password training --table treat --m 1 --target-dir.clinic;

----------------------------------------------------------------------------------------------------------------------------------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eval: -

-----

[training localhost~] $Sqoop eval --connect jdbc: mysql://localhost/movielens --username training --password training --query "select \* from treat limit 5"

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

+------------+-----------------------------------------------+---------+------------+-----------+-------------+------+-------+-------------+----------------+----------------+--------+-------+---------------------+------------------------------------+----------------+----------------------------------+-----------------------+-------+

| 1000 | APOLLO SPECIALITY HOSPITALS | CHENNAI | 10/29/2015 | 1243 | ABHINAVAN | 26 | M | FEVER | 99 | 100 | 87 | 100 | CHECKS THE PATIENTS | ACTH SUPPRESSION | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 23004 |

| 1001 | ASIAN HOSPITAL | CHENNAI | 12/31/2017 | 1565 | NALLANBAN | 15 | M | BACKPAIN | 100 | 96 | 94 | 108 | CHECKS THE PATIENTS | ADRENOCORTICOTROPIC HORMONE (ACTH) | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 4059 |

| 1002 | A.G. HOSPITAL | CHENNAI | 11/27/2015 | 5564 | ABHISEKA | 45 | M | NEURAL PAIN | 97 | 109 | 91 | 127 | CHECKS THE PATIENTS | ALANINE AMINOTRANSFERASE (ALT) | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 18508 |

| 1003 | A.G.M. HOSPITAL | CHENNAI | 1/19/2015 | 1147 | ABIJITH | 68 | M | APPENDIX | 97 | 96 | 94 | 136 | CHECKS THE PATIENTS | ALBUMIN | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 20115 |

| 1004 | A.V.M. MEDICALS ENT RESEARCH FOUNDATION PVT LT | CHENNAI | 4/11/2017 | 2013 | ACHYUTAM | 10 | M | CANCER | 97 | 107 | 80 | 120 | CHECKS THE PATIENTS | ALKALINE PHOSPHATASE | SENT TO DOCTOR | TREATS PATIENT BY SEEING REPORTS | DESCRIBED TO PATIENTS | 22761 |

5 rows in set (0.01 sec)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

listing databases:

-------------------

[training localhost~] $Sqoop list-databases --connect jdbc: mysql://localhost/movie lens --username training --password training

18/09/11 01:10:07 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.

18/09/11 01:10:07 INFO manager.MySQLManager: Preparing to use a MySQL streaming result set.

information schema

movie lens

training

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HIVE: -

\*\*\*\*

[training localhost~] $ hive

Hive history file=/tmp/training/hive\_job\_log\_training\_201809242142\_713934136.txt

hive>

hive> show databases;

OK

default

Time taken: 1.618 seconds

=====================================================================================================================================

hive> create database hospital;

OK

Time taken: 0.239 seconds

=====================================================================================================================================

hive> show databases;

OK

default

hospital

Time taken: 0.067 seconds

=====================================================================================================================================

hive> use hospital;

OK

Time taken: 0.067 seconds

=====================================================================================================================================

Create Hive Table: -

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

hive> create table hos1(HOSPITAL\_I int,HOSPITAL\_NAM string, ITstring, AT string,PATIENT\_I int,PATIENT\_NAM string,AG string,GENDE string,PROBLE string,TEMPERATURE\_MI int,TEMPERATURE\_MA int,BP\_MIN int,BP\_MA int,INTERACT\_WITH\_DOCTO string,LAB\_TES string,REPORT string,TREATMEN string,PRESCRIPTIO string,BIL int) row format delimited fields terminated by ',';

OK

Time taken: 0.334 seconds

Loading Data into The Table: -

hive> load data in path '/user/training/clin/part-m-00000' into table hos1;

-----------------------------------------------------------------------------

Loading data to table hospital. hos1

OK

Time taken: 0.399 seconds

HIVE ANALYSIS: -

\*\*\*\*\*\*\*\*\*\*\*\*\*

How to apply conditions and see the result?

++++++++++++++++++++++++++++++++++++++++++++++

hive> select \* from hos1 where PROBLE = 'MELANOMA' LIMIT 5;

----------------------------------------------------------------------

OK

hospital hospital\_namecit dat patient\_i patient\_nam ag gende proble temperature mi temperature\_ma bp\_min bp\_ma interact\_with\_doctolab\_tes report treatmenprescriptio bil

1011 APOLLO MEDICAL CENTRE CHENNAI 12/24/2015 3266 ADITHAN 49 M MELANOMA 100 104 97 131 CHECKS THE PATIENTS BICARBONATE (CARBON DIOXIDE) SENT TO DOCTOR TREATS PATIENT BY SEEING REPORTS DESCRIBED TO PATIENTS 9454

1073 JAYADEV POLYCLINIC HOSPITAL CHENNAI 6/16/2016 5641 TOKKA 54 F MELANOMA 98 116 100 102 CHECKS THE PATIENTS BICARBONATE (CARBON DIOXIDE) SENT TO DOCTOR TREATS PATIENT BY SEEING REPORTS DESCRIBED TO PATIENTS 9669

1083 LAKSHMI MATERNITY HOSPITAL CHENNAI 10/14/2015 4654 VIKASNI 20 F MELANOMA 98 95 86 101 CHECKS THE PATIENTS BICARBONATE (CARBON DIOXIDE)SENT TO DOCTOR TREATS PATIENT BY SEEING REPORTS DESCRIBED TO PATIENTS 24178

1059 HANDE HOSPITAL CHENNAI 3/21/2015 4978 ROSHITA 91 F MELANOMA 99 102 86 109 CHECKS THE PATIENTS BICARBONATE (CARBON DIOXIDE)SENT TO DOCTOR TREATS PATIENT BY SEEING REPORTS DESCRIBED TO PATIENTS 12559

1027 CHINNAMMAL ENT HOSPITAL CHENNAI 8/1/2017 2315 NALLAPERUMAL 34 M MELANOMA 98 103 81 123 CHECKS THE PATIENTS BICARBONATE (CARBON DIOXIDE)SENT TO DOCTOR TREATS PATIENT BY SEEING REPORTS DESCRIBED TO PATIENTS 17890

Time taken: 4.452 seconds

How to apply conditions on the clinical data?

+++++++++++++++++++++++++++++++++++++++++++++

hive> select COUNT (\*) from hos1 where PROBLE = 'MELANOMA';

------------------------------------------------------------

OK

792

Time taken: 14.559 seconds

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

hive> select HOSPITAL\_NAM, CIT, PATIENT\_NAM, count (PROBLE), PROBLE FROM hos1 where PROBLE = 'FEVER' GROUP BY HOSPITAL\_NAM, CIT, PATIENT\_NAM, PROBLE LIMIT 20;

-------------------------------------------------------------------------------------------------------------OK

hospital cit patient \_c3 proble

A.G. HOSPITAL CHENNAI VIKASNI 1 FEVER

A.G. HOSPITAL CHENNAI VISHETA 1 FEVER

AASHIANA HOSPITAL CHENNAI SENDHEN 1 FEVER

AGARWALLS EYE INSTITUTE PRIVATE LIMITED CHENNAI SANITHI 1 FEVER

AGARWALLS EYE INSTITUTE PRIVATE LIMITED CHENNAI VIRANGANA 1 FEVER

AGARWALLS EYE INSTITUTE PRIVATE LIMITED CHENNAI YASHNA 1 FEVER

AMAR HOSPITAL CHENNAI ADHIYAMAAN 1 FEVER

AMAR HOSPITAL CHENNAI KRISHNA-DWAIPAYANA 1 FEVER

AMAR HOSPITAL CHENNAI KUPPURAM 1 FEVER

AMAR HOSPITAL CHENNAI NALLAMUTHU 1 FEVER

AMAR HOSPITAL CHENNAI NANJAIYA 1 FEVER

AMAR HOSPITAL CHENNAI SAMPREETI 1 FEVER

AMAR HOSPITAL CHENNAI SEVANDHI 1 FEVER

AMAR HOSPITAL CHENNAI VRITTI 1 FEVER

APOLLO CANCER HOSPITAL CHENNAI ACHYUTAM 1 FEVER

APOLLO CANCER HOSPITAL CHENNAI ADHIYAMAAN 1 FEVER

APOLLO CANCER HOSPITAL CHENNAI NAMBI 1 FEVER

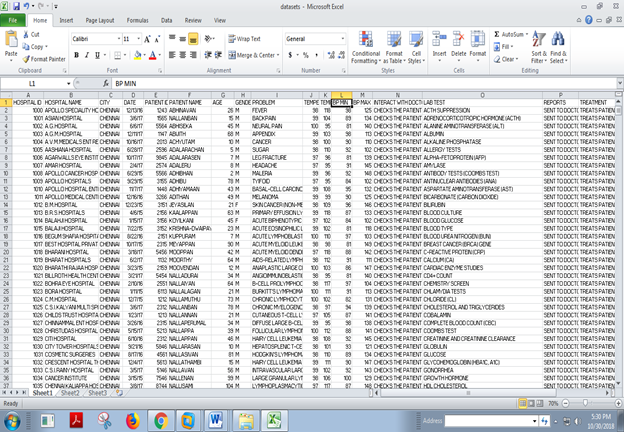
APOLLO CANCER HOSPITAL CHENNAI PARUSHI 1 FEVER

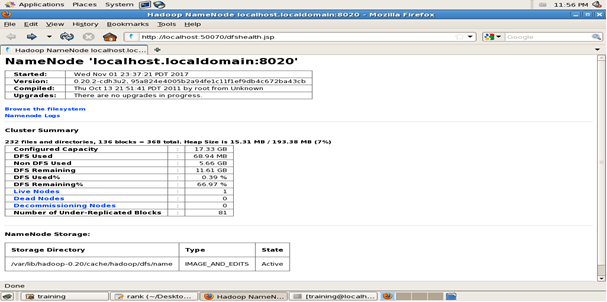
APOLLO CANCER HOSPITAL CHENNAI RIDHUSHNI 1 FEVER

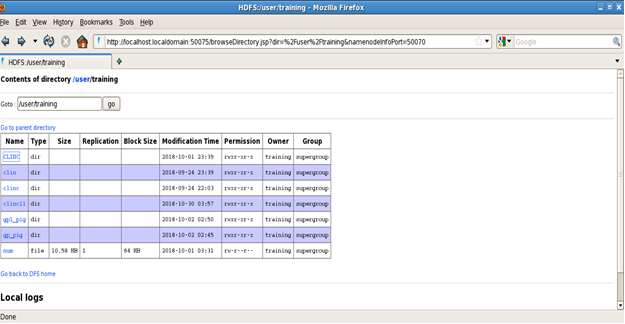
APOLLO CANCER HOSPITAL CHENNAI SANJUSHREE 1 FEVER

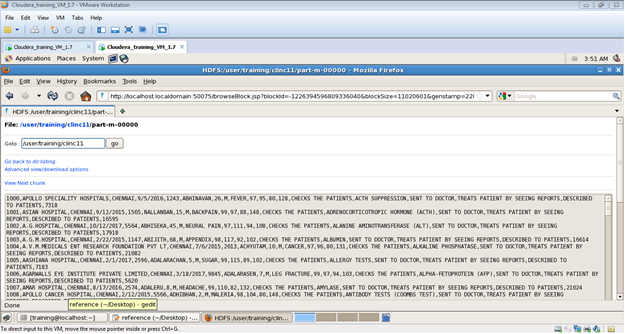
Time taken: 11.48 seconds

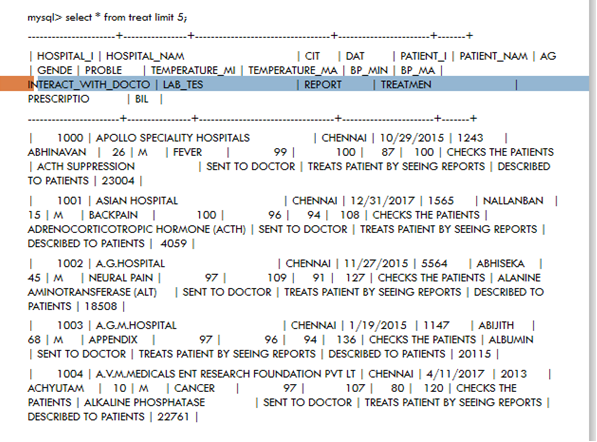
**SCREEN SHOTS**

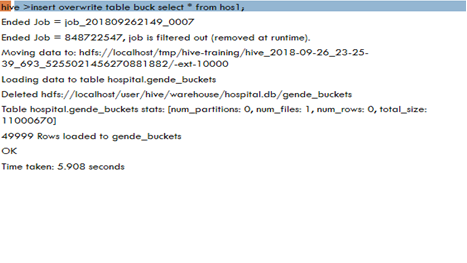
****

****

****

****

****

****

1. **PUBLICATION WITH PLAGIARISM REPORT**