# SYLLABUS: UNIT-I Getting an overview of Big Data: What is Big Data, History of Data Management-evaluation of Big Data, Structuring Big Data, Elements of Big Data, Big Data Analytics. Exploring the Use of Big Data in business Context: Use of Big Data in Social Networking, use of Big Data in preventing fraudulent activities, use of Big Data in detecting fraudulent activities in insurance sector.

# UNIT-II Introducing Technologies for Handling Big Data: Distributed and Parallel Computing for Big Data, Introducing Hadoop, Cloud Computing in Big Data. Understanding Hadoop Eco System: Hadoop Eco System, Hadoop Distributed File System, MapReduce, Hadoop Yarn, Hive, Pig, Sqoop, Zookeeper, Flume, Oozie. UNIT-III Understanding MapReduce Fundamentals and HBase: Map Reduce Frame Work, Techniques to Optimize MapReduceJobs, Use of MapReduce, Role of HBase in Big Data Processing. Exploring Hive : Introducing Hive , Getting Started With Hive, Hive Services, Data Types in Hive, Building Functions in Hive, Hive DDL, Hive DML. Unit-IV Analyzing Data with Pig: Introducing Pig, Running Pig, Getting Started with Pig Latin,Working with Operators in Pig, Working with Functions in Pig. No SQL Data Management: Introduction,Types of No SQL Data Models. UNIT-V Understanding Analytics in Big Data: Comparing, Reporting and Analysis, Types of Analytics, Points to Consider during Analysis and Understanding text Analytics. Applications: Social Media Analytics and Text Mining Mobile Analytics

# What is Big Data Analytics? – Definition, Working, Benefits

Big data analysis uses advanced analytical methods that can extract important business insights from bulk datasets. Within these datasets lies both structured (organized) and unstructured (unorganized) data. Its applications cover different industries such as healthcare, education, insurance, AI, retail, and manufacturing. By analyzing this data, organizations get better insight on what is good and what is bad, so they can make the necessary improvements, develop the production system, and increase profitability.

## **What is Big-Data Analytics?**

Big data analytics is all about crunching massive amounts of information to uncover hidden trends, patterns, and relationships. It’s like sifting through a giant mountain of data to find the gold nuggets of insight.

Here’s a breakdown of what it involves:

* **Collecting Data:**Such data is coming from various sources such as social media, web traffic, sensors and customer reviews.
* **Cleaning the Data:** Imagine having to assess a pile of rocks that included some gold pieces in it. You would have to clean the dirt and the debris first. When data is being cleaned, mistakes must be fixed, duplicates must be removed and the data must be formatted properly.
* **Analysing the Data:** It is here that the wizardry takes place. Data analysts employ powerful tools and techniques to discover patterns and trends. It is the same thing as looking for a specific pattern in all those rocks that you sorted through.

The multi-industrial utilization of big data analytics spans from healthcare to finance to retail. Through their data, companies can make better decisions, become more efficient, and get a competitive advantage.

## **How does big data analytics work?**

## Big Data Analytics is a powerful tool which helps to find the potential of large and complex datasets. To get better understanding, let’s break it down into key steps:

* **Data Collection:**Data is the core of Big Data Analytics. It is the gathering of data from different sources such as the customers’ comments, surveys, sensors, social media, and so on. The primary aim of data collection is to compile as much accurate data as possible. The more data, the more insights.
* **Data Cleaning (Data Preprocessing):** The next step is to process this information. It often requires some cleaning. This entails the replacement of missing data, the correction of inaccuracies, and the removal of duplicates. It is like sifting through a treasure trove, separating the rocks and debris and leaving only the valuable gems behind.
* **Data Processing:** After that we will be working on the data processing. This process contains such important stages as writing, structuring, and formatting of data in a way it will be usable for the analysis. It is like a chef who is gathering the ingredients before cooking. Data processing turns the data into a format suited for analytics tools to process.
* **Data Analysis:** Data analysis is being done by means of statistical, mathematical, and machine learning methods to get out the most important findings from the processed data. For example, it can uncover customer preferences, market trends, or patterns in healthcare data.
* **Data Visualization:** Data analysis usually is presented in visual form, for illustration – charts, graphs and interactive dashboards. The visualizations provided a way to simplify the large amounts of data and allowed for decision makers to quickly detect patterns and trends.
* **Data Storage and Management:**The stored and managed analyzed data is of utmost importance. It is like digital scrapbooking. May be you would want to go back to those lessons in the long run, therefore, how you store them has great importance. Moreover, data protection and adherence to regulations are the key issues to be addressed during this crucial stage.
* **Continuous Learning and Improvement:** Big data analytics is a continuous process of collecting, cleaning, and analyzing data to uncover hidden insights. It helps businesses make better decisions and gain a competitive edge.

## **Types of Big Data Analytics**

Big Data Analytics comes in many different types, each serving a different purpose:

1. [**Descriptive Analytic**](https://www.geeksforgeeks.org/what-are-descriptive-analytics/)**s:**This type helps us understand past events. In social media, it shows performance metrics, like the number of likes on a post.
2. [**Diagnostic Analytics**](https://www.geeksforgeeks.org/what-is-diagnostic-analytics/)**:**In Diagnostic analytics delves deeper to uncover the reasons behind past events. In healthcare, it identifies the causes of high patient re-admissions.
3. [**Predictive Analytics:**](https://www.geeksforgeeks.org/what-is-predictive-analytics-and-how-does-it-work/) Predictive analytics forecasts future events based on past data. Weather forecasting, for example, predicts tomorrow’s weather by analyzing historical patterns.
4. **Prescriptive Analytics:**However, this category not only predicts results but also offers recommendations for action to achieve the best results. In e-commerce, it may suggest the best price for a product to achieve the highest possible profit.
5. **Real-time Analytics:** The key function of real-time analytics is data processing in real time. It swiftly allows traders to make decisions based on real-time market events.
6. **Spatial Analytics:** Spatial analytics is about the location data. In urban management, it optimizes traffic flow from the data unde the sensors and cameras to minimize the traffic jam.
7. **Text Analytics:** Text analytics delves into the unstructured data of text. In the hotel business, it can use the guest reviews to enhance services and guest satisfaction.

## **Big Data Analytics Technologies and Tools**

Big Data Analytics relies on various technologies and tools that might sound complex, let’s simplify them:

* **Hadoop:**Imagine Hadoop as an enormous digital warehouse. It’s used by companies like Amazon to store tons of data efficiently. For instance, when Amazon suggests products you might like, it’s because Hadoop helps manage your shopping history.
* **Spark:**Think of[Spark](https://www.geeksforgeeks.org/introduction-pyspark-distributed-computing-apache-spark/) as the super-fast data chef. Netflix uses it to quickly analyze what you watch and recommend your next binge-worthy show.
* **NoSQL Databases:**NoSQL databases, like[MongoDB](https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/), are like digital filing cabinets that Airbnb uses to store your booking details and user data. These databases are famous because of their quick and flexible, so the platform can provide you with the right information when you need it.
* **Tableau:**Tableau is like an artist that turns data into beautiful pictures. The World Bank uses it to create interactive charts and graphs that help people understand complex economic data.
* **Python and R:** [Python](https://www.geeksforgeeks.org/python-programming-language/) and [R](https://www.geeksforgeeks.org/r-tutorial/) are like magic tools for data scientists. They use these languages to solve tricky problems. For example, Kaggle uses them to predict things like house prices based on past data.
* **Machine Learning Frameworks (e.g., TensorFlow):**In[Machine learning](https://www.geeksforgeeks.org/machine-learning/)frameworks are the tools who make predictions. Airbnb uses[TensorFlow](https://www.geeksforgeeks.org/introduction-to-tensorflow/) to predict which properties are most likely to be booked in certain areas. It helps hosts make smart decisions about pricing and availability.

## **Benefits of Big Data Analytics**

Big Data Analytics offers a host of real-world advantages, and let’s understand with examples:

1. **Informed Decisions:**Imagine a store like Walmart. Big Data Analytics helps them make smart choices about what products to stock. This not only reduces waste but also keeps customers happy and profits high.
2. **Enhanced Customer Experiences:**Think about Amazon. Big Data Analytics is what makes those product suggestions so accurate. It’s like having a personal shopper who knows your taste and helps you find what you want.
3. **Fraud Detection:** Credit card companies, like MasterCard, use Big Data Analytics to catch and stop fraudulent transactions. It’s like having a guardian that watches over your money and keeps it safe.
4. **Optimized Logistics:**FedEx, for example, uses Big Data Analytics to deliver your packages faster and with less impact on the environment. It’s like taking the fastest route to your destination while also being kind to the planet.

## **Challenges of Big data analytics**

While Big Data Analytics offers incredible benefits, it also comes with its set of challenges:

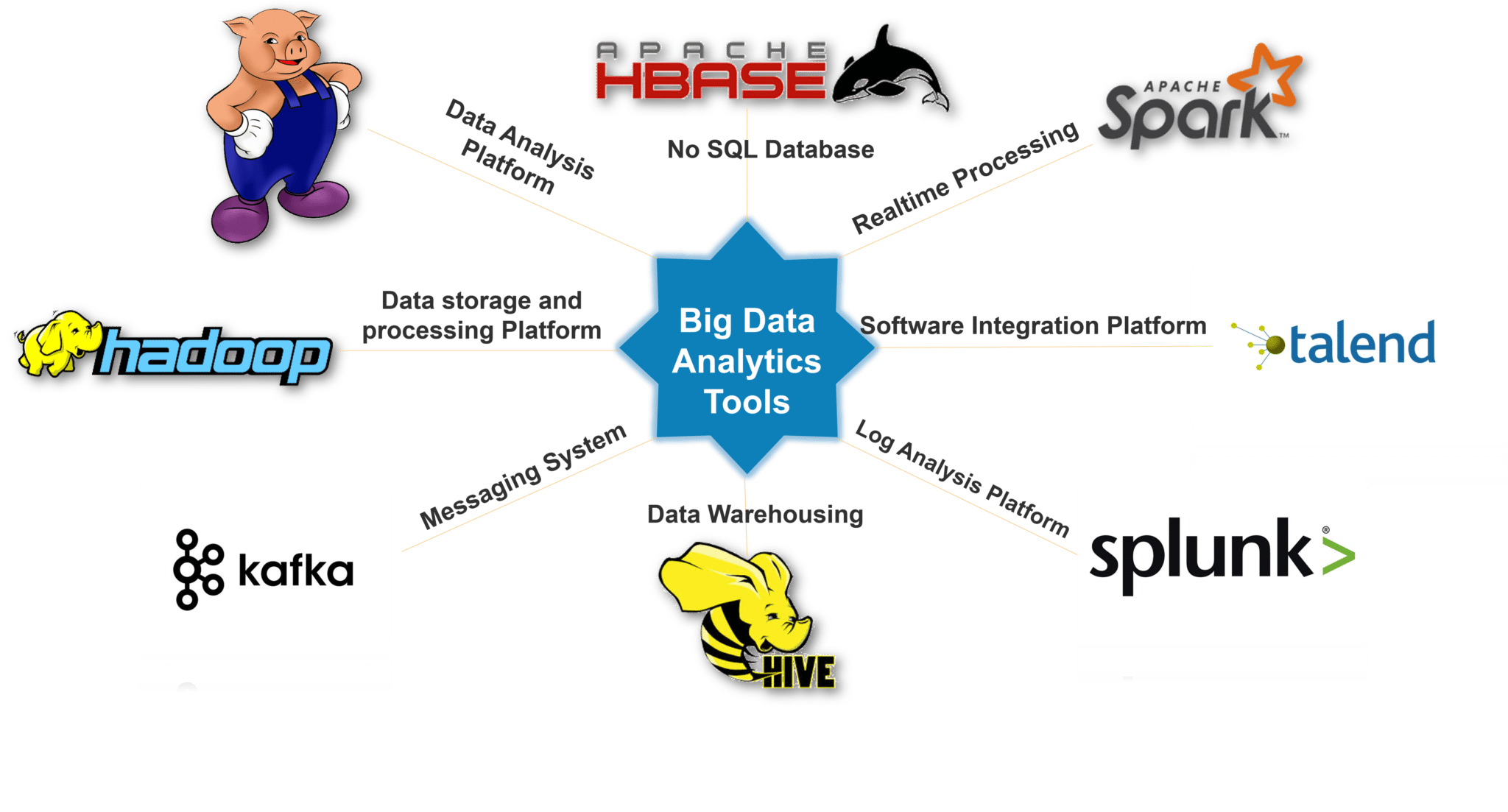
* **Data Overload:**Consider Twitter, where approximately 6,000 tweets are posted every second. The challenge is sifting through this avalanche of data to find valuable insights.
* **Data Quality:** If the input data is inaccurate or incomplete, the insights generated by Big Data Analytics can be flawed. For example, incorrect sensor readings could lead to wrong conclusions in weather forecasting.
* **Privacy Concerns:** With the vast amount of personal data used, like in Facebook’s ad targeting, there’s a fine line between providing personalized experiences and infringing on privacy.
* **Security Risks:** With cyber threats increasing, safeguarding sensitive data becomes crucial. For instance, banks use Big Data Analytics to detect fraudulent activities, but they must also protect this information from breaches.
* **Costs:**Implementing and maintaining Big Data Analytics systems can be expensive. Airlines like Delta use analytics to optimize flight schedules, but they need to ensure that the benefits outweigh the costs.

.

## **Usage of Big Data Analytics**

Big Data Analytics has a significant impact in various sectors:

* **Healthcare:**It aids in precise diagnoses and disease prediction, elevating patient care.
* **Retail:**Amazon’s use of Big Data Analytics offers personalized product recommendations based on your shopping history, creating a more tailored and enjoyable shopping experience.
* **Finance:** Credit card companies such as Visa rely on Big Data Analytics to swiftly identify and prevent fraudulent transactions, ensuring the safety of your financial assets.
* **Transportation:** Companies like Uber use Big Data Analytics to optimize drivers’ routes and predict demand, reducing wait times and improving overall transportation experiences.
* **Agriculture:**Farmers make informed decisions, boosting crop yields while conserving resources.
* **Manufacturing:**Companies like General Electric (GE) use Big Data Analytics to predict machinery maintenance needs, reducing downtime and enhancing operational efficiency.



## **History of Big Data**

The term ‘Big Data’ has been in use since the early 1990s. Although it is not exactly known who first used the term, most people credit John R. Mashey (who at the time worked at Silicon Graphics) for making the term popular.[[i]](https://www.bigdataframework.org/knowledge/a-short-history-of-big-data/" \l "_edn1) Big Data is now a well-established knowledge domain, both in academics as well as in industry.

In order to best understand how Big Data was able to grow to such popularity, it is important to place Big Data into its historic perspective. From a knowledge domain perspective, Big Data is the combination of the very mature domain of statistics with the relatively young domain of computer science. As such, it builds upon the collective knowledge of mathematics, statistics and data analysis techniques in general.

Ever since the early beginnings of civilization, people have tried to use ‘data’ towards better decision making, or to gain a competitive (or military) advantage. This quest can even be dated back to the ancient Egyptians and the Roman Empire. The famous Library of Alexandria, which was established around 300 B.C., can be considered as a first attempt by the ancient Egyptians to capture all ‘data’ within the empire. It is estimated that the library consisted of 40,000 to 400,000 scrolls (which would be the equivalent of around 100,000 books). Even the ancient leaders of the world realized that combining different data sources could result in an advantage over other competing empires.

Other well documented use cases of the first forms of data analysis come from the Roman empire. The ancient Roman military utilized very detailed statistical analysis to ‘predict’ at which border the chance of an enemy insurgency would be the most prevalent. Based on these analyses, they were able to deploy their armies in the most efficient way possible. It is not a far stretch to consider these calculations one of the earliest forms of ‘predictive’ data analysis. And again, these analysis techniques provided the Roman military with an advantage over other armies.

In order to understand the world of Big Data, it is therefore important to realize that most techniques that are used today (from predictive algorithms to classification techniques) have been developed centuries ago, and that Big Data continues to build on the work of some of the greatest minds in history. The key aspect that has changed, of course, is the availability and accessibility to massive quantities of data. Whereas up until the 1950s, most data analysis was done manually and on paper, we now have the technology and capability to analyse terabytes of data within split seconds.

Especially since the beginning of the 21st century, the volume and speed with which data is generated has changed beyond measures of human comprehension. The total amount of data in the world was 4.4 zettabytes in 2013. That is set to rise steeply to 44 zettabytes by 2020.[[iii]](https://www.bigdataframework.org/knowledge/a-short-history-of-big-data/" \l "_edn3) To put that in perspective, 44 zettabytes are the equivalent to 44 trillion gigabytes. Even with the most advanced technologies today, it is impossible to analyse all this data. The need to process these increasingly larger (and unstructured) data sets is how traditional data analysis transformed into ‘Big Data’ in the last decade.

[](https://www.bigdataframework.org/wp-content/uploads/2021/03/Overview-of-Data-Volumes.png)

Figure: Data and the volume of data in Perspective.

**The evolution of Big Data**

The evolution of Big Data can roughly be subdivided into three main phases.[[iv]](https://www.bigdataframework.org/knowledge/a-short-history-of-big-data/" \l "_edn4) Each phase was driven by technological advancements and has its own characteristics and capabilities. In order to understand the context of Big Data today, it is important to understand how each of these phases contributed to the modern meaning of Big Data.

### Big Data Phase 1 – Structured Content

Data analysis, data analytics and Big Data originate from the longstanding domain of database management. It relies heavily on the storage, extraction, and optimization techniques that are common in data that is stored in Relational Database Management Systems (RDBMS). The techniques that are used in these systems, such as structured query language (SQL) and the extraction, transformation and loading (ETL) of data, started to professionalize in the 1970s.

Database management and data warehousing systems are still fundamental components of modern-day Big Data solutions. The ability to quickly store and retrieve data from databases or find information in large data sets, is still a core requirement for the analysis of Big Data. Relational database management technology and other data processing technologies that were developed during this phase, are still strongly embedded in the Big Data solutions from leading IT vendors, such as Microsoft, Google and Amazon. A number of core technologies and characteristics of this first phase in the evolution of Big Data is outlined in figure 3.

### Big Data Phase 2 – Web Based Unstructured Content

From the early 2000s, the internet and corresponding web applications started to generate tremendous amounts of data. In addition to the data that these web applications stored in relational databases, IP-specific search and interaction logs started to generate web based unstructured data. These unstructured data sources provided organizations with a new form of knowledge: insights into the needs and behaviours of internet users. With the expansion of web traffic and online stores, companies such as Yahoo, Amazon and eBay started to analyse customer behaviour by analysing click-rates, IP-specific location data and search logs, opening a whole new world of possibilities.

From a technical point of view, HTTP-based web traffic introduced a massive increase in semi-structured and unstructured data (further discussed in chapter 1.6). Besides the standard structured data types, organizations now needed to find new approaches and storage solutions to deal with these new data types in order to analyse them effectively. The arrival and growth of social media data greatly aggravated the need for tools, technologies and analytics techniques that were able to extract meaningful information out of this unstructured data. New technologies, such as networks analysis, web-mining and spatial-temporal analysis, were specifically developed to analyse these large quantities of web based unstructured data effectively.

### Big Data Phase 3 – Mobile and Sensor-based Content

The third and current phase in the evolution of Big Data is driven by the rapid adoption of mobile technology and devices, and the data they generate.The number of mobile devices and tablets surpassed the number of laptops and PCs for the first time in 2011. In 2020, there are an estimated 10 billion devices that are connected to the internet. And all of these devices generate data every single second of the day.

Mobile devices not only give the possibility to analyse behavioural data (such as clicks and search queries), but they also provide the opportunity to store and analyse location-based GPS data. Through these mobile devices and tablets, it is possible to track movement, analyse physical behaviour and even health-related data (for example the number of steps you take per day). And because these devices are connected to the internet almost every single moment, the data that these devices generate provide a real-time and unprecedented picture of people’s behaviour.

Simultaneously, the rise of sensor-based internet-enabled devices is increasing the creation of data to even greater volumes. Famously coined the ‘Internet of Things’ (IoT), millions of new TVs, thermostats, wearables and even refrigerators are connected to the internet every single day, providing massive additional data sets. Since this development is not expected to stop anytime soon, it could be stated that the race to extract meaningful and valuable information out of these new data sources has only just begun. A summary of the evolution of Big Data and its key characteristics per phase is outlined in figure 3.

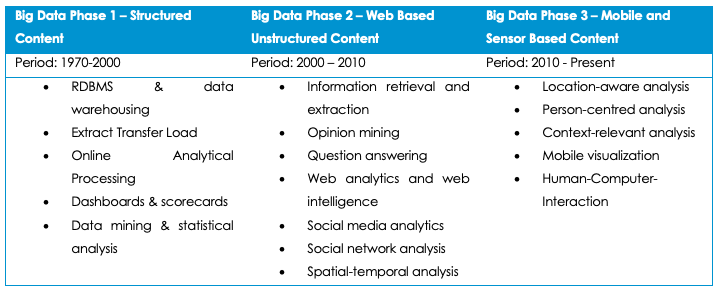
[](https://www.bigdataframework.org/wp-content/uploads/2020/11/Three-Phases-in-the-evolution-of-big-data.png)

Figure: The Three Major Phases in the evolution of Big Data

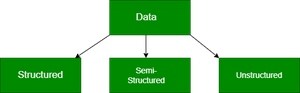
# Types of Big Data

2.5 quintillion bytes of data are generated every day by users. Predictions by Statista suggest that by the end of 2021, 74 Zettabytes (74 trillion GBs) of data would be generated by the internet. Managing such a vacuous and perennial outsourcing of data is increasingly difficult. So, to manage such huge complex data, Big data was introduced, it is related to the extraction of large and complex data into meaningful data which can’t be extracted or analyzed by traditional methods.

All data cannot be stored in the same way. The methods for data storage can be accurately evaluated after the type of data has been identified. A Cloud Service, like Microsoft Azure, is a one-stop destination for storing all kinds of data; blobs, queues, files, tables, disks, and applications data. However, even within the Cloud, there are special services to deal with specific sub-categories of data. For example, Azure Cloud Services like Azure SQL and Azure Cosmos DB help in handling and managing sparsely varied kinds of data.

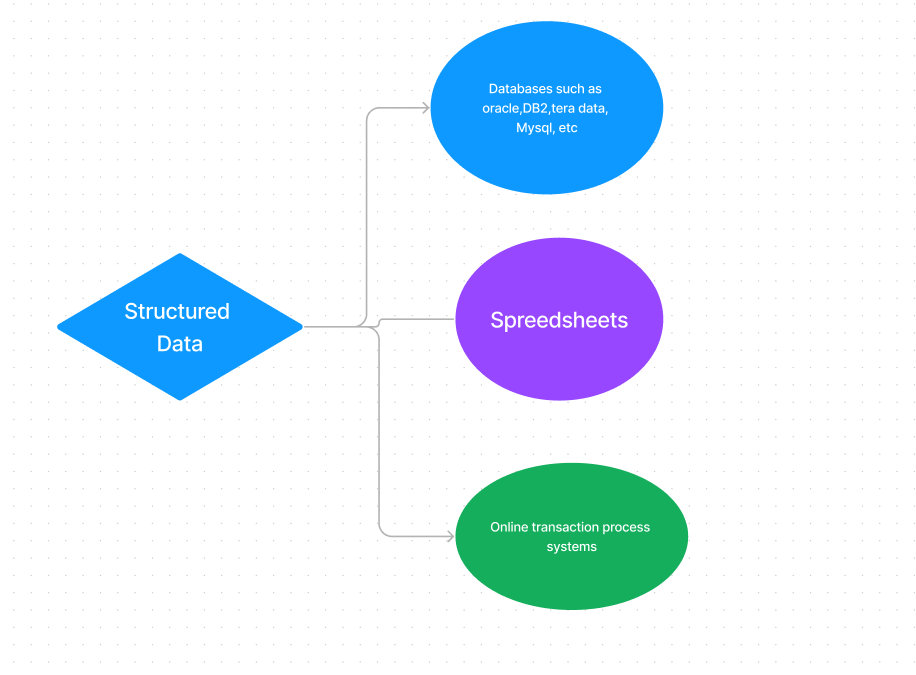
Applications Data is the data that is created, read, updated, deleted, or processed by applications. This data could be generated via web apps, android apps, iOS apps, or any applications whatsoever. Due to a varied diversity in the kinds of data being used, determining the storage approach is a little nuanced.

## **Types of Big Data**



### Structured Data

* Structured data can be crudely defined as the data that resides in a fixed field within a record.
* It is type of data most familiar to our everyday lives. for ex: birthday,address
* A certain schema binds it, so all the data has the same set of properties. Structured data is also called relational data. It is split into multiple tables to enhance the integrity of the data by creating a single record to depict an entity. Relationships are enforced by the application of table constraints.
* The business value of structured data lies within how well an organization can utilize its existing systems and processes for analysis purposes.



Sources of structured data

A Structured Query Language (SQL) is needed to bring the data together. Structured data is easy to enter, query, and analyze. All of the data follows the same format. However, forcing a consistent structure also means that any alteration of data is too tough as each record has to be updated to adhere to the new structure. Examples of structured data include numbers, dates, strings, etc. The business data of an e-commerce website can be considered to be structured data.

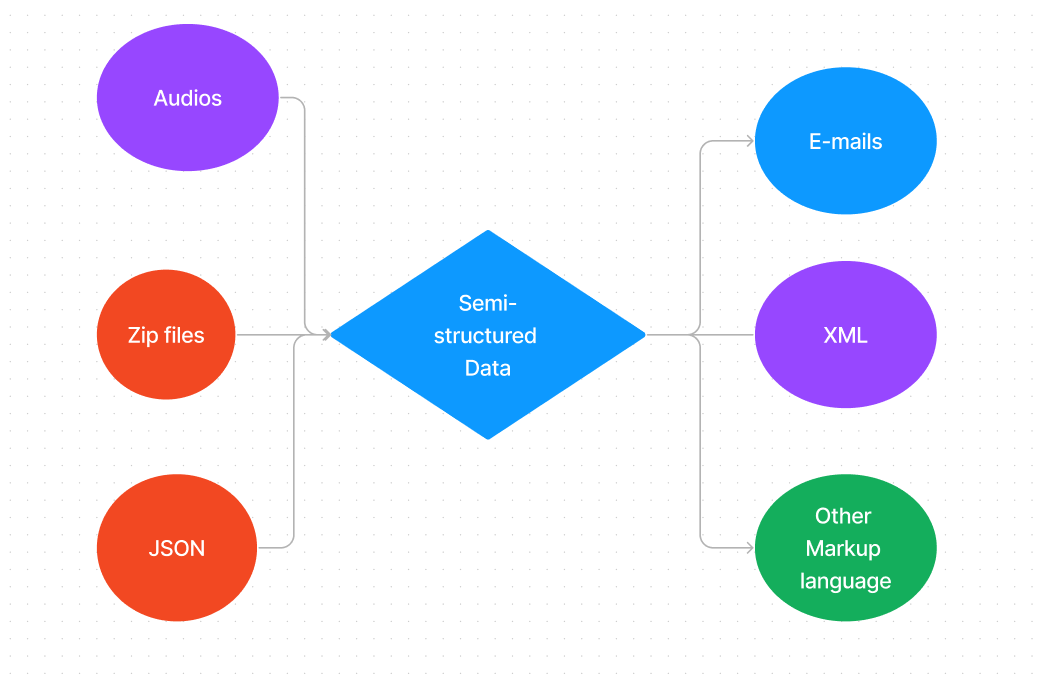
| **Name** | **Class** | **Section** | **Roll No** | **Grade** |
| --- | --- | --- | --- | --- |
| Geek1 | 11 | A | 1 | A |
| Geek2 | 11 | A | 2 | B |
| Geek3 | 11 | A | 3 | A |

#### Cons of Structured Data

1. Structured data can only be leveraged in cases of predefined functionalities. This means that structured data has limited flexibility and is suitable for certain specific use cases only.
2. Structured data is stored in a data warehouse with rigid constraints and a definite schema. Any change in requirements would mean updating all of that structured data to meet the new needs. This is a massive drawback in terms of resource and time management.

### Semi-Structured Data

* Semi-structured data is not bound by any rigid schema for data storage and handling. The data is not in the relational format and is not neatly organized into rows and columns like that in a spreadsheet. However, there are some features like key-value pairs that help in discerning the different entities from each other.
* Since semi-structured data doesn’t need a structured query language, it is commonly called NoSQL data.
* A data serialization language is used to exchange semi-structured data across systems that may even have varied underlying infrastructure.
* Semi-structured content is often used to store metadata about a business process but it can also include files containing machine instructions for computer programs.
* This type of information typically comes from external sources such as social media platforms or other web-based data feeds.

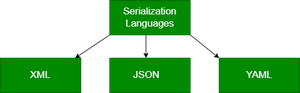


**Semi-Structured Data**

Data is created in plain text so that different text-editing tools can be used to draw valuable insights. Due to a simple format, data serialization readers can be implemented on hardware with limited processing resources and bandwidth.

#### Data Serialization Languages

Software developers use **serialization languages** to write memory-based data in files, transit, store, and parse. The sender and the receiver don’t need to know about the other system. As long as the same serialization language is used, the data can be understood by both systems comfortably. There are three predominantly used Serialization languages.



#### 

1. **XML**– XML stands for eXtensible Markup Language. It is a text-based markup language designed to store and transport data. XML parsers can be found in almost all popular development platforms. It is human and machine-readable. XML has definite standards for schema, transformation, and display. It is self-descriptive. Below is an example of a programmer’s details in XML.

XML

|  |
| --- |
| <ProgrammerDetails>      <FirstName>Jane</FirstName>      <LastName>Doe</LastName>      <CodingPlatforms>          <CodingPlatform Type="Fav">GeeksforGeeks</CodingPlatform>          <CodingPlatform Type="2ndFav">Code4Eva!</CodingPlatform>          <CodingPlatform Type="3rdFav">CodeisLife</CodingPlatform>     </CodingPlatforms>  </ProgrammerDetails>    <!--The 2ndFav and 3rdFav Coding Platforms are imaginative because Geeksforgeeks is the best!--> |

XML expresses the data using tags (text within angular brackets) to shape the data (for ex: FirstName) and attributes (For ex: Type) to feature the data. However, being a verbose and voluminous language, other formats have gained more popularity.

1. **JSON**–  JSON (JavaScript Object Notation) is a lightweight open-standard file format for data interchange. JSON is easy to use and uses human/machine-readable text to store and transmit data objects.

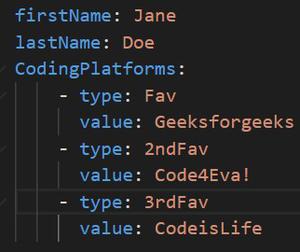
Javascript

|  |
| --- |
| {      "firstName": "Jane",      "lastName": "Doe",      "codingPlatforms": [          { "type": "Fav", "value": "Geeksforgeeks" },          { "type": "2ndFav", "value": "Code4Eva!" },          { "type": "3rdFav", "value": "CodeisLife" }      ]  } |

This format isn’t as formal as XML. It’s more like a key/value pair model than a formal data depiction. Javascript has inbuilt support for JSON. Although JSON is very popular amongst web developers, non-technical personnel find it tedious to work with JSON due to its heavy

dependence on JavaScript and structural characters (braces, commas, etc.)

**3.** **YAML**– YAML is a user-friendly data serialization language. Figuratively, it stands for YAML Ain’t Markup Language. It is adopted by technical and non-technical handlers all across the globe owing to its simplicity. The data structure is defined by line separation and indentation and reduces the dependency on structural characters. YAML is extremely comprehensive and its popularity is a result of its human-machine readability.

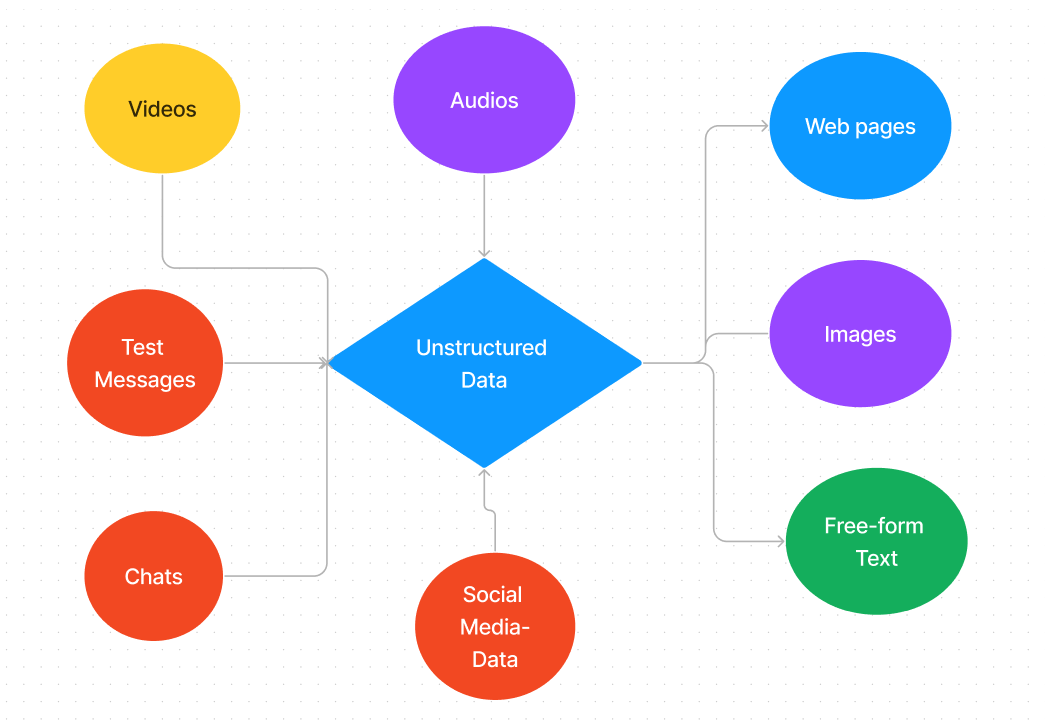


YAML example

A product catalog organized by tags is an example of semi-structured data.

### Unstructured Data

* Unstructured data is the kind of data that doesn’t adhere to any definite schema or set of rules. Its arrangement is unplanned and haphazard.
* Photos, videos, text documents, and log files can be generally considered unstructured data. Even though the metadata accompanying an image or a video may be semi-structured, the actual data being dealt with is unstructured.
* Additionally, Unstructured data is also known as “dark data” because it cannot be analyzed without the proper software tools.



**Un-structured Data**

## Summary

Applications data can be classified as structured, semi-structured, and unstructured data. Structured data is neatly organized and obeys a fixed set of rules. Semi-structured data doesn’t obey any schema, but it has certain discernible features for an organization. Data serialization languages are used to convert data objects into a byte stream. These include XML, JSON, and YAML. Unstructured data doesn’t have any structure at all. All these three kinds of data are present in an application. All three of them play equally important roles in developing resourceful and attractive applications.

**How Is Social Media Being Affected by Big Data?**

The extensive use of these big data tactics is clearly demonstrated by the influx of posts, comments, likes, dislikes, followings, and followers from social media sources, such as the top 3 leaders - Facebook, Youtube, and Instagram. Facebook is not going away, as evidenced by Statista's estimate that it had 2.38 billion active monthly users in the first quarter of 2019.

Operating these massive amounts of information created every single second is crucial. Successful firms pay attention to what their consumers say because both positive and bad comments can affect their ability to attract new customers and maintain their good name.

Big data is essential to marketing analytics' ability to forecast future customer behavior without exaggeration. Many businesses invest in big data solution technologies to track customers' experiences in social media in real-time.

**Advantages of Using Big Data in Social Media:**

advantages of big data analytics for social media marketing.

1. Channels of communication:

AI strategies enable the processing of data from various channels, particularly when synchronization and a widely used log-in technology are used. Many business websites encourage users to sign up using Google or Facebook accounts, allowing marketers to access data from social media activity, browser history, desktop and mobile applications, cloud storage, and other sources to learn more about their customers.

1. Real-time communication:

The key to a successful market study is user behavior on social media, such as advertising clicked, pages visited and followed, comments left, links saved, and friends added. No other source can provide a more accurate and current picture of market demand. The most important thing is to take advantage of the circumstance earlier than competitors because it changes so quickly.

1. Intended audience:

Similar to other company endeavors, social media marketing aims to boost sales, but it serves no use in feeding vegan meat. Knowing your intended audience is crucial, therefore. The breadth of ML solutions allows for extracting useful insights from various social network activities, including millions of photographs, music preferences, locations, and many other activities.

1. Future forecasts:

Using big data strategy and predictive analytics in the media allows for better decision-making based on historical data. Data-driven businesses frequently achieve great success because computers can predict future customer preferences. Even if they evolve over time, habits and interests generally stay connected. Following a purchase on social media, there is a strong likelihood that the consumer will select related goods.

1. Security concerns:

Private information is extremely important to customers due to the rise of social media and the public presentation of personal information, weird as it may sound. Although there is still much need for improvement in this area, most businesses give security concerns a top priority. Data vendors, marketers, and business owners must provide data security against leaks to unauthorized third parties. Different forms of protection are suggested by big data solutions, such as voice and facial recognition, authorization, check-in notifications, etc.

1. Campaign analysis:

The seesaw dynamics of ROI indicators may be properly tracked thanks to big data analytics. Marketers can learn more about a social media campaign's success. Predictive analytics tools excel when it comes to predicting the goods and services that customers will demand. Measuring user interactions and responses to online advertisements across various social media platforms can reveal much about consumer behavior and purchasing habits. Overall, the success or failure of a campaign can be predicted based on past customer behavior gathered from social media, historical website data, email subscriptions, and other forms of digital contact.

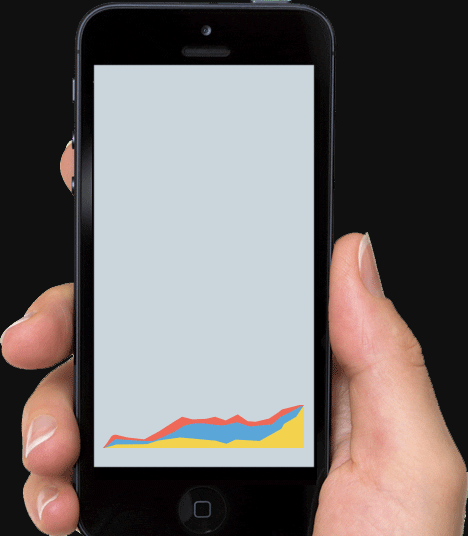
1. Affordable costs:

Because so many elements must be considered, pricing selections can occasionally be difficult. Typically, it begins with product costs, problems with competition, market demand, positive revenue, levels of currency and inflation, and finishes with a global economic scenario. In order to fully understand how much your loyal customers are willing to spend on your products, a solid Big Data strategy on social media should not only involve lavish payments to your Instagram influencers. It should also involve regular communication with these customers, perhaps through A/B testing or online surveys. All of this can assist marketers in making more precise and flexible price adjustments to meet client expectations.

1. Innovation potential:

Through media monitoring, businesses can thoroughly grasp their goods and target market using data science Tools for social media analytics that can be set up to find market-wide capability gaps. For instance, user input expressing a need for lighter, more relaxed running shoes helped to propel the minimalist innovation in the market for running shoes. The most prosperous businesses in recent years have been those that can mine consumer feedback from social media platforms and use it to reinvent their businesses.

# How big data analytics is used in social media



There are many social media platforms run by different companies in the world. Not to forget, the large number of social media users that have been added in recent years. With the rise of social media, the amount of data produced by different platforms is unmatchable. The likes, shares, and comments across social media platforms contain information regarding user behavior.

It is why business organizations are using big data analytics to make the best use of the data available on social media platforms. **Big data analytics** is widely used in social media to shape marketing strategies and much more. Read on to know seven ways how **big data analytics** is used in social media.

* **Omnichannel presence**

Many business applications and websites have a social media integration. Customers can log in to a business application using their social media credentials. It helps businesses to collect customer data from social media platforms and use them to provide better services. You can get access to social media posts, browser history, and much more. Since customers have an omnichannel presence, you can collect data from all sources to know more about the preferences of customers.

* **Real-time activity monitoring**

Social media is a place where you immediately get to know when someone has liked a post or shared a product link. Businesses monitor the activity of customers on social media to know about their current mood. If a social media user is liking your product posts, you can show them an email quickly to convert them into a customer. No other platform can inform customer preferences in real-time other than social media. Big data is used extensively to collect real-time activity reports on social media.

* **Forecasting**

When big data is mixed with modern-day technologies like ML and AI, it can predict customer preferences. Based on customer habits on social media, AI/ML algorithms predict their demands. Businesses then focus on releasing new products/services as per the future demands. For example, when a customer buys something online, the chances of them buying similar products increase.

* **Security**

Data vendors cannot illegally transfer customer data to the wrong hands. When customers share data on social media, that data can only be used for business purposes. Your social media data cannot be placed in the wrong hands by business organizations. Big data is used for enhancing the security of social media platforms based on customer suggestions.

* **Campaign monitoring**

Marketers run social media campaigns to boost their ROI (Return on Investment). Using big data, marketers can know how well a social media campaign has performed. Young aspirants can go for [**big data training**](https://imarticus.org/postgraduate-program-in-data-science-analytics/) to know more about how to run social media campaigns and study high-end analytics.

* **Product pricing**

When a firm launches its product on social media, customers give their valuable opinions. Social media is widely used to determine whether customers are satisfied with the pricing of a product or not. **Big data training** includes data collection from social media channels and how to analyze them.

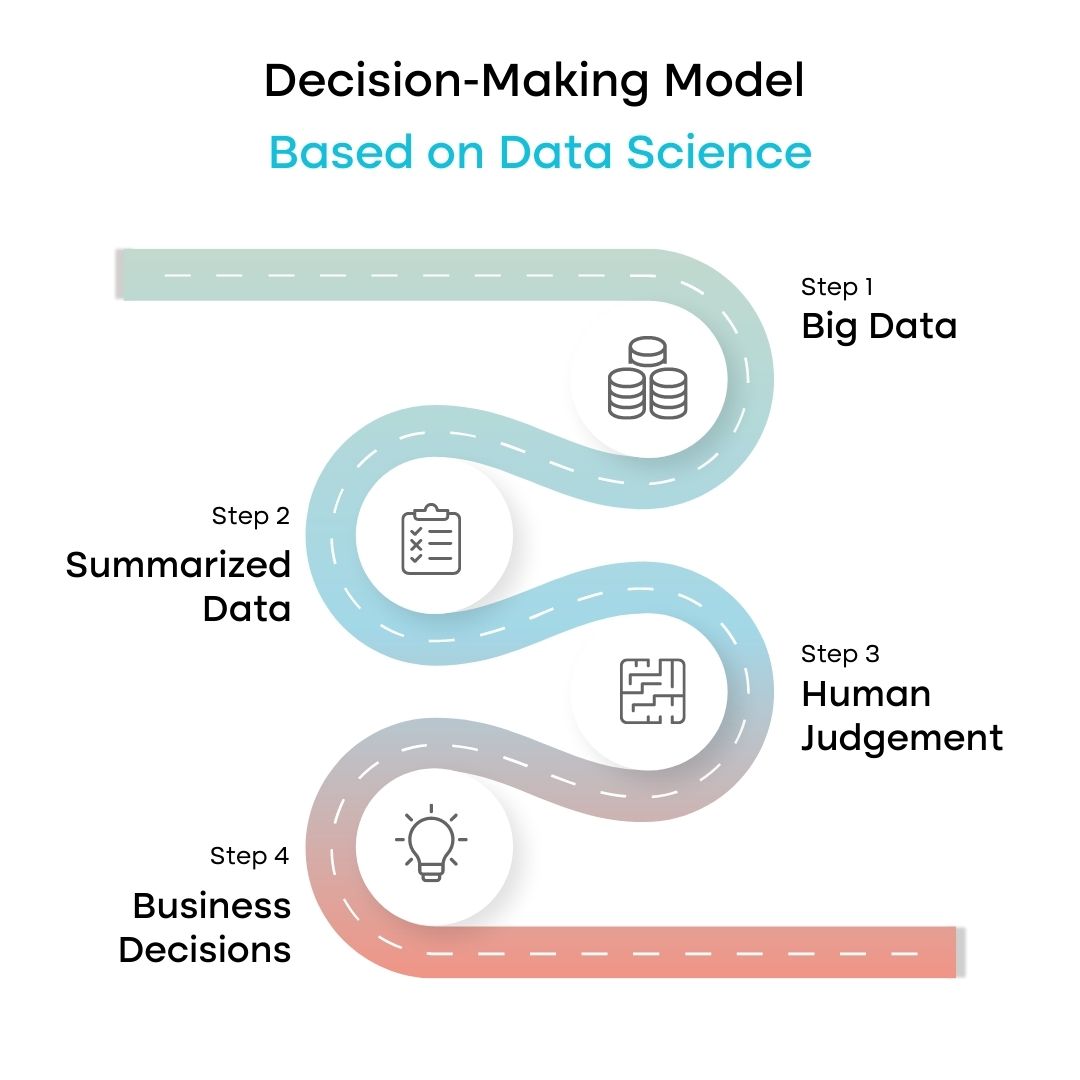
* **Ad creation**

Social media is used to collect info about customer preferences. Based on that info, targeted, and personalized advertisements are displayed on social media channels. Technologies like [**Hadoop programming**](https://imarticus.org/postgraduate-program-in-data-science-analytics/)**and Python programming** are also used by big data analysts in social media.

Young aspirants can go for the big data analytics programs launched by Imarticus Learning. Its [**PG Program in ML and Data Analytics**](https://imarticus.org/postgraduate-program-in-data-science-analytics/) can help working professionals in getting a raise. Start your big data course and learn Python and Hadoop programming!

## **The Importance of Big Data Analytics in Terms of Fraud Prevention:**

As online purchase, payment, and money transfer transactions increase, the risks of fraud that may occur through these transactions also increase. It was very difficult for companies to process and analyze the huge amount of data that emerged from these transactions and use it in fraud detection. At this point, we come across an indispensable facilitating tool: big data analytics for fraud detection. Using big data analytics in some points of fraud detection provides many advantages.



One of the most important points when detecting fraud is to take action quickly. It may take a long time to identify the suspicious ones among this large number of irregular data resulting from transactions.

Some transactions may be perceived as suspicious by misinterpretations as a result of these long analyses. During this evaluation process, there will still be a need for people, namely a manual workload, to analyze the data and check for suspicious transactions or misinterpretations.

To protect the company and customers from harm, it is necessary to draw up rules based on this data and look at past fraudulent activities, so that we can establish systems that can prevent possible damages and frauds that may occur.

All these mean more cost, time, and manual work. Big data analytics plays the biggest helping role in solving these issues. Using data analyzed with techniques in big data analytics can provide:

* Low costs
* More accurate and precise detections
* Optimized workflows and efficiency of systems
* Better services to customers

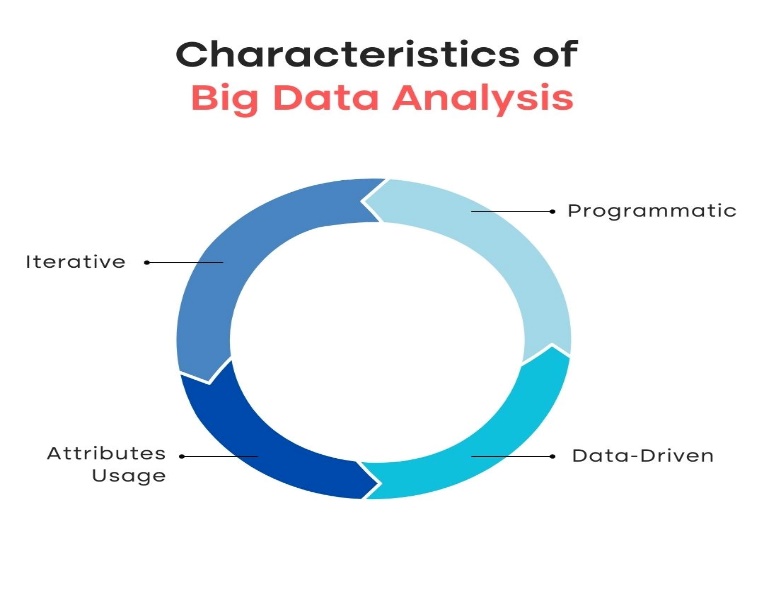


In addition, data mining and machine learning made by big data analytics are used in fraud analytics. These tools enable the implementation of payment fraud analytics, financial fraud analytics, and insurance fraud detection analytics.

## **What are the Common Problems in Big Data Analytics in Fraud Detection?**

We mentioned the importance of big data analytics in detecting fraud. Although it makes it easier to detect fraud, it can also bring some problems with it. Some of these problems can be listed as:

* **Unrelated or Insufficient Data:** The data from the transactions may come from many different sources. In some cases, false results can be obtained in fraud detectiondue to insufficient or irrelevant data. Detection can be based on the inappropriate rules used in the algorithm. Because of this risk of failure, companies may be hesitant to use big data analytics and machine learning.
* **High Costs:** Big data analytics and fraud detection systems may cause some costs such as the cost of software, and hardware systems, the cost of components used for the sustainability of these systems, and the time spent.
* **Dynamic Fraud Methods:**As technology develops, fraud methods develop at the same pace. In order to catch this speed and detect fraud, it is necessary to constantly monitor the data and give rules to the algorithms with new and accurate data analytics.
* **Data Security:** While processing the data and making decisions with this data analytics system, the security of the data is also a problem to be considered. That means the security of data should be checked.



## **Solutions to Big Data Analytics Problems**

* It is necessary to separate unnecessary data by processing complex data coming from many channels with certain analyses and big data analytics. This organized, prepared data is given to the algorithms. These algorithms ensure that fraudulent transactions are detected and quick action is taken.
* Monitoring access to this data, reports, and alarms from a single tool with easy and visualized dashboards prevents wasting money and time. Even if you pay for this tool in the first place, invest in it, it will provide much more benefit than what is paid to you in the long run by preventing fraudulent transactions detected with these tools.
* In conclusion, an engineering system should be established to analyze big data and manage and control its analytics. It is necessary to ensure data security by including cyber security experts. Most importantly, it will provide many benefits to use software such as **Formica**, which will provide features such as data processing, analysis, inference, and alarming in the field of fraud within the company and will prevent time and effort spent by helping analysts and engineers.

Fraudulent activities, including e-commerce scams, insurance fraud, cybersecurity threats, and financing fraud, pose significant risks to both individuals and companies across various industries such as retail, insurance, banking, and healthcare.

To combat these risks, businesses increasingly adopt advanced fraud prevention technologies and robust risk management strategies that depend on Big Data. For instance, predictive analytics models, alternative data sources, and advanced machine learning techniques empower decision-makers to develop innovative approaches and methodologies to proactively prevent fraud.

These technologies analyze large volumes of data to identify patterns and anomalies in transactions that indicate fraudulent behavior, allowing businesses to take proper action.

## **Understanding big data analytics**

Big data analytics involves processing and analyzing large and complex data sets, known as big data, to extract valuable insights. This field helps in the discovery of trends, patterns, and correlations within vast amounts of raw data, assisting analysts in making informed decisions.

By using the growing data generated from various sources, such as IoT sensors, social media, and even financial data from institutions, transactions, and smart devices, organizations can get actionable insights through advanced analytic techniques.

In response to current challenges, companies are shifting to advanced data analytics techniques for fraud prevention technologies and risk management strategies that use Big Data. Techniques like predictive analytics, alternative data, and machine learning are helping create new ways to prevent fraud.

## **Applications of big data analytics in fraud detection**

Here are a few applications of big data analytics in fraud detection:

### ****Real-time fraud monitoring****

One of the main benefits of using big data in fraud detection is the ability to perform real-time analytics and monitoring. Traditional methods of detecting fraud often depend on past data analysis, which may not be fast enough to stop advanced fraudsters.

Big data analytics supports instant analysis of transactions, user behavior, and patterns, allowing organizations to monitor, detect, and respond to potential fraud as it happens.

### ****Pattern recognition****

Integrating machine learning algorithms with big data analytics boosts insurance fraud detection analytics and prevention. These algorithms learn from historical data, identifying patterns and trends linked to fraudulent activities.

As they continuously evolve, machine learning models become highly effective at predicting and against fraudulent activity and preventing fraud before it happens, offering a proactive defense mechanism.

### ****Anomaly detection****

Big data allows advanced behavioral analytics, which involves analyzing user behavior patterns to identify anomalies. By specifying a baseline of normal user behavior, organizations can quickly detect deviations that may indicate fraud.

This approach to [payment fraud](https://hyperverge.co/blog/payment-fraud/) analytics is specifically effective in online banking, e-commerce, and other digital transactions, where abnormal patterns can be easily identified and investigated.

### ****Predictive modeling and risk assessment****

Predictive models can assist organizations in predicting fraud scenarios and identifying suspicious activities. These models can include variables such as transaction volume, velocity, or customer behavior patterns to evaluate the likelihood of fraud.

With these insights, organizations can consider the risk and allocate the resources to detect fraud more effectively and take proactive steps to prevent fraudulent activities before they occur.

### ****Examples of big data analytics for fraud detection and prevention****

To demonstrate further investigation into how big data analytics improves [credit card fraud](https://hyperverge.co/blog/card-not-present-fraud/) detection and prevention in banking and finance. Here are these real-world applications:

* **PayPal**: Using machine learning to analyze billions of transactions, PayPal detects potentially fraudulent transactions and activities in milliseconds, guiding significant savings and improved customer satisfaction.
* **Mastercard**: Using data mining to identify fraud patterns across millions of merchants and cardholders, Mastercard offers [fraud prevention solutions](https://hyperverge.co/in/use-cases/fraud-prevention-solutions/) like Mastercard Safety Net and Mastercard Identity Check.
* **HSBC**: By integrating and analyzing data from customer profiles, data records, transaction records, and external databases, HSBC combats money laundering and financial crime using techniques such as network analysis and commodity resolution.
* **American Express**: Using natural language generation and geospatial analysis, American Express analyzes customers’ spending habits, preferences, and locations to create personalized fraud alerts.

## **Challenges and considerations**

Here are some challenges to using big data for fraud detection.

### ****Data quality and integration issues****

Ensuring high-quality and reliable data involves techniques such as data cleaning, validation, and integration. These processes are important to make sure the data used for [financial fraud](https://hyperverge.co/blog/types-of-financial-frauds/) analytics and detection is accurate and trustworthy. Many organizations face integration issues while using big data analysis, for those who don’t have advanced high-quality computers. Consider using the latest computers to get along with these technologies.

### ****Privacy and security concerns****

Maintaining the privacy and security of data is necessary. This requires following regulations and ethical standards and implementing measures like encryption device data mine, anonymization, and access control to protect sensitive information. Find tools that will help you maintain your privacy and keep compliance with all rules and regulations.

### ****Regulatory compliance requirements****

Organizations must adhere to various regulatory requirements to maintain correct data handling. They must also keep themselves current on applicable laws and standards and build compliance measures into data management practices.

## **Benefits of big data analytics for fraud detection**

Here are some of the benefits of big data analysis for fraud detection:

### ****Improved accuracy and efficiency in fraud detection****

Big data analytics improves the accuracy and efficiency of fraud detection by processing vast amounts of data quickly. This capability of big data fraud detection allows organizations to mine call data records and identify fraudulent activities more accurately and efficiently.

### ****Reduced false positives and false negatives****

By using advanced analytic techniques, big data analytics reduces false positives for legitimate transactions and false negatives for fraudulent transactions. This guarantees that genuine transactions are not mistakenly flagged as fraud and that actual fraudulent activities are accurately detected.

### ****Faster response times and better decision-making****

The real-time processing capabilities of big data analytics allow faster response times to potential fraud. This quick data analysis also supports better decision-making, evolving fraud tactics, and allowing organizations to act accordingly to prevent fraud.

### ****Easier compliance with regulatory standards****

Big data analytics helps organizations comply with regulatory standards by providing robust data management and data security measures. It assures that data handling practices meet regulatory requirements, reducing the risk of non-compliance.

## **Best practices for big data analytics in fraud detection and prevention**

To maximize the benefits of fraud data analytics and address the challenges of big data analytics for fraud detection and prevention in banking and finance, consider the following best practices:

* **Define clear objectives and metrics:** Set targets people can work towards Your fraud data analytics will, therefore, be specific to helping achieve these goals, rather than general and perhaps less relevant.
* **Implement a robust data governance framework**: Develop a comprehensive data governance framework that includes policies, processes, roles, and responsibilities for maintaining data quality, privacy, security, and compliance. This framework should be flexible enough to adapt to evolving regulations and business needs.
* **Adopt an integrated approach to data management**: This approach uses various data sources, types, formats, and technologies, allowing for more comprehensive and accurate fraud detection capabilities.
* **Apply a combination of analytical methods**: Use a combination of testing and confirming analytical methods to find patterns that balance discovery and validation. This approach allows you to find new fraud patterns while confirming the validity of known patterns, balancing complexity with simplicity.
* **Implement a continuous improvement process**: Set an iterative process of testing, learning, and improving fraud detection lies. This continuous cycle helps you adapt to the ever-changing nature and dynamics of fraud, assuring your [fraud detection solutions](https://hyperverge.co/blog/fraud-detection-solutions/) remain effective.

By following these best practices, organizations can improve their ability to detect and prevent fraud, using Big Data and advanced analytics techniques to stay ahead of evolving threats.

## **Conclusion**

Throughout this article, we’ve explored the role of big data analytics in fraud detection and prevention. We’ve discussed how various techniques, such as real-time [fraud monitoring](https://hyperverge.co/blog/guide-fraud-monitoring/), pattern recognition, anomaly detection, and predictive modeling, can significantly improve the accuracy, efficiency, and effectiveness of fraud detection efforts.

Real-world examples from industry leaders like PayPal, Mastercard, HSBC, and American Express have shown the practical applications, potential threats, and benefits of using Big Data analytics tools in combating fraudulent activities.

The future of big data analytics in fraud detection and prevention looks promising, with continuous advancements in machine learning, AI, and data processing technologies. These developments will further refine and improve the capabilities of [fraud detection software](https://hyperverge.co/blog/financial-fraud-detection-software/), making them more adaptive and robust.

# How a Big Data Strategy Can Fight Insurance Fraud?

At the same time, insurers have also understood that they need a Big Data strategy for various purposes. Not all, however, already use tools to detect fraud.

The fight against banking and insurance crimes is a daily challenge for financial institutions around the world.

Fraud comes in many forms, from credit card scams to fake bank slips, data theft on fake websites, and irregular purchases. The fact is that during the pandemic, fraud has increased by 70%. The growth of fraud attempts has led banks and insurers to invest in anti-fraud technologies, but fraudsters are getting smarter.

For all institutions, the sophistication of this type of attack is a problem that needs to be solved efficiently.

Winning the war on fraud requires companies to outsmart criminals. The good news is that technology can help.

Thanks to Data Science, it’s now possible to improve fraud management in real-time, with more effective results and increased customer satisfaction. With data processing and analysis, Big Data, Artificial Intelligence, and Machine Learning, we can identify new attack patterns quickly.

Continue reading and understand how a Data Science strategy can help insurers avoid headaches and financial damage!

## **Fraud Fighting Challenges**

1. According to a survey, the most prominent challenges institutions face in the fight against fraud are directly linked to the digital transformation that the banking and insurance sector has undergone.
2. The increase in the use of digital channels during 2020 has expanded the scale of fraud. Right now, banks and startups are opening accounts through apps. Not installing mechanisms to combat fraud from account inception could put future operations at risk.
3. These problems are not unique to the industry but are of particular concern, as fraudsters heavily target them.
4. The two main components in fighting fraud are detection and prevention.
5. Fraud detection refers to the ability to detect fraudulent events, recognize patterns, and identify if fraud has occurred.
6. Prevention, which is much more complicated, seeks to analyze and predict fraudulent events before they occur.

**The most common moments where fraud occurs are:**

•Issuing a creditcard  
• Financing electronics  
• Buying a cell phone  
• Opening a bank account  
• Buying a car  
• Starting a business

The main concerns are related to:

**1. Data theft:** Institutions are more prone to crimes based on stolen identity, and customers are more prone to scams, as personal data is used to gain company/client trust.

**2. Faster Payment Processing:** Shortening the time it takes to process payments poses the challenge of real-time prevention, which requires well-protected systems and automation.

**3. Open banking:** Data accessibility requires robust security mechanisms – such as identity verification – interconnected between various institutions.

**4. Increase in digital channels:** Being present on multiple channels makes the fraud prevention strategy and policy more complex. Cohesion is needed.

**5. Social engineering:** Scams that customers are voluntarily coerced into, such as payments or transfers to fraudsters. They are notoriously difficult to detect.

Lack of protection also brings risks to institutions, which may become legally liable for customer losses.

Furthermore, a lack of security damages credibility and the operations as a whole. Investing in prevention solutions prevents losses and criminal liability, in addition to improving the institution’s image.

## **Fraud Fighting Strategies and Standards**

1. Prevention is the key to strategies against banking scams. But it’s not enough. It needs to have the tools to predict, detect, and respond to threats.
2. So we’re talking about a strategy that integrates data science across the institution, from tools to people, and from governance to culture.
3. Yes, technology is an excellent ally in ensuring that fraud monitoring is proactive rather than reactive. Allowing banking institutions to identify and anticipate fraudulent actions before they generate losses.

Imagine that the bank wants to start a relationship with a company or individual: **how do you prevent fraud?**

1. The first step is to carry out extensive research on the history of that institution or potential client, understanding their behavior.
2. The good news is that this process can be fully automated and executed quickly.
3. With a simple search on a sophisticated Big Data platform, it is possible to gather relevant information and make decisions based on that data.
4. The benefits of this data sweep are clear. From the data consolidated in a single report, prepared with combined criteria, managers can understand the consumer’s profile before closing the deal, validating the registration and identifying possible risk factors.
5. Another preventive measure available to companies is the definition of interest groups that are frequently monitored, springing to receive alerts in case of suspicious actions. There are also more advanced anti-fraud mechanisms, which we call enhanced intelligence.
6. In this case, an extra layer of technology is added to solutions to increase the power of analytics for decisions that come from data packs.
7. Personal documents offered as part of the validation process undergo rigorous verification procedures, including the use of facial recognition as proof of life.

## **Fraud Fighting Case Study**

Denmark’s largest bank has a great example of how Artificial Intelligence and Machine Learning can provide excellent results in fraud detection.

The institution adopted a set of technologies to create and launch a fraud detection platform based on Artificial Intelligence. The solution uses Machine Learning to analyze tens of thousands of resources, monitoring millions of banking transactions online in real-time to provide insight that differentiates honest activities from fraudulent ones.

The Danish bank’s anti-fraud program is the first to put Machine Learning techniques into production while also developing deep learning models to test out strategies.

The team began work within the bank’s existing infrastructure and then created advanced Machine Learning models to detect fraud in millions of transactions per year and at peak hours.

To ensure transparency and encourage trust, the mechanism includes an interpretation layer on top of the Machine Learning models, explaining blocked activity.

The fact is, every bank needs a scalable and robust analytics platform and a roadmap and digitization strategy to bring data science into the organization.

With so many online transactions, credit cards, and mobile payments, banks demand real-time solutions to detect fraud efficiently.

AI helps uncover data ‘anomalies’ through transaction analysis and identifies fraudulent operations through data and user behavior. Machine Learning contributes its predictive capacity thanks to current technological capabilities. Rapid machine learning ‘disarms’ criminals, preventing financial theft in real-time.

This entire process takes place in a matter of minutes, sometimes seconds.

Soon after that, new fraud patterns are developed. In other words, they are short windows of action and learning to be solved by ML/AI.

## **The Fraud Prevention Cycle: Continuous Improvement in Defense**

Data processing is at the heart of the project! Gathering, storing, structuring, and cross-checking information is the best way to detect fraud efficiently. The analysis of fraudulent behaviour is crucial to the definition of a propensity indicator.

This acts as an irregularity alert to interrupt the payment process and deepen the claim analysis. For this, it is essential that managers carefully look into the monitoring and detection of threats.

As this is a continuous cycle, actions must be constant, organized, and closely monitored.

The efficient work of fraud prevention depends on the team’s analytical capacity.

Below, we have a step-by-step guide for creating a fraud prevention cycle.

**DATA**

To identify patterns of fraudulent behaviour, companies need to process datasets – often unstructured ones.

From Data Science, it is possible to identify fraud-prone behaviours. After processing, this sea of ​​data is organized for visualization and understanding, bringing to life sets of information in dashboards.

**ANALYSIS**

This is where you find out if there is fraud being committed at that time and understand the path pursued by the scammers and their strategies. Here, the organization and dashboard visualization takes place only with the information necessary for the fraud-fighting processes.

Upon finding the pattern defined by the indicator, it performs a more accurate analysis in search of irregularities that prove fraud. Digital solutions can provide detailed, real-time information for diagnoses that lead to more informed decisions.

**CORRECTION**

The third stage of the fraud prevention cycle comes into play when the previously taken steps are insufficient to prevent fraudulent attacks.

In addition to reviewing the security techniques applied to prevent the recurrence of cases, it is essential to check the entire preventive process, accumulate lessons learned, and reinforce the need for policies to combat fraud in institutions.

## **Combating fraud deserves your attention**

The message for banks and insurance companies is: **invest in analytics and data technologies**.

Even within a sector continuously developing ‘State of the Art’ solutions to financial crimes, a more focused look is needed for data analysis and monitoring. Advanced analytics models facilitate this process through the use of detailed customer information.

The challenge is to do this work without compromising the quality of your customer experience, which is at the heart of the strategy. One that is increasingly demanding, with intuitive, responsive, and secure solutions.

Invest in combining solutions with multiple layers of defense. Not least because, as markets become more mature from a digital perspective, threats gain new levels of complexity. Fraud has been and will increasingly become a digital arms race.

**UNIT-II**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Introducing Technologies for Handling Big Data: Distributed and Parallel Computing for Big Data, Introducing Hadoop, Cloud Computing in Big Data. Understanding Hadoop Eco System: Hadoop Eco System, Hadoop Distributed File System, MapReduce, Hadoop Yarn, Hive, Pig, Sqoop, Zookeeper, Flume, Oozie.**  **Difference between Parallel Computing and Distributed Computing**  There are mainly two computation types, including **parallel computing** and **distributed computing.** A computer system may perform tasks according to human instructions. A single processor executes only one task in the computer system, which is not an effective way. Parallel computing solves this problem by allowing numerous processors to accomplish tasks simultaneously. Modern computers support parallel processing to improve system performance. In contrast, distributed computing enables several computers to communicate with one another and achieve a goal. All of these computers communicate and collaborate over the network. Distributed computing is commonly used by organizations such as **Facebook** and **Google** that allow people to share resources.  In this article, you will learn about the difference between **Parallel Computing** and **Distributed Computing.** But before discussing the differences, you must know about parallel computing and distributed computing.  **What is Parallel Computing?**  It is also known as **parallel processing.** It utilizes several processors. Each of the processors completes the tasks that have been allocated to them. In other words, parallel computing involves performing numerous tasks simultaneously. A shared memory or distributed memory system can be used to assist in parallel computing. All CPUs in shared memory systems share the memory. Memory is shared between the processors in distributed memory systems.  Parallel computing provides numerous advantages. Parallel computing helps to increase the CPU utilization and improve the performance because several processors work simultaneously. Moreover, the failure of one CPU has no impact on the other CPUs' functionality. Furthermore, if one processor needs instructions from another, the CPU might cause latency.  **Advantages and Disadvantages of Parallel Computing**  There are various advantages and disadvantages of parallel computing. Some of the advantages and disadvantages are as follows:  **Advantages**   1. It saves time and money because many resources working together cut down on time and costs. 2. It may be difficult to resolve larger problems on Serial Computing. 3. You can do many things at once using many computing resources. 4. Parallel computing is much better than serial computing for modeling, simulating, and comprehending complicated real-world events.   **Disadvantages**   1. The multi-core architectures consume a lot of power. 2. Parallel solutions are more difficult to implement, debug, and prove right due to the complexity of communication and coordination, and they frequently perform worse than their serial equivalents.   **What is Distributing Computing?**  It comprises several software components that reside on different systems but operate as a single system. A distributed system's computers can be physically close together and linked by a local network or geographically distant and linked by a **wide area network (WAN).** A distributed system can be made up of any number of different configurations, such as mainframes, PCs, workstations, and minicomputers. The main aim of distributed computing is to make a network work as a single computer.  There are various benefits of using distributed computing. It enables scalability and makes it simpler to share resources. It also aids in the efficiency of computation processes.  **Advantages and Disadvantages of Distributed Computing**  There are various advantages and disadvantages of distributed computing. Some of the advantages and disadvantages are as follows:  **Advantages**   1. It is flexible, making it simple to install, use, and debug new services. 2. In distributed computing, you may add multiple machines as required. 3. If the system crashes on one server, that doesn't affect other servers. 4. A distributed computer system may combine the computational capacity of several computers, making it faster than traditional systems.   **Disadvantages**   1. Data security and sharing are the main issues in distributed systems due to the features of open systems 2. Because of the distribution across multiple servers, troubleshooting and diagnostics are more challenging. 3. The main disadvantage of distributed computer systems is the lack of software support.   **Key differences between the Parallel Computing and Distributed Computing**  Here, you will learn the various key differences between parallel computing and distributed computation. Some of the key differences between parallel computing and distributed computing are as follows:   1. Parallel computing is a sort of computation in which various tasks or processes are run at the same time. In contrast, distributed computing is that type of computing in which the components are located on various networked systems that interact and coordinate their actions by passing messages to one another. 2. In parallel computing, processors communicate with another processor via a bus. On the other hand, computer systems in distributed computing connect with one another via a network. 3. Parallel computing takes place on a single computer. In contrast, distributed computing takes place on several computers. 4. Parallel computing aids in improving system performance. On the other hand, distributed computing allows for scalability, resource sharing, and the efficient completion of computation tasks. 5. The computer in parallel computing can have shared or distributed memory. In contrast, every system in distributed computing has its memory. 6. Multiple processors execute multiple tasks simultaneously in parallel computing. In contrast, many computer systems execute tasks simultaneously in distributed computing.   **Head-to-head Comparison between the Parallel Computing and Distributed Computing**   |  |  |  | | --- | --- | --- | | **Features** | **Parallel Computing** | **Distributed Computing** | | **Definition** | It is a type of computation in which various processes runs simultaneously. | It is that type of computing in which the components are located on various networked systems that interact and coordinate their actions by passing messages to one another. | | **Communication** | The processors communicate with one another via a bus. | The computer systems connect with one another via a network. | | **Functionality** | Several processors execute various tasks simultaneously in parallel computing. | Several computers execute tasks simultaneously. | | **Number of Computers** | It occurs in a single computer system. | It involves various computers. | | **Memory** | The system may have distributed or shared memory. | Each computer system in distributed computing has its own memory. | | **Usage** | It helps to improve the system performance | It allows for scalability, resource sharing, and the efficient completion of computation tasks. | |

# Role of Cloud Computing in Big Data Analytics.

In this day and age where information is everything, organizations are overwhelmed. This information, often called**“big data,”** refers to huge, complicated datasets that ordinary procedures cannot process. Businesses are increasingly turning to cloud computing in order to unlock the true value of big data and make use of it.

This article examines how cloud platforms can be used for storing vast amounts of data effectively as well as managing and analyzing such information.**It will reveal what exactly are some benefits brought by cloud computing into big-data analytics**, and discuss different services offered by providers among other things like considerations for adopting a cloud-based strategy towards big-data.

## **The Challenges of Big Data**

Big data poses several problems that impede traditional methods of analyzing data. These include:

1. **Volume**: The amount of data being created today is mind-bogglingly large. Regular storage systems do not have enough space to accommodate all these massive sets.
2. **Variety:**Big data comes in different forms such as**structured (relational databases)**,**unstructured (text files, pictures or videos from social media posts)**, and**semi-structured logs or emails.** Traditional tools struggle with this complexity.
3. **Velocity:** The speed at which new records are generated keeps rising every time; hence real-time analysis becomes difficult due to slow processing speed.
4. **Veracity:**If you want accurate findings from your research then you must ensure that your facts are correct since the garbage in garbage out rule applies here too. There is nothing worse than cleaning up after the traditional method has been used on a large dataset because it can take forever.

[**Cloud computing**](https://www.geeksforgeeks.org/cloud-computing/)**offers an effective solution towards dealing with big size information sets.**Organizations can store their big-data efficiently manage them as well analyze them by leveraging scalability provided through clouds on demand resources such as storage capacity . Here’s how:

* **Scalability: O**ne thing about these platforms is scalability; they provide large amounts storage when needed most without having to buy any hardware infrastructure in advance. For instance, if you know that there will be a lot of processing power required during certain periods then scaling up becomes very easy and quick.
* **Cost Effectiveness:** It also saves on costs since organizations only pay for what has been utilized unlike maintaining on-site infrastructure which may not be used all year round thus resulting into huge savings.
* **Performance:**Cloud computing offers high performance computing resources like servers with advanced networking features plus memory based in-memory capabilities which**enable faster data processing real-time analytics**
* **Accessibility:**geographical location should never hinder any business from getting value out of its information stores hence cloud-based solutions being accessible everywhere provided there’s internet connection. This encourages team work among members who are far apart geographically as well enables analysis to happen around the clock.
* **Security:**It is important that sensitive data is well guarded against unauthorized access, modification or loss hence cloud providers investing heavily in security measures such as **encryption, access control**and**residency options**for compliance purposes.

## **Cloud Services for Big Data Analytics**

### 1. Data Ingestion

* **Managed data pipelines:** These services automate the collection, transformation and loading of data from different sources into your cloud storage i.e., [**Apache Airflow**](https://www.geeksforgeeks.org/what-is-apache-airflow/)or [**AWS Glue**](https://www.geeksforgeeks.org/introduction-to-aws-glue-etl/) offered by various service providers.
* **Streaming ingestion:** Real time ingestion can be achieved using services like [**Apache Kafka**](https://www.geeksforgeeks.org/apache-kafka/) which allows integration with social media feeds among others

### 2. Data Storage

* **Object storage:** The best option for storing vast quantities of unstructured and semi-structured data are highly scalable and cost-effective object storage options such as[**Amazon S3**](https://www.geeksforgeeks.org/introduction-to-aws-simple-storage-service-aws-s3/)**,**[**Azure Blob Storage**](https://www.geeksforgeeks.org/azure-blob-storage/)**,**[**Google Cloud Storage**](https://www.geeksforgeeks.org/cloud-storage-in-google-cloud-platform-gcp/) among others.
* **Lakes of Data:**A cloud data lake serves as a centralized storage system that saves all of the data in its original format, giving users the opportunity to examine and analyze it at a later time. **Time is saved because of the flexible procedures that may be performed on the data.**
* **Data Warehouses:** When dealing with large datasets, **structured schemas are required for storage and analysis purposes**; this is exactly what a cloud data warehouse does. The method has made querying and reporting processes easier hence faster.

### 3. Data Processing and Transformation:

* **Managed Hadoop and Spark environments:**Complex infrastructure setup can be avoided by using pre-configured managed[**Hadoop clusters**](https://www.geeksforgeeks.org/basics-of-hadoop-cluster/)or**Spark clusters**provided by various cloud services.
* **Serverless information processing:**With serverless compute services like[**AWS Lambda**](https://www.geeksforgeeks.org/introduction-to-aws-lambda/) or [**Azure Functions**](https://www.geeksforgeeks.org/what-is-microsoft-azure-functions/), you can run data processing tasks without managing servers. This simplifies development and scaling.
* **Data anonymization and masking:**Cloud platforms provide tools and services to comply with privacy regulations by anonymizing or masking confidential datasets.

### 4. Data Analytics and Visualization:

* **Business intelligence (BI) tools:**Some cloud-based BI applications like [**Tableau**](https://www.geeksforgeeks.org/what-is-tableau-and-its-importance-in-data-visualization/)**,**[**Power BI**](https://www.geeksforgeeks.org/power-bi-tools-and-functionalities/)**, Looker**etc. provide interactive dashboards and reports for visual big data analysis.
* **Managed machine learning (ML) platforms** such as[**Google Cloud AI Platform**](https://www.geeksforgeeks.org/what-is-google-cloud-platform-gcp/)**,**[**Amazon SageMaker**](https://www.geeksforgeeks.org/what-is-sagemaker-in-aws/)**,**[**Azure Machine Learning**](https://www.geeksforgeeks.org/microsoft-azure-getting-started-with-azure-machine-learning-service/)etc., allow ML models development, testing, and deployment on massive datasets.
* **Predictive analytics and data mining:**Cloud platforms are equipped with built-in facilities both for [**predictive analytics**](https://www.geeksforgeeks.org/step-by-step-predictive-analysis-machine-learning/)and[**data mining**](https://www.geeksforgeeks.org/data-mining/) that can help you find patterns or trends in your data to assist you in future forecasting or better decision making.

## Benefits Beyond Core Analytics Services

* **Collaboration:**You can collaborate between a **data scientist/analyst/business** user since all your team members will have access through one centralized location where they can share insights with each other easily using; shared storage space or communication channels provided by these platforms themselves.
* **Disaster Recovery:** In case something unexpected happens such as power failure then rest assured because most cloud providers always ensure that there is minimum downtime experienced during any disaster recovery process thanks to their robustness in this area.
* **Innovation:**Organizations can take advantage of various cutting-edge technologies that are available through cloud platforms like Artificial Intelligence (AI) which will help them come up with new **data-driven solutions.**

## **Choosing the Right Cloud Platform for Big Data Analytics**

When choosing a [**cloud platform**](https://www.geeksforgeeks.org/cloud-computing-platforms-and-technologies/) for[**big data analytics**](https://www.geeksforgeeks.org/what-is-big-data-analytics/), there are several factors that need to be considered:

* **Scalability requirements:** Evaluate whether the platform can scale resources up or down as per your fluctuating needs in terms of processing power or storage space etc.
* **Security features:**Make sure the chosen provider has good security measures put in place especially when dealing with sensitive datasets so as not compromise privacy rights of individuals involved**directly/indirectly during analysis** process itself .
* **Cost considerations:**Compare pricing models offered by various providers against usage patterns based on current budgetary allocation then go ahead selecting most appropriate one among them all at hand.
* **Integration capabilities:**Check how well does it integrate with existing data infrastructure i.e., databases, warehouses etc., including [**ETL tools**](https://www.geeksforgeeks.org/etl-tools-overview/) like Informatica Power Center which might be already installed within organization environment thus avoiding compatibility issues arising later during implementation phase itself.
* **Vendor lock-in:**This is very crucial because you should always choose a platform that supports open standards thus providing flexibility needed incase one decides or wishes migrate from his/her current vendor/product line due change management related reasons where such may require significant investment both time wise as well financially too.

## **Security Considerations for Cloud-Based Big Data Analytics**

Security is always paramount when dealing with large volumes of information. Here are some key security considerations regarding cloud-based big-data analytics:

* **Data encryption:**Ensure all your stored files/data are encrypted; this helps safeguard against unauthorized access especially during transmission over unsecured networks where**they might get intercepted easily before reaching intended recipient(s).**
* **Access control:**Always make sure that only authorized personnel have access rights granted either individually or collectively towards particular dataset(s) held within a given storage location (s3 bucket etc.) so as not compromise security aspects involved during analysis phase itself.
* **Compliance regulations:** Confirm whether these cloud providers comply fully with relevant industry standards/regulations pertaining data protection act especially if dealing with health sector related information which should remain confidential throughout its lifecycle while being processed through various stages involved till final decision making moment reached upon by responsible parties concerned here.
* **Regular security audits:**Regularly conduct comprehensive security audits on your cloud environment to identify any potential vulnerability areas & address them accordingly before they can be exploited by malicious actors who might wish take advantage such weaknesses thereby causing harm intentionally against organization reputation or even financial loss too.
* **Data Copying and Restoration:** Keep an all-inclusive plan for data copying and restoration so that you could retrieve your files if a security breach occurs.

## Real-World Examples: Unveiling Insights Across Industries

Cloud-supported massive information analysis is changing the ways of working and decision-making in many companies. Here are a few interesting instances that demonstrate such technology’s capabilities:

### 1. Retail Industry: The Power of Personalization

Think about a retail environment where product recommendations seem uncannily accurate and marketing campaigns speak to your soul. This is made possible by cloud-based big data analytics. **Retailers use these tools to process immense volumes of customer information, such as purchase history, browsing habits and social media sentiment.** They then apply this knowledge to:

* **Customize marketing campaigns:**Higher conversion rates and increased customer satisfaction are achieved through targeted email blasts and social media ads that cater for individual preferences.
* **Optimize product recommendations:** Recommender systems driven by big data analytics propose products customers are likely to find interesting thereby increasing sales and reducing cart abandonment rates.
* **Enhance inventory management:**Retailers can optimize their inventory levels by scrutinizing sales trends alongside customer demand patterns which eliminates stockouts while minimizing clearance sales.

### 2. Healthcare: From Diagnosis to Personalized Care

The healthcare industry has rapidly adopted cloud-based big data analytics for better patient care and operational efficiency. Here’s how:

* **Improved diagnosis:**Healthcare providers can now diagnose patients faster and more accurately by analyzing medical records together with imaging scans besides wearable device sensor data.
* **Individual treatment plans:** Big data analytics makes it possible to create individualized treatment plans through identification of factors affecting response to certain drugs or therapies.
* **Predictive prevention care:**Through cloud based analytics it is possible to identify people at high risk of particular illnesses before they actually occur thus leading to better outcomes for patients and lower healthcare expenses.

### 3. Financial Services: Risk Management & Fraud Detection

Effectively managing risks and making informed decisions are crucial in the ever changing banking industry. Here’s how financial companies can use big data analytics in the cloud:

* **Identify fraudulent activity:**By using advanced algorithms to make sense of real-time transaction patterns, banks are able to detect and prevent fraudulent transactions from taking place, thereby protecting both themselves and customers.
* **Evaluate credit riskiness:** By checking borrowers’ financial histories against other types of relevant data points, lenders can make better choices concerning approvals on loans and interest rates hence reducing credit risk.
* **Develop cutting-edge financial products:**Banks can use big data analytics to craft unique financial products for different market segments as they continue studying their clients’ desires and preferences.

## **The Future of Cloud Computing and Big Data Analytics**

The future of big data analysis is directly related to that of cloud computing. The significance of cloud platforms will only increase as enterprises grapple with information overload and seek deeper insights. The following are some tendencies to watch out for:

* **Hybrid and Multi-Cloud Environments:**As per their unique needs, companies will use more and more [**Hybrid and Multi Cloud**](https://www.geeksforgeeks.org/difference-between-multi-cloud-and-hybrid-cloud/) approaches to take advantage of the specific capabilities typical for different providers.
* **Serverless Computing:** Businesses will increasingly adopt serverless computing due to its liberation of administrators from the management of underlying infrastructure to concentrate on analytics functions.
* **Integration Of AI & ML:**Cloud platforms will seamlessly integrate[**artificial intelligence (AI)**](https://www.geeksforgeeks.org/artificial-intelligence-an-introduction/) alongside [**machine learning (ML)**](https://www.geeksforgeeks.org/machine-learning/)functionalities thus enabling advanced analytics as well as automated decision making.
* **Emphasis on Data Governance and Privacy:**To keep pace with shifting rules on data security and privacy, businesses will need more advanced means of governing their information, which cloud providers can supply.

**Introduction to Hadoop Distributed File System(HDFS)**

With growing data velocity the data size easily outgrows the storage limit of a machine. A solution would be to store the data across a network of machines. Such filesystems are called distributed filesystems. Since data is stored across a network all the complications of a network come in.   
This is where Hadoop comes in. It provides one of the most reliable filesystems. HDFS (Hadoop Distributed File System) is a unique design that provides storage for extremely large files with streaming data access pattern and it runs on commodity hardware. Let’s elaborate the terms:

* **Extremely large files**: Here we are talking about the data in range of petabytes (1000 TB).
* **Streaming Data Access Pattern**: HDFS is designed on principle of write-once and read-many-times. Once data is written large portions of dataset can be processed any number times.
* **Commodity hardware:** Hardware that is inexpensive and easily available in the market. This is one of feature which specially distinguishes HDFS from other file system.

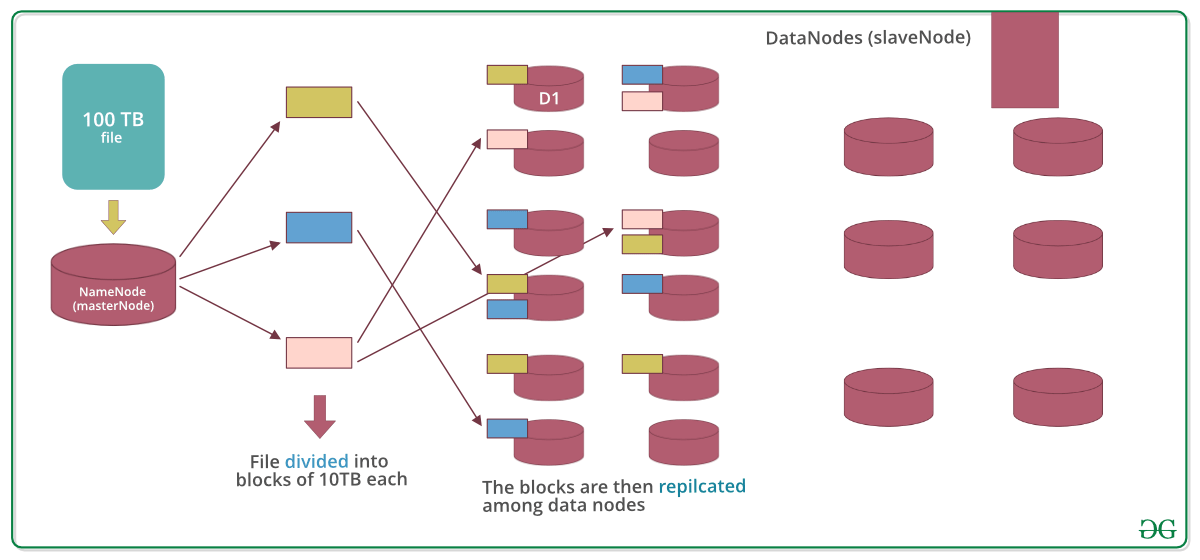
**Nodes:** Master-slave nodes typically form the HDFS cluster.

1. **NameNode(MasterNode):**
   * Manages all the slave nodes and assign work to them.
   * It executes filesystem namespace operations like opening, closing, renaming files and directories.
   * It should be deployed on reliable hardware which has the high config. not on commodity hardware.
2. **DataNode(SlaveNode):**
   * Actual worker nodes, who do the actual work like reading, writing, processing etc.
   * They also perform creation, deletion, and replication upon instruction from the master.
   * They can be deployed on commodity hardware.

**HDFS daemons:** Daemons are the processes running in background.

* **Namenodes:**
  + Run on the master node.
  + Store metadata (data about data) like file path, the number of blocks, block Ids. etc.
  + Require high amount of RAM.
  + Store meta-data in RAM for fast retrieval i.e to reduce seek time. Though a persistent copy of it is kept on disk.
* **DataNodes:**
  + Run on slave nodes.
  + Require high memory as data is actually stored here.

**Data storage in HDFS:** Now let’s see how the data is stored in a distributed manner.



Let’s assume that 100TB file is inserted, then masternode(namenode) will first divide the file into blocks of 10TB (default size is 128 MB in Hadoop 2.x and above). Then these blocks are stored across different datanodes(slavenode). Datanodes(slavenode)replicate the blocks among themselves and the information of what blocks they contain is sent to the master. Default replication factor is 3 means for each block 3 replicas are created (including itself). In hdfs.site.xml we can increase or decrease the replication factor i.e we can edit its configuration here.

**Why divide the file into blocks?**

Answer: Let’s assume that we don’t divide, now it’s very difficult to store a 100 TB file on a single machine. Even if we store, then each read and write operation on that whole file is going to take very high seek time. But if we have multiple blocks of size 128MB then its become easy to perform various read and write operations on it compared to doing it on a whole file at once. So we divide the file to have faster data access i.e. reduce seek time.

**Why replicate the blocks in data nodes while storing?**

Answer: Let’s assume we don’t replicate and only one yellow block is present on datanode D1. Now if the data node D1 crashes we will lose the block and which will make the overall data inconsistent and faulty. So we replicate the blocks to achieve fault-tolerance.

**Terms related to HDFS:**

* **HeartBeat** : It is the signal that datanode continuously sends to namenode. If namenode doesn’t receive heartbeat from a datanode then it will consider it dead.
* **Balancing** : If a datanode is crashed the blocks present on it will be gone too and the blocks will be under-replicated compared to the remaining blocks. Here master node(namenode) will give a signal to datanodes containing replicas of those lost blocks to replicate so that overall distribution of blocks is balanced.
* **Replication:**: It is done by datanode.

**Note:** No two replicas of the same block are present on the same datanode.

**Features:**

* Distributed data storage.
* Blocks reduce seek time.
* The data is highly available as the same block is present at multiple datanodes.
* Even if multiple datanodes are down we can still do our work, thus making it highly reliable.
* High fault tolerance.

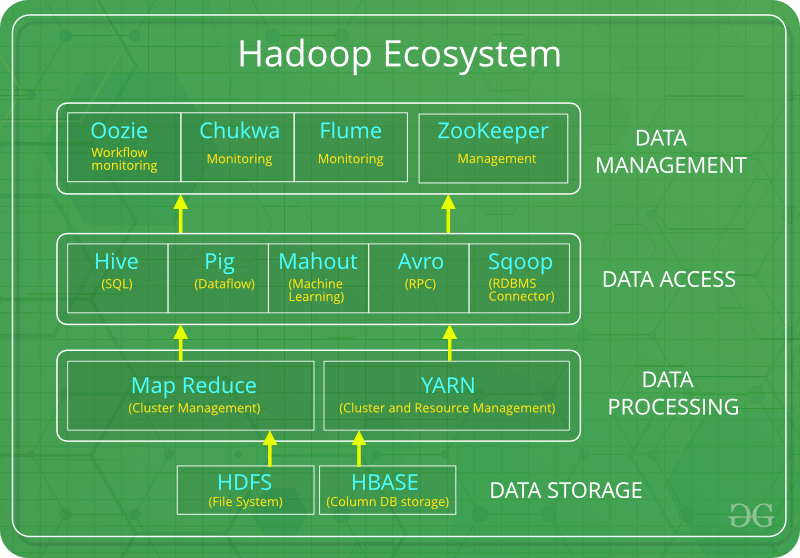
**Limitations:** Though HDFS provide many features there are some areas where it doesn’t work well.

* **Low latency data access**: Applications that require low-latency access to data i.e in the range of milliseconds will not work well with HDFS, because HDFS is designed keeping in mind that we need high-throughput of data even at the cost of latency.
* **Small file problem**: Having lots of small files will result in lots of seeks and lots of movement from one datanode to another datanode to retrieve each small file, this whole process is a very inefficient data access pattern.

**HADOOP ECOSYSTEM**

**Overview:** Apache Hadoop is an open source framework intended to make interaction with [**big data**](https://www.geeksforgeeks.org/what-is-big-data/) easier, However, for those who are not acquainted with this technology, one question arises that what is big data ? Big data is a term given to the data sets which can’t be processed in an efficient manner with the help of traditional methodology such as RDBMS. Hadoop has made its place in the industries and companies that need to work on large data sets which are sensitive and needs efficient handling. Hadoop is a framework that enables processing of large data sets which reside in the form of clusters. Being a framework, Hadoop is made up of several modules that are supported by a large ecosystem of technologies.   
**Introduction:** Hadoop Ecosystem is a platform or a suite which provides various services to solve the big data problems. It includes Apache projects and various commercial tools and solutions. There are four major elements of Hadoop i.e. **HDFS, MapReduce, YARN, and Hadoop Common Utilities**. Most of the tools or solutions are used to supplement or support these major elements. All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.  
Following are the components that collectively form a Hadoop ecosystem:

* **HDFS:**Hadoop Distributed File System
* **YARN:** Yet Another Resource Negotiator
* **MapReduce:** Programming based Data Processing
* **Spark:** In-Memory data processing
* **PIG, HIVE:** Query based processing of data services
* **HBase:**NoSQL Database
* **Mahout, Spark MLLib:** [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)algorithm libraries
* **Solar, Lucene:** Searching and Indexing
* **Zookeeper:** Managing cluster
* **Oozie:** Job Scheduling



**HDFS:**

* HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.
* HDFS consists of two core components i.e.
  1. Name node
  2. Data Node
* Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores the actual data. These data nodes are commodity hardware in the distributed environment. Undoubtedly, making Hadoop cost effective.
* HDFS maintains all the coordination between the clusters and hardware, thus working at the heart of the system.

**YARN:**

* Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters. In short, it performs scheduling and resource allocation for the Hadoop System.
* Consists of three major components i.e.
  1. Resource Manager
  2. Nodes Manager
  3. Application Manager
* Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager. Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

**MapReduce:**

* By making the use of distributed and parallel algorithms, MapReduce makes it possible to carry over the processing’s logic and helps to write applications which transform big data sets into a manageable one.
* MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:
  1. Map() performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.
  2. Reduce(), as the name suggests does the summarization by aggregating the mapped data. In simple, Reduce() takes the output generated by Map() as input and combines those tuples into smaller set of tuples.

**PIG:**  
Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.

* It is a platform for structuring the data flow, processing and analyzing huge data sets.
* Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS.
* Pig Latin language is specially designed for this framework which runs on Pig Runtime. Just the way Java runs on the [JVM](https://www.geeksforgeeks.org/jvm-works-jvm-architecture/).
* Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

**HIVE:**

* With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).
* It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.
* Similar to the Query Processing frameworks, HIVE too comes with two components: JDBC Drivers and HIVE Command Line.
* JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

**Mahout:**

* Mahout, allows Machine Learnability to a system or application. [Machine Learning](https://www.geeksforgeeks.org/ml-machine-learning/), as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction or on the basis of algorithms.
* It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning. It allows invoking algorithms as per our need with the help of its own libraries.

**Apache Spark:**

* It’s a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.
* It consumes in memory resources hence, thus being faster than the prior in terms of optimization.
* Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably.

**Apache HBase:**

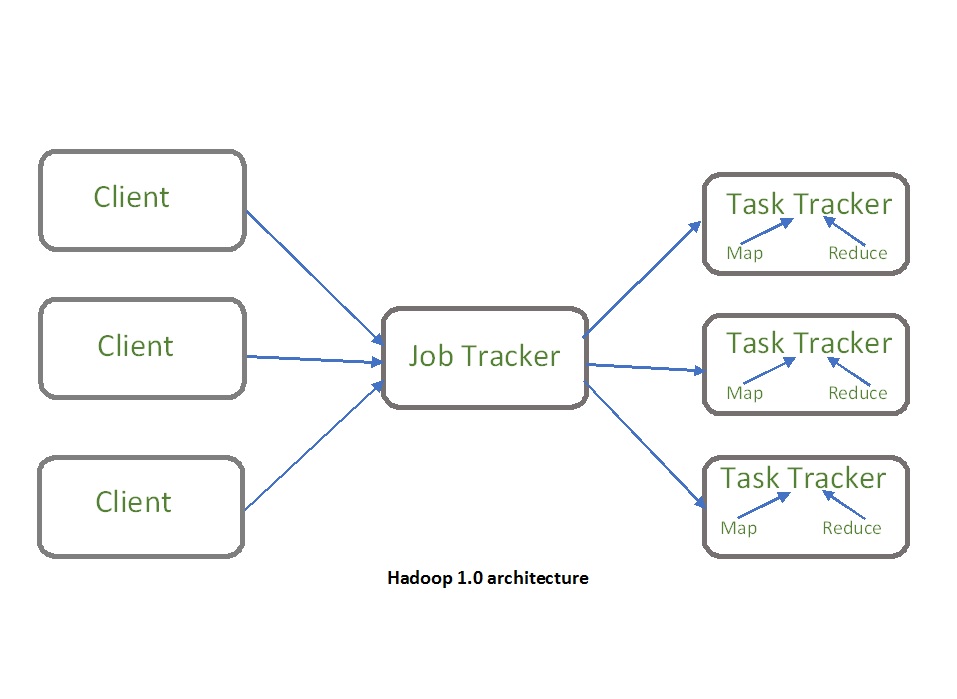
* It’s a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of Google’s BigTable, thus able to work on Big Data sets effectively.
* At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data

**Other Components:** Apart from all of these, there are some other components too that carry out a huge task in order to make Hadoop capable of processing large datasets. They are as follows:

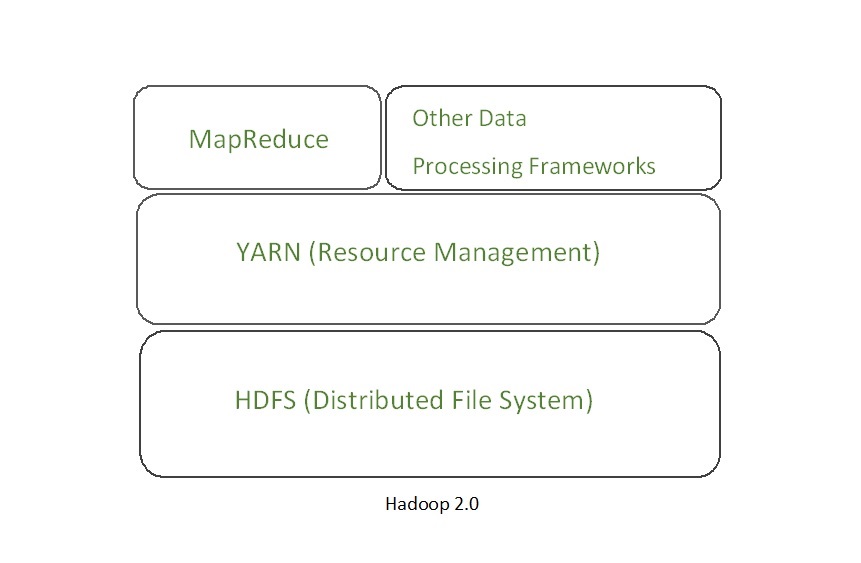
* **Solr, Lucene:**These are the two services that perform the task of searching and indexing with the help of some java libraries, especially Lucene is based on Java which allows spell check mechanism, as well. However, Lucene is driven by Solr.
* **Zookeeper:**There was a huge issue of management of coordination and synchronization among the resources or the components of Hadoop which resulted in inconsistency, often. Zookeeper overcame all the problems by performing synchronization, inter-component based communication, grouping, and maintenance.
* **Oozie:**Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs .i.e Oozie workflow and Oozie coordinator jobs. Oozie workflow is the jobs that need to be executed in a sequentially ordered manner whereas Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

**Hadoop YARN Architecture**

YARN stands for “**Yet Another Resource Negotiator**“. It was introduced in Hadoop 2.0 to remove the bottleneck on Job Tracker which was present in Hadoop 1.0. YARN was described as a “Redesigned Resource Manager” at the time of its launching, but it has now evolved to be known as large-scale distributed operating system used for Big Data processing.



YARN architecture basically separates resource management layer from the processing layer. In Hadoop 1.0 version, the responsibility of Job tracker is split between the resource manager and application manager.



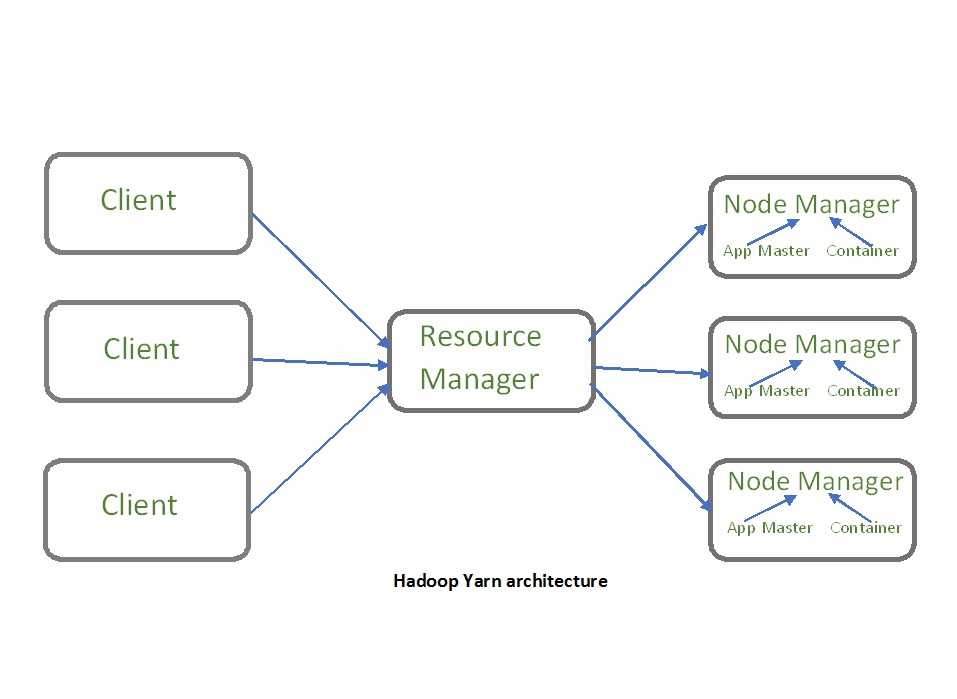
YARN also allows different data processing engines like graph processing, interactive processing, stream processing as well as batch processing to run and process data stored in HDFS (Hadoop Distributed File System) thus making the system much more efficient. Through its various components, it can dynamically allocate various resources and schedule the application processing. For large volume data processing, it is quite necessary to manage the available resources properly so that every application can leverage them.

**YARN Features:**

 YARN gained popularity because of the following features- 

* **Scalability:** The scheduler in Resource manager of YARN architecture allows Hadoop to extend and manage thousands of nodes and clusters.
* **Compatibility:** YARN supports the existing map-reduce applications without disruptions thus making it compatible with Hadoop 1.0 as well.
* **Cluster Utilization:**Since YARN supports Dynamic utilization of cluster in Hadoop, which enables optimized Cluster Utilization.
* **Multi-tenancy:** It allows multiple engine access thus giving organizations a benefit of multi-tenancy.

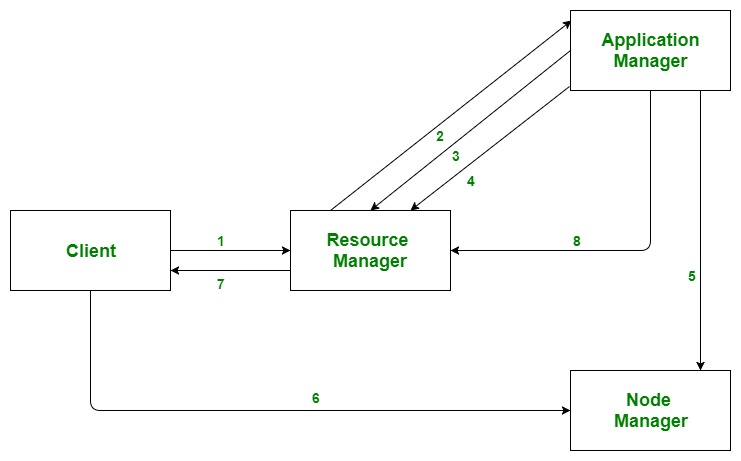
**Hadoop YARN Architecture**



The main components of YARN architecture include:

* **Client:** It submits map-reduce jobs.
* **Resource Manager:** It is the master daemon of YARN and is responsible for resource assignment and management among all the applications. Whenever it receives a processing request, it forwards it to the corresponding node manager and allocates resources for the completion of the request accordingly. It has two major components:
  + **Scheduler:** It performs scheduling based on the allocated application and available resources. It is a pure scheduler, means it does not perform other tasks such as monitoring or tracking and does not guarantee a restart if a task fails. The YARN scheduler supports plugins such as Capacity Scheduler and Fair Scheduler to partition the cluster resources.
  + **Application manager:** It is responsible for accepting the application and negotiating the first container from the resource manager. It also restarts the Application Master container if a task fails.
* **Node Manager:** It take care of individual node on Hadoop cluster and manages application and workflow and that particular node. Its primary job is to keep-up with the Resource Manager. It registers with the Resource Manager and sends heartbeats with the health status of the node. It monitors resource usage, performs log management and also kills a container based on directions from the resource manager. It is also responsible for creating the container process and start it on the request of Application master.
* **Application Master:**An application is a single job submitted to a framework. The application master is responsible for negotiating resources with the resource manager, tracking the status and monitoring progress of a single application. The application master requests the container from the node manager by sending a Container Launch Context(CLC) which includes everything an application needs to run. Once the application is started, it sends the health report to the resource manager from time-to-time.
* **Container:** It is a collection of physical resources such as RAM, CPU cores and disk on a single node. The containers are invoked by Container Launch Context(CLC) which is a record that contains information such as environment variables, security tokens, dependencies etc.

**Application workflow in Hadoop YARN:**



1. Client submits an application
2. The Resource Manager allocates a container to start the Application Manager
3. The Application Manager registers itself with the Resource Manager
4. The Application Manager negotiates containers from the Resource Manager
5. The Application Manager notifies the Node Manager to launch containers
6. Application code is executed in the container
7. Client contacts Resource Manager/Application Manager to monitor application’s status
8. Once the processing is complete, the Application Manager un-registers with the Resource Manager

**Advantages :**

* **Flexibility:** YARN offers flexibility to run various types of distributed processing systems such as Apache Spark, Apache Flink, Apache Storm, and others. It allows multiple processing engines to run simultaneously on a single Hadoop cluster.
* **Resource Management:** YARN provides an efficient way of managing resources in the Hadoop cluster. It allows administrators to allocate and monitor the resources required by each application in a cluster, such as CPU, memory, and disk space.
* **Scalability:** YARN is designed to be highly scalable and can handle thousands of nodes in a cluster. It can scale up or down based on the requirements of the applications running on the cluster.
* **Improved Performance:**YARN offers better performance by providing a centralized resource management system. It ensures that the resources are optimally utilized, and applications are efficiently scheduled on the available resources.
* **Security:** YARN provides robust security features such as Kerberos authentication, Secure Shell (SSH) access, and secure data transmission. It ensures that the data stored and processed on the Hadoop cluster is secure.

**Disadvantages :**

* **Complexity:**YARN adds complexity to the Hadoop ecosystem. It requires additional configurations and settings, which can be difficult for users who are not familiar with YARN.
* **Overhead:** YARN introduces additional overhead, which can slow down the performance of the Hadoop cluster. This overhead is required for managing resources and scheduling applications.
* **Latency:**YARN introduces additional latency in the Hadoop ecosystem. This latency can be caused by resource allocation, application scheduling, and communication between components.
* **Single Point of Failure:** YARN can be a single point of failure in the Hadoop cluster. If YARN fails, it can cause the entire cluster to go down. To avoid this, administrators need to set up a backup YARN instance for high availability.
* **Limited Support:** YARN has limited support for non-Java programming languages. Although it supports multiple processing engines, some engines have limited language support, which can limit the usability of YARN in certain environments.

**Apache Hive**

* **Apache Hive** is a data warehouse and an ETL(ETL stands for "extract, transform, and load". It's a process that combines data from multiple sources into a single repository, such as a data warehouse, data store, or data lake) tool which provides an SQL-like interface between the user and the Hadoop distributed file system (HDFS) which integrates Hadoop.
* It is built on top of Hadoop. It is a software project that provides data query and analysis. It facilitates reading, writing and handling wide datasets that stored in distributed storage and queried by Structure Query Language (SQL) syntax.
* It is not built for Online Transactional Processing (OLTP) workloads. It is frequently used for data warehousing tasks like data encapsulation, Ad-hoc Queries, and analysis of huge datasets. It is designed to enhance scalability, extensibility, performance, fault-tolerance and loose-coupling with its input formats.
* Initially Hive is developed by Facebook and Amazon, Netflix and It delivers standard SQL functionality for analytics.
* Traditional SQL queries are written in the MapReduce Java API to execute SQL Application and SQL queries over distributed data.
* Hive provides portability as most data warehousing applications functions with SQL-based query languages like NoSQL.
* Apache Hive is a data warehouse software project that is built on top of the Hadoop ecosystem. It provides an SQL-like interface to query and analyze large datasets stored in Hadoop’s distributed file system (HDFS) or other compatible storage systems.
* Hive uses a language called HiveQL, which is similar to SQL, to allow users to express data queries, transformations, and analyses in a familiar syntax. HiveQL statements are compiled into MapReduce jobs, which are then executed on the Hadoop cluster to process the data.
* Hive includes many features that make it a useful tool for big data analysis, including support for partitioning, indexing, and user-defined functions (UDFs). It also provides a number of optimization techniques to improve query performance, such as predicate pushdown, column pruning, and query parallelization.
* Hive can be used for a variety of data processing tasks, such as data warehousing, ETL (extract, transform, load) pipelines, and ad-hoc data analysis. It is widely used in the big data industry, especially in companies that have adopted the Hadoop ecosystem as their primary data processing platform.

**Components of Hive:**

1. **HCatalog –**   
   It is a Hive component and is a table as well as a store management layer for Hadoop. It enables user along with various data processing tools like Pig and MapReduce which enables to read and write on the grid easily.
2. **WebHCat –**   
   It provides a service which can be utilized by the user to run Hadoop MapReduce (or YARN), Pig, Hive tasks or function Hive metadata operations with an HTTP interface.

**Modes of Hive:**

1. **LocalMode–**   
   It is used, when the Hadoop is built under pseudo mode which has only one data node, when the data size is smaller in term of restricted to single local machine, and when processing will be faster on smaller datasets existing in the local machine.
2. **MapReduceMode–**   
   It is used, when Hadoop is built with multiple data nodes and data is divided across various nodes, it will function on huge datasets and query is executed parallelly, and to achieve enhanced performance in processing large datasets.

**Characteristics of Hive:**

1. Databases and tables are built before loading the data.
2. Hive as data warehouse is built to manage and query only structured data which is residing under tables.
3. At the time of handling structured data, MapReduce lacks optimization and usability function such as UDFs whereas Hive framework have optimization and usability.
4. Programming in Hadoop deals directly with the files. So, Hive can partition the data with directory structures to improve performance on certain queries.
5. Hive is compatible for the various file formats which are TEXTFILE, SEQUENCEFILE, ORC, RCFILE, etc.
6. Hive uses derby database in single user metadata storage and it uses MYSQL for multiple user Metadata or shared Metadata.

**Features of Hive:**

1. It provides indexes, including bitmap indexes to accelerate the queries. Index type containing compaction and bitmap index as of 0.10.
2. Metadata storage in a RDBMS, reduces the time to function semantic checks during query execution.
3. Built in user-defined functions (UDFs) to manipulation of strings, dates, and other data-mining tools. Hive is reinforced to extend the UDF set to deal with the use-cases not reinforced by predefined functions.
4. DEFLATE, BWT, snappy, etc are the algorithms to operation on compressed data which is stored in Hadoop Ecosystem.
5. It stores schemas in a database and processes the data into the Hadoop File Distributed File System (HDFS).
6. It is built for Online Analytical Processing (OLAP).
7. It delivers various types of querying language which are frequently known as Hive Query Language (HVL or HiveQL).

**Advantages:**

**Scalability:** Apache Hive is designed to handle large volumes of data, making it a scalable solution for bigdatap rocessing.

**Familiar SQL-like interface:**Hive uses a SQL-like language called HiveQL, which makes it easy for SQL users to learn and.

**Integration with Hadoop ecosystem:** Hive integrates well with the Hadoop ecosystem, enabling users to process data using other Hadoop tools like Pig, MapReduce, and Spark.  
**Supports partitioning and bucketing:**Hive supports partitioning and bucketing, which can improve query performance by limiting the amount of data scanned.

**User-defined unctions:** Hive allows users to define their own functions, which can be used in HiveQL queries.

**Disadvantages:**

**Limited real-time processing:** Hive is designed for batch processing, which means it may not be the best tool for real-time data processing.

**Slow performance:**Hive can be slower than traditional relational databases because it is built on top of Hadoop, which is optimized for batch processing rather than interactive querying.  
**Steep learning curve:**While Hive uses a SQL-like language, it still requires users to have knowledge of Hadoop and distributed computing, which can make it difficult for beginners to use.

**Limited flexibility:** Hive is not as flexible as other data warehousing tools because it is designed to work specifically with Hadoop, which can limit its usability in other environments.

**Introduction to Apache Pig:**

* Pig Represents Big Data as data flows. Pig is a high-level platform or tool which is used to process the large datasets.
* It provides a high-level of abstraction for processing over the MapReduce. It provides a high-level scripting language, known as Pig Latin which is used to develop the data analysis codes. First, to process the data which is stored in the HDFS, the programmers will write the scripts using the Pig Latin Language.
* Internally Pig Engine(a component of Apache Pig) converted all these scripts into a specific map and reduce task. But these are not visible to the programmers in order to provide a high-level of abstraction.
* Pig Latin and Pig Engine are the two main components of the Apache Pig tool. The result of Pig always stored in the HDFS.

**Need of Pig:**

1. One limitation of MapReduce is that the development cycle is very long.
2. Writing the reducer and mapper, compiling packaging the code, submitting the job and retrieving the output is a time-consuming task.
3. Apache Pig reduces the time of development using the multi-query approach. Also, Pig is beneficial for programmers who are not from [Java](https://www.geeksforgeeks.org/java/)background.
4. 200 lines of Java code can be written in only 10 lines using the Pig Latin language. Programmers who have SQL knowledge needed less effort to learn Pig Latin.

* It uses query approach which results in reducing the length of the code.
* Pig Latin is SQL like language.
* It provides many builtIn operators.
* It provides nested data types (tuples, bags, map).

**Evolution of Pig:**Earlier in 2006, Apache Pig was developed by Yahoo’s researchers. At that time, the main idea to develop Pig was to execute the MapReduce jobs on extremely large datasets. In the year 2007, it moved to Apache Software Foundation(ASF) which makes it an open source project. The first version(0.1) of Pig came in the year 2008. The latest version of Apache Pig is 0.18 which came in the year 2017.

**Features of Apache Pig:**

* For performing several operations Apache Pig provides rich sets of operators like the filtering, joining, sorting, aggregation etc.
* Easy to learn, read and write. Especially for SQL-programmer, Apache Pig is a boon.
* Apache Pig is extensible so that you can make your own process and  user-defined functions(UDFs) written in python, java or other programming languages .
* Join operation is easy in Apache Pig.
* Fewer lines of code.
* Apache Pig allows splits in the pipeline.
* By integrating with other components of the Apache Hadoop ecosystem, such as Apache Hive, Apache Spark, and Apache ZooKeeper, Apache Pig enables users to take advantage of these components’ capabilities while transforming data.
* The data structure is multivalued, nested, and richer.
* Pig can handle the analysis of both structured and unstructured data.

**Difference between Pig and MapReduce**

| **Apache Pig** | **MapReduce** |
| --- | --- |
| It is a scripting language. | It is a compiled programming language. |
| Abstraction is at higher level. | Abstraction is at lower level. |
| It have less line of code as compared to MapReduce. | Lines of code is more. |
| Less effort is needed for Apache Pig. | More development efforts are required for MapReduce. |
| Code efficiency is less as compared to MapReduce. | As compared to Pig efficiency of code is higher. |
| Pig provides built in functions for ordering, sorting and union. | Hard to perform data operations. |
| It allows nested data types like map, tuple and bag | It does not allow nested data types |

**Applications of Apache Pig:**

* For exploring large datasets Pig Scripting is used.
* Provides the supports across large data-sets for Ad-hoc queries.
* In the prototyping of large data-sets processing algorithms.
* Required to process the time sensitive data loads.
* For collecting large amounts of datasets in form of search logs and web crawls.
* Used where the analytical insights are needed using the sampling.

**Types of Data Models in Apache Pig:** It consist of the 4 types of data models as follows:

* **Atom**: It is a atomic data value which is used to store as a string. The main use of this model is that it can be used as a number and as well as a string.
* **Tuple**: It is an ordered set of the fields.
* **Bag**: It is a collection of the tuples.
* **Map**: It is a set of key/value pairs.

**Overview of SQOOP in Hadoop**

**SQOOP:**

Previously when there was no [Hadoop](https://www.geeksforgeeks.org/hadoop-an-introduction/) or there was no concept of big data at that point in time all the data is used to be stored in the relational database management system. But nowadays after the introduction of concepts of Big data, the data need to be stored in a more concise and effective way. Thus Sqoop comes into existence.

So all the data which are stored in a relational database management system needed to be transferred into the Hadoop structure.

So the transfer of this large amount of data manually is not possible but with the help of Sqoop, we can able to do it.

Thus Sqoop is defined as the tool which is used to perform data transfer operations from relational database management system to Hadoop server. Thus it helps in transfer of bulk of data from one point of source to another point of source.

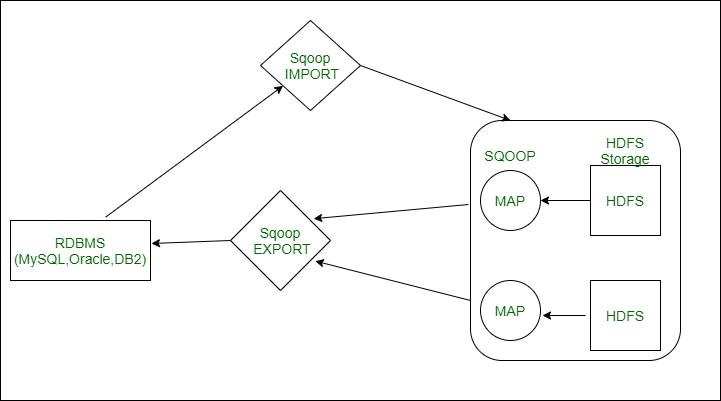
**Some of the important Features of the Sqoop :**

* Sqoop also helps us to connect the result from the SQL Queries into Hadoop distributed file system.
* Sqoop helps us to load the processed data directly into the hive or Hbase.
* It performs the security operation of data with the help of Kerberos.
* With the help of Sqoop, we can perform compression of processed data.
* Sqoop is highly powerful and efficient in nature.

**There are two major operations performed in Sqoop :**

1. Import
2. Export

**Sqoop Working:**



**SQOOP ARCHITECTURE**

* Basically the operations that take place in Sqoop are usually user-friendly. Sqoop used the command-line interface to process command of user.
* The Sqoop can also use alternative ways by using Java APIs to interact with the user. Basically, when it receives command by the user, it is handled by the Sqoop and then the further processing of the command takes place.
* Sqoop will only be able to perform the import and export of data based on user command it is not able to form an aggregation of data.
* **Sqoop** is a tool in which works in the following manner, it first parses argument which is provided by user in the command-line interface and then sends those arguments to a further stage where arguments are induced for Map only job.
* Once the Map receives arguments it then gives command of release of multiple mappers depending upon the number defined by the user as an argument in command line Interface.
* Once these jobs are then for Import command, each mapper task is assigned with respective part of data that is to be imported on basis of key which is defined by user in the command line interface.
* To increase efficiency of process Sqoop uses parallel processing technique in which data is been distributed equally among all mappers. After this, each mapper then creates an individual connection with the database by using java database connection model and then fetches individual part of the data assigned by Sqoop.
* Once the data is been fetched then the data is been written in HDFS or Hbase or Hive on basis of argument provided in command line. thus the process Sqoop import is completed.
* The export process of the data in **Sqoop**is performed in same way, Sqoop export tool which available performs the operation by allowing set of files from the Hadoop distributed system back to the Relational Database management system.
* The files which are given as an input during import process are called records, after that when user submits its job then it is mapped into Map Task that brings the files of data from Hadoop data storage, and these data files are exported to any structured data destination which is in the form of relational database management system such as MySQL, SQL Server, and Oracle, etc.

Let us now understand the two main **operations** in detail:

**Sqoop Import:**

Sqoop import command helps in implementation of the operation. With the help of the import command, we can import a table from the Relational database management system to the Hadoop database server. Records in Hadoop structure are stored in text files and each record is imported as a separate record in Hadoop database server. We can also create load and partition in Hive while importing data..Sqoop also supports incremental import of data which means in case we have imported a database and we want to add some more rows, so with the help of these functions we can only add the new rows to existing database, not the complete database.

**Sqoop Export :**

Sqoop export command helps in the implementation of operation. With the help of the export command which works as a reverse process of operation. Herewith the help of the export command we can transfer the data from the Hadoop database file system to the Relational database management system. The data which will be exported is processed into records before operation is completed. The export of data is done with two steps, first is to examine the database for metadata and second step involves migration of data.

[Here you can get the idea of how the import and export operation is performed in Hadoop with the help of Sqoop.](https://www.geeksforgeeks.org/import-and-export-data-using-sqoop/)

**Advantages of Sqoop :**

* With the help of Sqoop, we can perform transfer operations of data with a variety of structured data stores like Oracle, Teradata, etc.
* Sqoop helps us to perform ETL operations in a very fast and cost-effective manner.
* With the help of Sqoop, we can perform parallel processing of data which leads to fasten the overall process.
* Sqoop uses the MapReduce mechanism for its operations which also supports fault tolerance.

**Disadvantages of Sqoop :**

* The failure occurs during the implementation of operation needed a special solution to handle the problem.
* The Sqoop uses JDBC connection to establish a connection with the relational database management system which is an inefficient way.
* The performance of Sqoop export operation depends upon hardware configuration relational database management system.

**What is Apache ZooKeeper?**

* **Zookeeper**is a distributed, open-source coordination service for distributed applications. It exposes a simple set of primitives to implement higher-level services for synchronization, configuration maintenance, and group and naming.
* In a distributed system, there are multiple nodes or machines that need to communicate with each other and coordinate their actions. ZooKeeper provides a way to ensure that these nodes are aware of each other and can coordinate their actions.
* It does this by maintaining a hierarchical tree of data nodes called “**Znodes**“, which can be used to store and retrieve data and maintain state information. ZooKeeper provides a set of primitives, such as locks, barriers, and queues, that can be used to coordinate the actions of nodes in a distributed system.
* It also provides features such as leader election, failover, and recovery, which can help ensure that the system is resilient to failures. ZooKeeper is widely used in distributed systems such as Hadoop, Kafka, and HBase, and it has become an essential component of many distributed applications.

**Why do we need it?**

* **Coordination services**: The integration/communication of services in a distributed environment.
* Coordination services are complex to get right. They are especially prone to errors such as race conditions and deadlock.
* **Race condition**-Two or more systems trying to perform some task.
* **Deadlocks**– Two or more operations are waiting for each other.
* To make the coordination between distributed environments easy, developers came up with an idea called zookeeper so that they don’t have to relieve distributed applications of the responsibility of implementing coordination services from scratch.

**What is distributed system?**

* Multiple computer systems working on a single problem.
* It is a network that consists of autonomous computers that are connected using distributed middleware.
* **Key Features**: Concurrent, resource sharing, independent, global, greater fault tolerance, and price/performance ratio is much better.
* **Key Goal**s: Transparency, Reliability, Performance, Scalability.
* **Challenges**: Security, Fault, Coordination, and resource sharing.

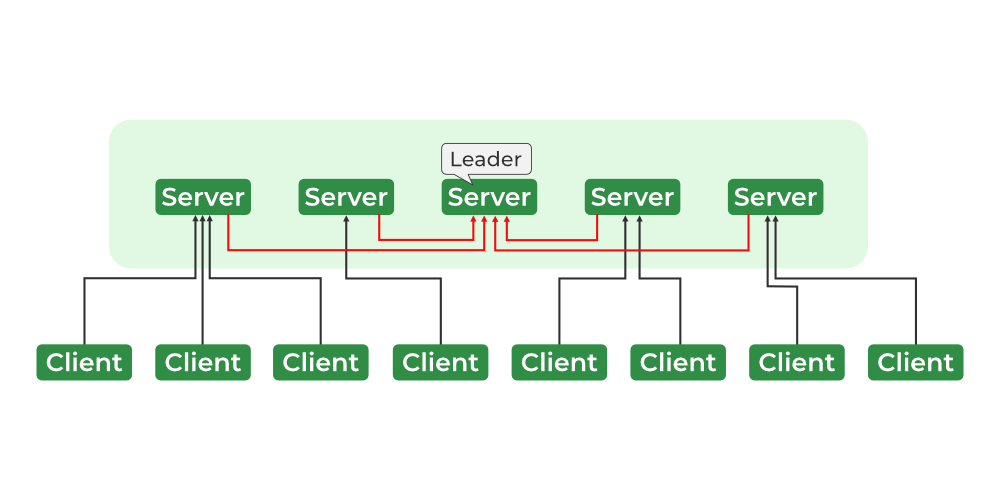
**Coordination Challenge**

* Why is coordination in a distributed system the hard problem?
* Coordination or configuration management for a distributed application that has many systems.
* Master Node where the cluster data is stored.
* Worker nodes or slave nodes get the data from this master node.
* single point of failure.
* synchronization is not easy.
* Careful design and implementation are needed.

**Apache Zookeeper**

* Apache Zookeeper is a distributed, open-source coordination service for distributed systems. It provides a central place for distributed applications to store data, communicate with one another, and coordinate activities.
* Zookeeper is used in distributed systems to coordinate distributed processes and services. It provides a simple, tree-structured data model, a simple API, and a distributed protocol to ensure data consistency and availability.
* Zookeeper is designed to be highly reliable and fault-tolerant, and it can handle high levels of read and write throughput.
* Zookeeper is implemented in Java and is widely used in distributed systems, particularly in the Hadoop ecosystem. It is an Apache Software Foundation project and is released under the Apache License 2.0.

**Architecture of Zookeeper**



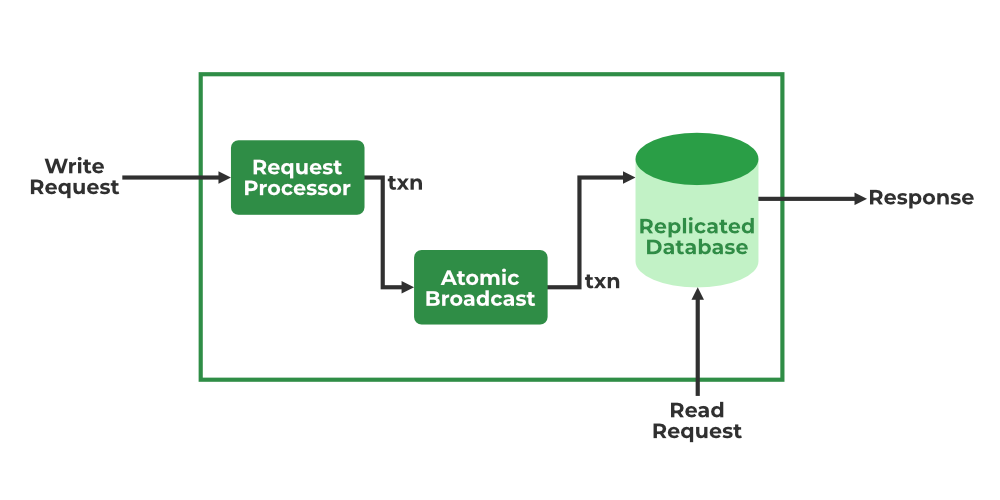
**Zookeeper Services**

The ZooKeeper architecture consists of a hierarchy of nodes called znodes, organized in a tree-like structure. Each znode can store data and has a set of permissions that control access to the znode.

The znodes are organized in a hierarchical namespace, similar to a file system. At the root of the hierarchy is the root znode, and all other znodes are children of the root znode.

The hierarchy is similar to a file system hierarchy, where each znode can have children and grandchildren, and so on.

**Important Components in Zookeeper**



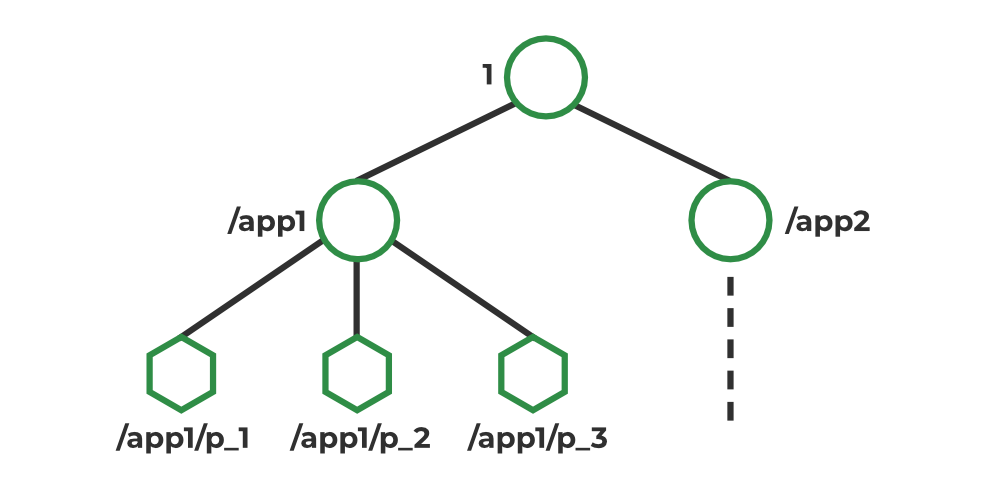
**ZooKeeper Services**

* **Leader & Follower**
* **Request Processor** – Active in Leader Node and is responsible for processing write requests. After processing, it sends changes to the follower nodes
* **Atomic Broadcast** – Present in both Leader Node and Follower Nodes. It is responsible for sending the changes to other Nodes.
* **In-memory Databases** (Replicated Databases)-It is responsible for storing the data in the zookeeper. Every node contains its own databases. Data is also written to the file system providing recoverability in case of any problems with the cluster.

**Other Components**

* **Client**– One of the nodes in our distributed application cluster. Access information from the server. Every client sends a message to the server to let the server know that client is alive.
* **Server**– Provides all the services to the client. Gives acknowledgment to the client.
* **Ensemble**– Group of Zookeeper servers. The minimum number of nodes that are required to form an ensemble is 3.

**Zookeeper Data Model**



**ZooKeeper data model**

In Zookeeper, data is stored in a hierarchical namespace, similar to a file system. Each node in the namespace is called a Znode, and it can store data and have children. Znodes are similar to files and directories in a file system. Zookeeper provides a simple API for creating, reading, writing, and deleting Znodes. It also provides mechanisms for detecting changes to the data stored in Znodes, such as watches and triggers. Znodes maintain a stat structure that includes: Version number, ACL, Timestamp, Data Length

**Types of Znodes**:

* **Persistence**: Alive until they’re explicitly deleted.
* **Ephemeral**: Active until the client connection is alive.
* **Sequential**: Either persistent or ephemeral.

**Why do we need ZooKeeper in the Hadoop?**

Zookeeper is used to manage and coordinate the nodes in a Hadoop cluster, including the NameNode, DataNode, and ResourceManager. In a Hadoop cluster, Zookeeper helps to:

* Maintain configuration information: Zookeeper stores the configuration information for the Hadoop cluster, including the location of the NameNode, DataNode, and ResourceManager.
* Manage the state of the cluster: Zookeeper tracks the state of the nodes in the Hadoop cluster and can be used to detect when a node has failed or become unavailable.
* Coordinate distributed processes: Zookeeper can be used to coordinate distributed processes, such as job scheduling and resource allocation, across the nodes in a Hadoop cluster.

Zookeeper helps to ensure the availability and reliability of a Hadoop cluster by providing a central coordination service for the nodes in the cluster.

**How ZooKeeper in Hadoop Works?**

* ZooKeeper operates as a distributed file system and exposes a simple set of APIs that enable clients to read and write data to the file system. It stores its data in a tree-like structure called a znode, which can be thought of as a file or a directory in a traditional file system.
* ZooKeeper uses a consensus algorithm to ensure that all of its servers have a consistent view of the data stored in the Znodes.
* This means that if a client writes data to a znode, that data will be replicated to all of the other servers in the ZooKeeper ensemble.
* One important feature of ZooKeeper is its ability to support the notion of a “watch.” A watch allows a client to register for notifications when the data stored in a znode changes.
* This can be useful for monitoring changes to the data stored in ZooKeeper and reacting to those changes in a distributed system.

In Hadoop, ZooKeeper is used for a variety of purposes, including:

* Storing configuration information: ZooKeeper is used to store configuration information that is shared by multiple Hadoop components. For example, it might be used to store the locations of NameNodes in a Hadoop cluster or the addresses of JobTracker nodes.
* Providing distributed synchronization: ZooKeeper is used to coordinate the activities of various Hadoop components and ensure that they are working together in a consistent manner. For example, it might be used to ensure that only one NameNode is active at a time in a Hadoop cluster.
* Maintaining naming: ZooKeeper is used to maintain a centralized naming service for Hadoop components. This can be useful for identifying and locating resources in a distributed system.

ZooKeeper is an essential component of Hadoop and plays a crucial role in coordinating the activity of its various subcomponents.

**Reading and Writing in Apache Zookeeper**

ZooKeeper provides a simple and reliable interface for reading and writing data. The data is stored in a hierarchical namespace, similar to a file system, with nodes called znodes. Each znode can store data and have children znodes. ZooKeeper clients can read and write data to these znodes by using the getData() and setData() methods, respectively. Here is an example of reading and writing data using the ZooKeeper Java API:

* Java
* Python3

|  |
| --- |
| // Connect to the ZooKeeper ensemble  ZooKeeper zk = new ZooKeeper("localhost:2181", 3000, null);  // Write data to the znode "/myZnode"  String path = "/myZnode";  String data = "hello world";  zk.create(path, data.getBytes(), Ids.OPEN\_ACL\_UNSAFE, CreateMode.PERSISTENT);  // Read data from the znode "/myZnode"  byte[] bytes = zk.getData(path, false, null);  String readData = new String(bytes);  // Prints "hello world"  System.out.println(readData);  // Closing the connection  // to the ZooKeeper ensemble  zk.close(); |

**Session and Watches**

**Session**

* Requests in a session are executed in FIFO order.
* Once the session is established then the**session id** is assigned to the client.
* Client sends **heartbeats** to keep the session valid
* session timeout is usually represented in milliseconds

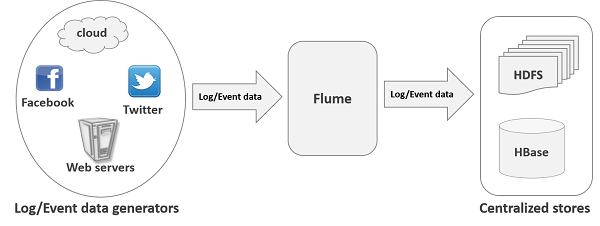
**Watches**

* Watches are mechanisms for clients to get notifications about the changes in the Zookeeper
* Client can watch while reading a particular znode.
* Znodes changes are modifications of data associated with the znodes or changes in the znode’s children.
* Watches are triggered only once.
* If the session is expired, watches are also removed.

## **What is Flume?**

Apache Flume is a tool/service/data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log files, events (etc...) from various sources to a centralized data store.

Flume is a highly reliable, distributed, and configurable tool. It is principally designed to copy streaming data (log data) from various web servers to HDFS.



## **Applications of Flume**

Assume an e-commerce web application wants to analyze the customer behavior from a particular region. To do so, they would need to move the available log data in to Hadoop for analysis. Here, Apache Flume comes to our rescue.

Flume is used to move the log data generated by application servers into HDFS at a higher speed.

## **Advantages of Flume**

Here are the advantages of using Flume −

* Using Apache Flume we can store the data in to any of the centralized stores (HBase, HDFS).
* When the rate of incoming data exceeds the rate at which data can be written to the destination, Flume acts as a mediator between data producers and the centralized stores and provides a steady flow of data between them.
* Flume provides the feature of **contextual routing**.
* The transactions in Flume are channel-based where two transactions (one sender and one receiver) are maintained for each message. It guarantees reliable message delivery.
* Flume is reliable, fault tolerant, scalable, manageable, and customizable.

## **Features of Flume**

Some of the notable features of Flume are as follows −

* Flume ingests log data from multiple web servers into a centralized store (HDFS, HBase) efficiently.
* Using Flume, we can get the data from multiple servers immediately into Hadoop.
* Along with the log files, Flume is also used to import huge volumes of event data produced by social networking sites like Facebook and Twitter, and e-commerce websites like Amazon and Flipkart.
* Flume supports a large set of sources and destinations types.
* Flume supports multi-hop flows, fan-in fan-out flows, contextual routing, etc.
* Flume can be scaled horizontally.

# Apache Oozie - Introduction

## **What is Apache Oozie?**

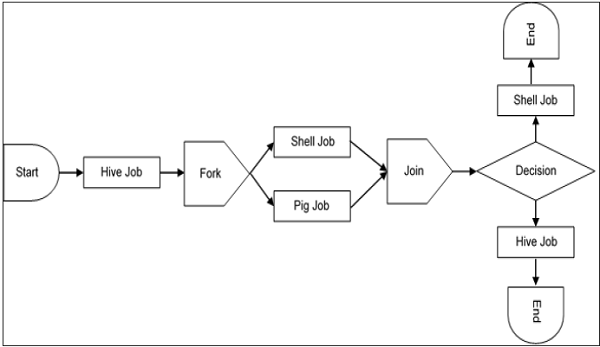
* Apache Oozie is a scheduler system to run and **manage Hadoop jobs** in a distributed environment. It allows to combine multiple complex jobs to be run in a sequential order to achieve a bigger task.
* Within a sequence of task, two or more jobs can also be programmed to run parallel to each other.
* One of the main advantages of Oozie is that it is tightly integrated with Hadoop stack supporting various Hadoop jobs like **Hive, Pig, Sqoop** as well as system-specific jobs like **Java and Shell**.
* Oozie is an **Open Source Java Web-Application** available under Apache license 2.0. It is responsible for triggering the workflow actions, which in turn uses the Hadoop execution engine to actually execute the task.
* Hence, Oozie is able to leverage the existing Hadoop machinery for load balancing, fail-over, etc. Oozie detects completion of tasks through callback and polling.
* When Oozie starts a task, it provides a unique **callback HTTP URL** to the task, and notifies that URL when it is complete. If the task fails to invoke the callback URL, Oozie can poll the task for completion.

Following three types of jobs are common in Oozie −

* **Oozie Workflow Jobs** − These are represented as Directed Acyclic Graphs (DAGs) to specify a sequence of actions to be executed.
* **Oozie Coordinator Jobs** − These consist of workflow jobs triggered by time and data availability.
* **Oozie Bundle** − These can be referred to as a package of multiple coordinator and workflow jobs.

We will look into each of these in detail in the following chapters.

A sample workflow with Controls (Start, Decision, Fork, Join and End) and Actions (Hive, Shell, Pig) will look like the following diagram −



Workflow will always start with a W567-Start tag and end with an End tag.

### Use-Cases of Apache Oozie

Apache Oozie is used by Hadoop system administrators to run complex log analysis on **HDFS**. Hadoop Developers use Oozie for performing ETL operations on data in a sequential order and saving the output in a specified format (Avro, ORC, etc.) in HDFS.

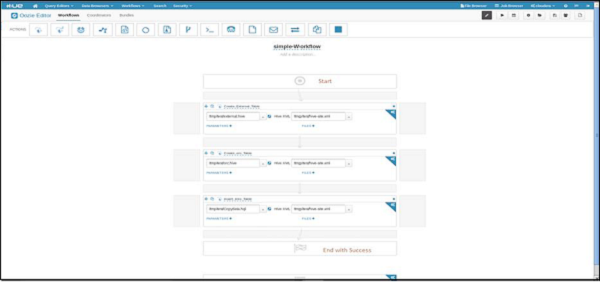
In an enterprise, Oozie jobs are scheduled as coordinators or bundles.

## **Oozie Editors**

* Before we dive into Oozie lets have a quick look at the available editors for Oozie.
* Most of the time, you won’t need an editor and will write the workflows using any popular text editors (like Notepad++, Sublime or Atom) as we will be doing in this tutorial.
* But as a beginner it makes some sense to create a workflow by the drag and drop method using the editor and then see how the workflow gets generated.
* Also, to map **GUI** with the actual **workflow.xml** created by the editor. This is the only section where we will discuss about Oozie editors and won’t use it in our tutorial.
* The most popular among Oozie editors is **Hue**.

### Hue Editor for Oozie

* This editor is very handy to use and is available with almost all Hadoop vendors’ solutions.
* The following screenshot shows an example workflow created by this editor.



You can drag and drop controls and actions and add your job inside these actions.

A good resource to learn more on this topic −

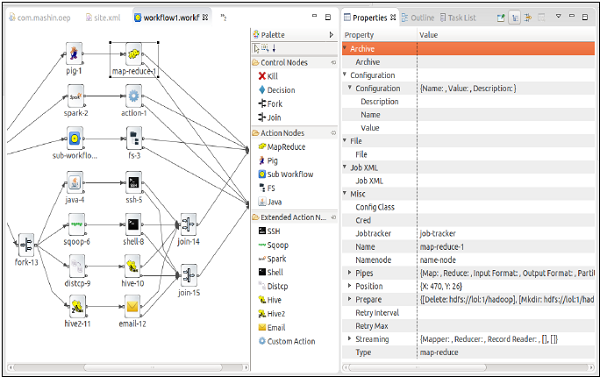
<http://gethue.com/new-apache-oozie-workflow-coordinator-bundle-editors/>

### Oozie Eclipse Plugin (OEP)

Oozie Eclipse plugin (OEP) is an Eclipse plugin for editing Apache Oozie workflows graphically. It is a graphical editor for editing Apache Oozie workflows inside Eclipse.

Composing Apache Oozie workflows is becoming much simpler. It becomes a matter of drag-and-drop, a matter of connecting lines between the nodes.

The following screenshots are examples of OEP.

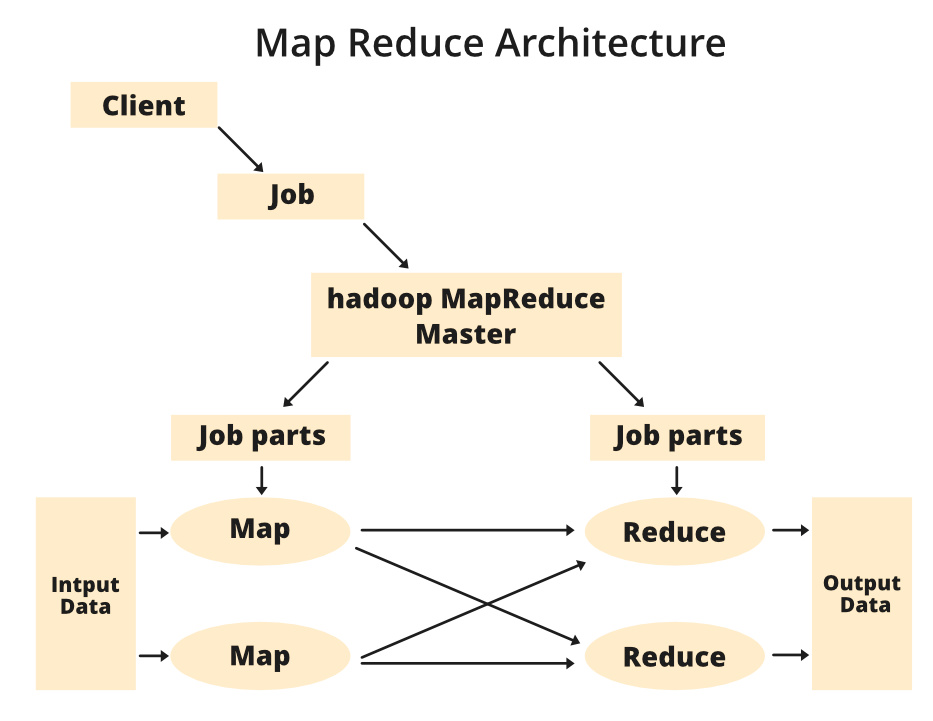


**UNIT-III**

**MapReduce Architecture**

* [MapReduce](https://www.geeksforgeeks.org/map-reduce-in-hadoop/) and [HDFS](https://www.geeksforgeeks.org/hadoop-hdfs-hadoop-distributed-file-system/) are the two major components of [Hadoop](https://www.geeksforgeeks.org/hadoop-introduction/) which makes it so powerful and efficient to use.
* MapReduce is a programming model used for efficient processing in parallel over large data-sets in a distributed manner.
* The data is first split and then combined to produce the final result. The libraries for MapReduce is written in so many programming languages with various different-different optimizations.
* The purpose of MapReduce in Hadoop is to Map each of the jobs and then it will reduce it to equivalent tasks for providing less overhead over the cluster network and to reduce the processing power.
* The MapReduce task is mainly divided into two phases [Map Phase](https://www.geeksforgeeks.org/hadoop-mapper-in-mapreduce/) and [Reduce Phase](https://www.geeksforgeeks.org/hadoop-reducer-in-map-reduce/).

**MapReduce Architecture:**



**Components of MapReduce Architecture:**

1. **Client:** The MapReduce client is the one who brings the Job to the MapReduce for processing. There can be multiple clients available that continuously send jobs for processing to the Hadoop MapReduce Manager.
2. **Job:** The MapReduce Job is the actual work that the client wanted to do which is comprised of so many smaller tasks that the client wants to process or execute.
3. **Hadoop MapReduce Master:** It divides the particular job into subsequent job-parts.
4. **Job-Parts:**  The task or sub-jobs that are obtained after dividing the main job. The result of all the job-parts combined to produce the final output.
5. **Input Data:** The data set that is fed to the MapReduce for processing.
6. **Output Data:** The final result is obtained after the processing.

* In **MapReduce**, we have a client. The client will submit the job of a particular size to the Hadoop MapReduce Master. Now, the MapReduce master will divide this job into further equivalent job-parts.
* These job-parts are then made available for the Map and Reduce Task. This Map and Reduce task will contain the program as per the requirement of the use-case that the particular company is solving.
* The developer writes their logic to fulfill the requirement that the industry requires. The input data which we are using is then fed to the Map Task and the Map will generate intermediate key-value pair as its output.
* The output of Map i.e. these key-value pairs are then fed to the Reducer and the final output is stored on the HDFS. There can be n number of Map and Reduce tasks made available for processing the data as per the requirement.
* The algorithm for Map and Reduce is made with a very optimized way such that the time complexity or space complexity is minimum.

Let’s discuss the MapReduce phases to get a better understanding of its architecture:

The MapReduce task is mainly divided into **2 phases** i.e. Map phase and Reduce phase.

1. **Map:** As the name suggests its main use is to map the input data in key-value pairs. The input to the map may be a key-value pair where the key can be the id of some kind of address and value is the actual value that it keeps. The Map() function will be executed in its memory repository on each of these input key-value pairs and generates the intermediate key-value pair which works as input for the Reducer or Reduce() function.
2. **Reduce:** The intermediate key-value pairs that work as input for Reducer are shuffled and sort and send to the Reduce() function. Reducer aggregate or group the data based on its key-value pair as per the reducer algorithm written by the developer.

How Job tracker and the task tracker deal with MapReduce:

1. **Job Tracker:** The work of Job tracker is to manage all the resources and all the jobs across the cluster and also to schedule each map on the Task Tracker running on the same data node since there can be hundreds of data nodes available in the cluster.
2. **Task Tracker:** The Task Tracker can be considered as the actual slaves that are working on the instruction given by the Job Tracker. This Task Tracker is deployed on each of the nodes available in the cluster that executes the Map and Reduce task as instructed by Job Tracker.

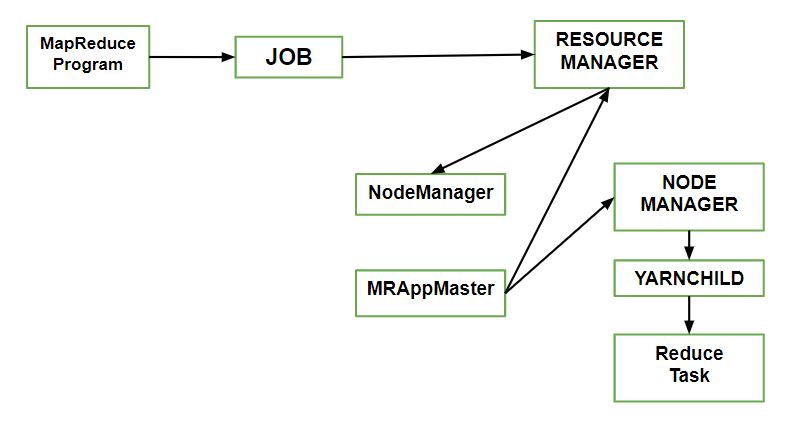
There is also one important component of MapReduce Architecture known as **Job History Server**. The Job History Server is a daemon process that saves and stores historical information about the task or application, like the logs which are generated during or after the job execution are stored on Job History Server.

**How Job runs on MapReduce**

MapReduce can be used to work with a solitary method call: **submit()** on a Job object (you can likewise call **waitForCompletion()**, which presents the activity on the off chance that it hasn’t been submitted effectively, at that point sits tight for it to finish).

Let’s understand the components –

1. **Client:** Submitting the MapReduce job.
2. **Yarn node manager:** In a cluster, it monitors and launches the compute containers on machines.
3. **Yarn resource manager:** Handles the allocation of computing resources coordination on the cluster.
4. **MapReduce application master**Facilitates the tasks running the MapReduce work.
5. **Distributed Filesystem:**Shares job files with other entities.



**How to submit Job?**

To create an internal JobSubmitter instance, use the **submit()** which further calls **submitJobInternal()** on it. Having submitted the job,

**waitForCompletion()** polls the job’s progress after submitting the job once per second. If the reports have changed since the last report, it further reports the progress to the console.

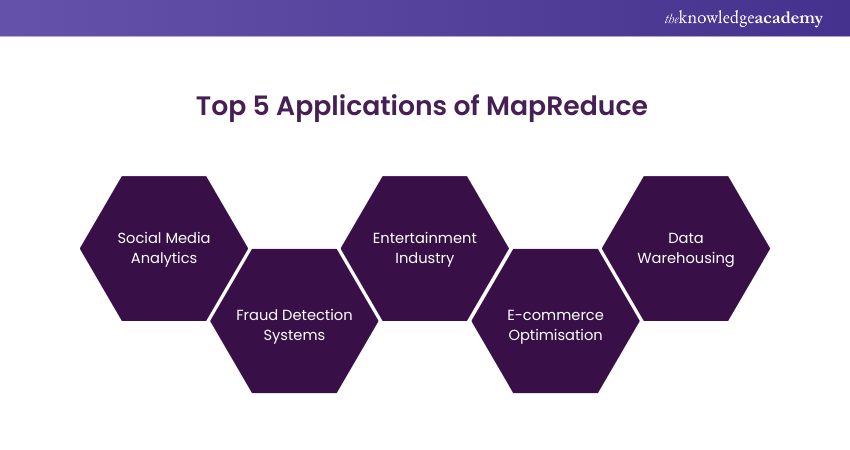
The job counters are displayed when the job completes successfully. Else the error (that caused the job to fail) is logged to the console.

Processes implemented by **JobSubmitter** for submitting the Job :

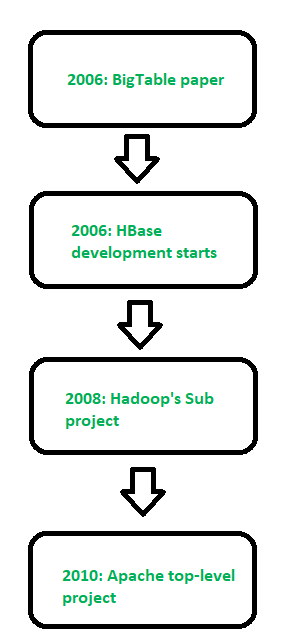
* The resource manager asks for a new application ID that is used for MapReduce Job ID.
* Output specification of the job is checked. For e.g. an error is thrown to the MapReduce program or the job is not submitted or the output directory already exists or it has not been specified.
* If the splits cannot be computed, it computes the input splits for the job. This can be due to the job is not submitted and an error is thrown to the MapReduce program.
* Resources needed to run the job are copied – it includes the job JAR file, and the computed input splits, to the shared filesystem in a directory named after the job ID and the configuration file.
* It copies job **JAR** with a high replication factor, which is controlled by **mapreduce.client.submit.file.replication property**. AS there are a number of copies across the cluster for the node managers to access.
* By calling **submitApplication()**, submits the job to the resource manager

**USE OF MAP REDUCE**

* Here are the top 5 uses of MapReduce:



* **a) Social Media Analytics:** MapReduce is used to analyse social media data to find trends and patterns. This analysis, facilitated by MapReduce, empowers organisations to make data-driven decisions and tailor their strategies to better engage with their target audience.
* **b) Fraud Detection Systems:** MapReduce is used to detect fraudulent activities in financial transactions. By leveraging this technology, organisations can enhance their fraud detection capabilities, mitigate risks, and safeguard the integrity of economic systems.
* **c) Entertainment Industry:**MapReduce is used to analyse user preferences and viewing history to recommend movies and TV shows. By analysing this information, the industry can deliver personalised recommendations for movies and TV shows, enhancing user experience and satisfaction.
* **d) E-commerce Optimisation:** MapReduce evaluates consumer buying patterns based on customers’ interests or historical purchasing patterns. This personalised approach enhances the overall shopping experience for consumers while improving the efficiency of e-commerce operations.
* e) Data Warehousing: MapReduce is used to process large volumes of data in data warehousing applications. In this way, organisations can derive actionable insights from their data, supporting informed decision-making processes across various business functions.
* Apache HBase Prerequisite– [Introduction to Hadoop](https://www.geeksforgeeks.org/hadoop-an-introduction/) HBase is a data model that is similar to Google’s big table. It is an open source, distributed database developed by Apache software foundation written in Java. HBase is an essential part of our Hadoop ecosystem. HBase runs on top of HDFS (Hadoop Distributed File System).
* It can store massive amounts of data from terabytes to petabytes. It is column oriented and horizontally scalable.



**Figure –** History of HBase

**Applications of Apache HBase:**

**Real-time analytics:**HBase is an excellent choice for real-time analytics applications that require low-latency data access. It provides fast read and write performance and can handle large amounts of data, making it suitable for real-time data analysis.

**Social media applications:** HBase is an ideal database for social media applications that require high scalability and performance. It can handle the large volume of data generated by social media platforms and provide real-time analytics capabilities.

**IoT applications:** HBase can be used for Internet of Things (IoT) applications that require storing and processing large volumes of sensor data. HBase’s scalable architecture and fast write performance make it a suitable choice for IoT applications that require low-latency data processing.

**Online transaction processing:** HBase can be used as an online transaction processing (OLTP) database, providing high availability, consistency, and low-latency data access. HBase’s distributed architecture and automatic failover capabilities make it a good fit for OLTP applications that require high availability.

**Ad serving and clickstream analysis:** HBase can be used to store and process large volumes of clickstream data for ad serving and clickstream analysis. HBase’s column-oriented data storage and indexing capabilities make it a good fit for these types of applications.

**Features of HBase –**It is linearly scalable across various nodes as well as modularly scalable, as it divided across various nodes. HBase provides consistent read and writes. It provides atomic read and write means during one read or write process, all other processes are prevented from performing any read or write operations. It provides easy to use Java API for client access.

1. It supports Thrift and REST API for non-Java front ends which supports XML, Protobuf and binary data encoding options. It supports a Block Cache and Bloom Filters for real-time queries and for high volum equerry optimization. HBase provides automatic failure support between Region Servers. It supports for exporting metrics with the Hadoop metrics subsystem of files. It doesn’t enforce relationship within your data. It is a platform for storing and retrieving data with random access.
2. **Facebook Messenger Platform** was using Apache Cassandra but it shifted from Apache Cassandra to HBase in November 2010. Facebook was trying to build a scalable and robust infrastructure to handle set of services like messages, email, chat and SMS into a real time conversation so that’s why HBase is best suited for that.

**RDBMSVsHBase –** 

1. RDBMS is mostly Row Oriented whereas HBase is Column Oriented.
2. RDBMS has fixed schema but in HBase we can scale or add columns in run time also.
3. RDBMS is good for structured data whereas HBase is good for semi-structured data.
4. RDBMS is optimized for joins but HBase is not optimized for joins.

Apache HBase is a NoSQL, column-oriented database that is built on top of the Hadoop ecosystem. It is designed to provide low-latency, high-throughput access to large-scale, distributed datasets. Here are some of the advantages and disadvantages of using HBase:

**Advantages Of Apache HBase:**

1. **Scalability:**HBase can handle extremely large datasets that can be distributed across a cluster of machines. It is designed to scale horizontally by adding more nodes to the cluster, which allows it to handle increasingly larger amounts of data.
2. **High-performance:**HBase is optimized for low-latency, high-throughput access to data. It uses a distributed architecture that allows it to process large amounts of data in parallel, which can result in faster query response times.
3. **Flexible data model:** HBase’s column-oriented data model allows for flexible schema design and supports sparse datasets. This can make it easier to work with data that has a variable or evolving schema.
4. **Fault tolerance:** HBase is designed to be fault-tolerant by replicating data across multiple nodes in the cluster. This helps ensure that data is not lost in the event of a hardware or network failure.

**Disadvantages Of Apache HBase:**

1. **Complexity:** HBase can be complex to set up and manage. It requires knowledge of the Hadoop ecosystem and distributed systems concepts, which can be a steep learning curve for some users.
2. **Limited query language:** HBase’s query language, HBase Shell, is not as feature-rich as SQL. This can make it difficult to perform complex queries and analyses.
3. **No support for transactions:** HBase does not support transactions, which can make it difficult to maintain data consistency in some use cases.
4. **Not suitable for all use cases:**HBase is best suited for use cases where high throughput and low-latency access to large datasets is required. It may not be the best choice for applications that require real-time processing or strong consistency guarantees.

**Apache Hive – Getting Started With HQL Database Creation And Drop Database**

HiveQL or HQL is a Hive query language that we used to process or query structured data on Hive. HQL syntaxes are very much similar to [MySQL](https://www.geeksforgeeks.org/sql-tutorial/) but have some significant differences. We will use the **hive**command, which is a bash shell script to complete our hive demo using CLI(Command Line Interface). We can easily start hive shell by simply typing hive in the terminal. Make sure that the **/bin**directory of your hive installation is mentioned in the **.basrc**file. The **.bashrc**file executes automatically when the user logs into the systemand all necessarycommands mentioned in this script file will run. We can simply check whether the **/bin**directory is available or not by simply opening it with the command as shown below.

sudo gedit ~/.bashrc

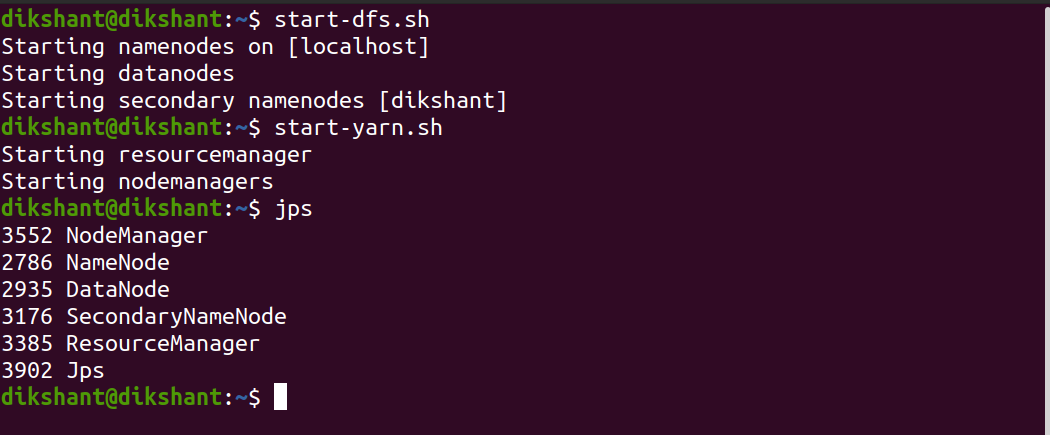
In case if the path is not added then add it so that we can directly run the hive shell from the terminal without moving to the hive directory. Otherwise, we can start hive manually by moving to **apache-hive-3.1.2/bin/**directory and by pressing the **hive** command.

Before performing hive make sure that all of your Hadoop daemons are started and working. We can simply start all the Hadoop daemon with the below command.

start-dfs.sh # this will start namenode, datanode and secondary namenode

start-yarn.sh # this will start node manager and resource manager

jps # To check running daemons



**Databases In Apache Hive**

The Database is a storage schema that contains multiple tables. The Hive Databases refer to the namespace of tables. If you don’t specify the database name by default Hive uses its default database for table creation and other purposes. Creating a Database allows multiple users to create tables with a similar name in different schemas so that their names don’t match.

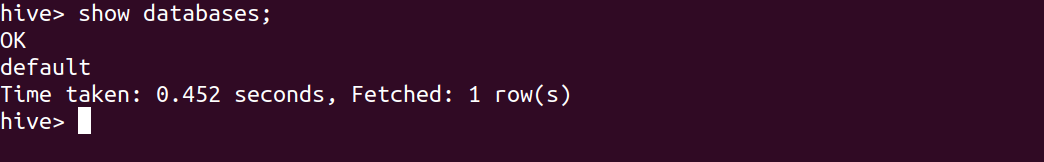
So, let’s start our hive shell for performing our tasks with the below command.

hive



See the already existing databases using the below command.

show databases; # this will show the existing databases



**Create Database Syntax:**

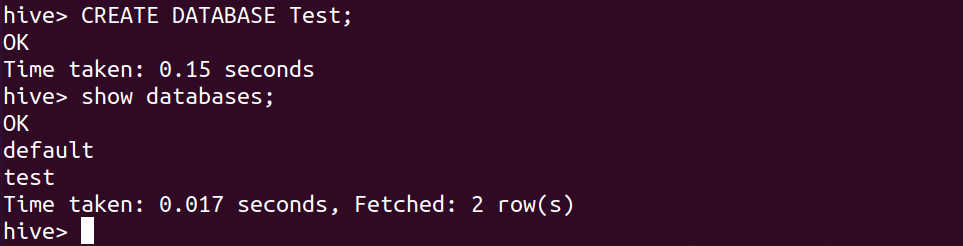
We can create a database with the help of the below command but if the database already exists then, in that case, Hive will throw an error.

CREATE DATABASE|SCHEMA <database name> # we can use DATABASE or SCHEMA for creation of DB

**Example:**

CREATE DATABASE Test; # create database with name Test

show databases; # this will show the existing databases



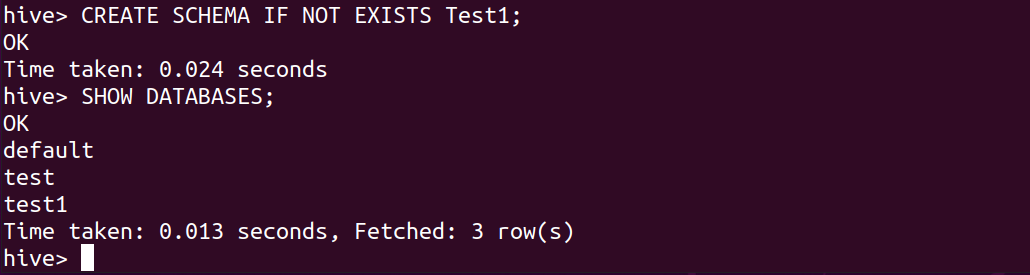
If we again try to create a Test database hive will throw an error/warning that the database with the name Test already exists. In general, we don’t want to get an error if the database exists. So we use the create database command with [IF NOT EXIST] clause. This will do not throw any error.

CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>

**Example:**

CREATE SCHEMA IF NOT EXISTS Test1;

SHOW DATABASES;



**Syntax To Drop Existing Databases:**

DROP DATABASE <db\_name>; or DROP DATABASE IF EXIST <db\_name> # The IF EXIST clause again is used to suppress error

**Example:**

DROP DATABASE IF EXISTS Test;

DROP DATABASE Test1;



Now quit hive shell with quit command.

quit;

https://media.geeksforgeeks.org/wp-content/uploads/20201030130512/quithive.png

## **Hive Services**

The following are the services provided by Hive:-

* Hive CLI - The Hive CLI (Command Line Interface) is a shell where we can execute Hive queries and commands.
* Hive Web User Interface - The Hive Web UI is just an alternative of Hive CLI. It provides a web-based GUI for executing Hive queries and commands.
* Hive MetaStore - It is a central repository that stores all the structure information of various tables and partitions in the warehouse. It also includes metadata of column and its type information, the serializers and deserializers which is used to read and write data and the corresponding HDFS files where the data is stored.
* Hive Server - It is referred to as Apache Thrift Server. It accepts the request from different clients and provides it to Hive Driver.
* Hive Driver - It receives queries from different sources like web UI, CLI, Thrift, and JDBC/ODBC driver. It transfers the queries to the compiler.
* Hive Compiler - The purpose of the compiler is to parse the query and perform semantic analysis on the different query blocks and expressions. It converts HiveQL statements into MapReduce jobs.
* Hive Execution Engine - Optimizer generates the logical plan in the form of DAG of map-reduce tasks and HDFS tasks. In the end, the execution engine executes the incoming tasks in the order of their dependencies.

# HIVE Data Types

Hive data types are categorized in numeric types, string types, misc types, and complex types. A list of Hive data types is given below.

## **Integer Types**

|  |  |  |
| --- | --- | --- |
| **Type** | **Size** | **Range** |
| TINYINT | 1-byte signed integer | -128 to 127 |
| SMALLINT | 2-byte signed integer | 32,768 to 32,767 |
| INT | 4-byte signed integer | 2,147,483,648 to 2,147,483,647 |
| BIGINT | 8-byte signed integer | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |

# HiveQL – Functions

# The Hive provides various in-built functions to perform mathematical and aggregate type operations. Here, we are going to execute such type of functions on the records of the below table:

# HiveQL - Functions

### Example of Functions in Hive

Let's create a table and load the data into it by using the following steps: -

Advertisement

* Select the database in which we want to create a table.

1. hive**>** use hql;

* Create a hive table using the following command: -

1. hive**>** create table employee\_data (Id int, Name string , Salary float)
2. row format delimited
3. fields terminated by ',' ;

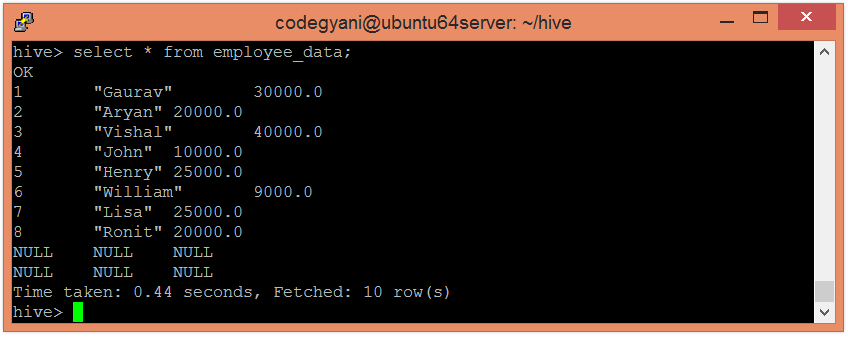
* Now, load the data into the table.

1. hive**>** load data local inpath '/home/codegyani/hive/emp\_details' into table employee\_data;

* Let's fetch the loaded data by using the following command: -

1. hive**>** select \* from employee\_data;

|  |  |  |
| --- | --- | --- |
| **Return type** | **Functions** | **Description** |
| BIGINT | round(num) | It returns the BIGINT for the rounded value of DOUBLE num. |
| BIGINT | floor(num) | It returns the largest BIGINT that is less than or equal to num. |
| BIGINT | ceil(num), ceiling(DOUBLE num) | It returns the smallest BIGINT that is greater than or equal to num. |
| DOUBLE | exp(num) | It returns exponential of num. |
| DOUBLE | ln(num) | It returns the natural logarithm of num. |
| DOUBLE | log10(num) | It returns the base-10 logarithm of num. |
| DOUBLE | sqrt(num) | It returns the square root of num. |
| DOUBLE | abs(num) | It returns the absolute value of num. |
| DOUBLE | sin(d) | It returns the sin of num, in radians. |
| DOUBLE | asin(d) | It returns the arcsin of num, in radians. |
| DOUBLE | cos(d) | It returns the cosine of num, in radians. |
| DOUBLE | acos(d) | It returns the arccosine of num, in radians. |
| DOUBLE | tan(d) | It returns the tangent of num, in radians. |
| DOUBLE | atan(d) | It returns the arctangent of num, in radians. |



Now, we discuss mathematical, aggregate and other in-built functions with the corresponding examples.

## **Mathematical Functions in Hive**

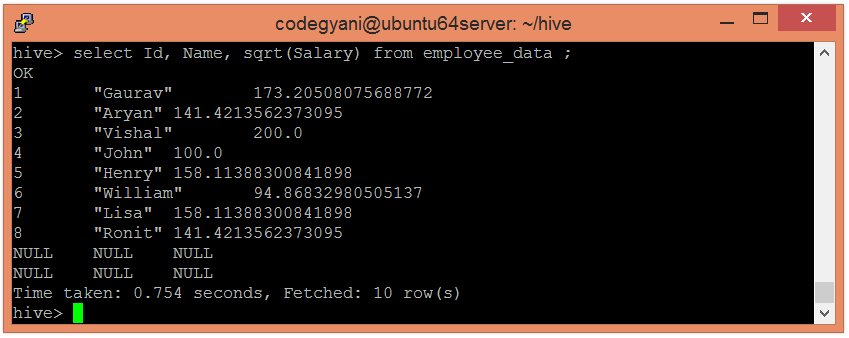
The commonly used mathematical functions in the hive are: -

Advertisement

ADVERTISING

### Example of Mathematical Functions in Hive

* Let's see an example to fetch the square root of each employee's salary.

1. hive**>** select Id, Name, sqrt(Salary) from employee\_data ;  
   

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Return Type** | | **Operator** | | **Description** | |
| BIGINT | count(\*) | | It returns the count of the number of rows present in the file. | |
| DOUBLE | sum(col) | | It returns the sum of values. | |
| DOUBLE | sum(DISTINCT col) | | It returns the sum of distinct values. | |
| DOUBLE | avg(col) | | It returns the average of values. | |
| DOUBLE | avg(DISTINCT col) | | It returns the average of distinct values. | |
| DOUBLE | min(col) | | It compares the values and returns the minimum one form it. | |
| DOUBLE | max(col) | | It compares the values and returns the maximum one form it. | |

## **Aggregate Functions in Hive**

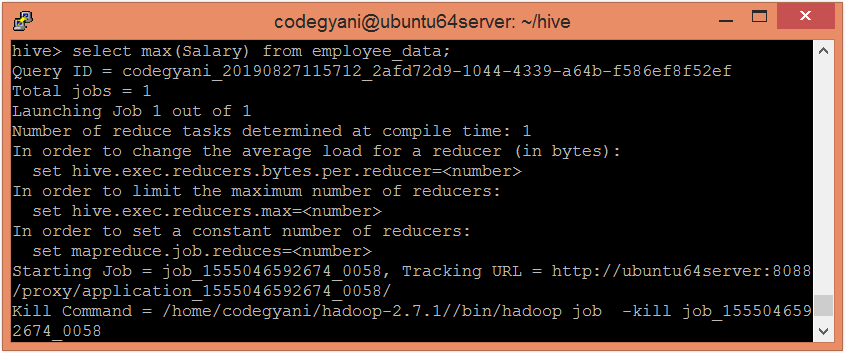
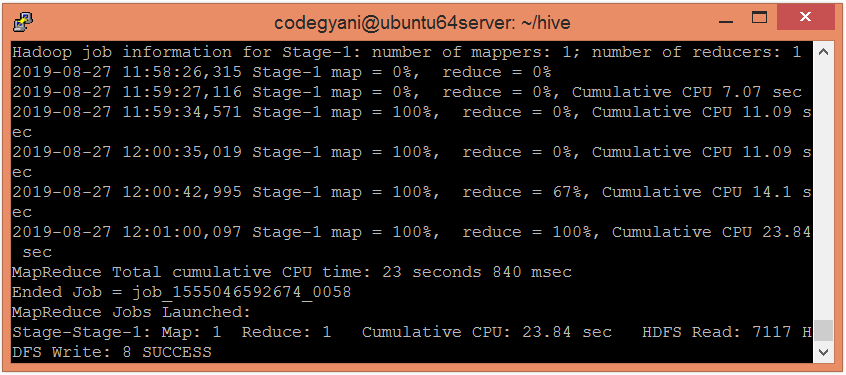
In Hive, the aggregate function returns a single value resulting from computation over many

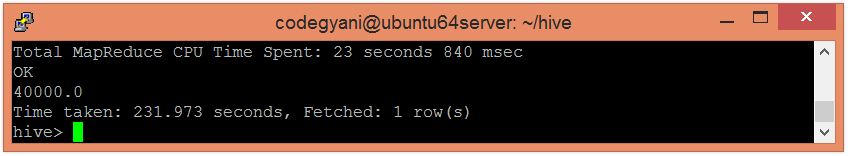
rows. Let''s see some commonly used aggregate functions: -

### Examples of Aggregate Functions in Hive

* Let's see an example to fetch the maximum salary of an employee.

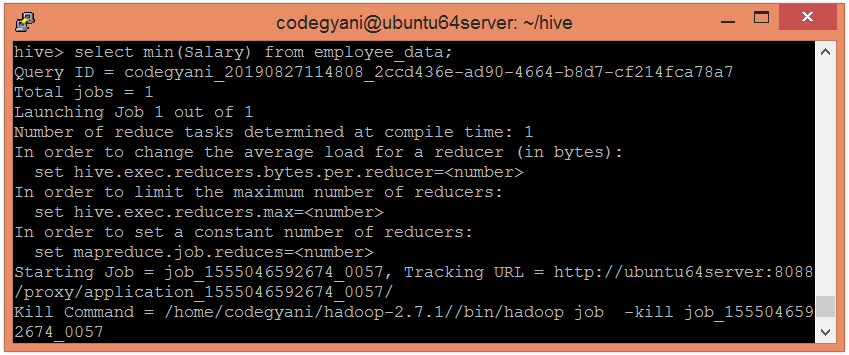
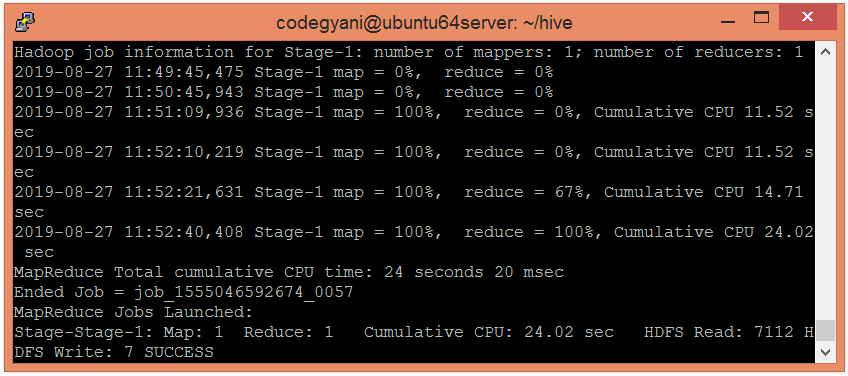
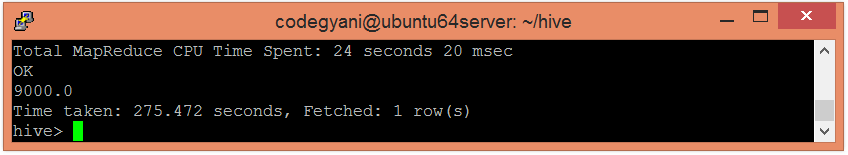
1. hive**>** select max(Salary) from employee\_data;



|  |  |  |
| --- | --- | --- |
| **Return Type** | **Operator** | **Description** |
| INT | length(str) | It returns the length of the string. |
| STRING | reverse(str) | It returns the string in reverse order. |
| STRING | concat(str1, str2, ...) | It returns the concatenation of two or more strings. |
| STRING | substr(str, start\_index) | It returns the substring from the string based on the provided starting index. |
| STRING | substr(str, int start, int length) | It returns the substring from the string based on the provided starting index and length. |
| STRING | upper(str) | It returns the string in uppercase. |
| STRING | lower(str) | It returns the string in lowercase. |
| STRING | trim(str) | It returns the string by removing whitespaces from both the ends. |
| STRING | ltrim(str) | It returns the string by removing whitespaces from left-hand side. |
| TRING | rtrim(str) | It returns the string by removing whitespaces from right-hand side. |

* Let's see an example to fetch the minimum
* salary of an employee.

1. hive**>** select min(Salary) from employee\_data;  
     
     
   

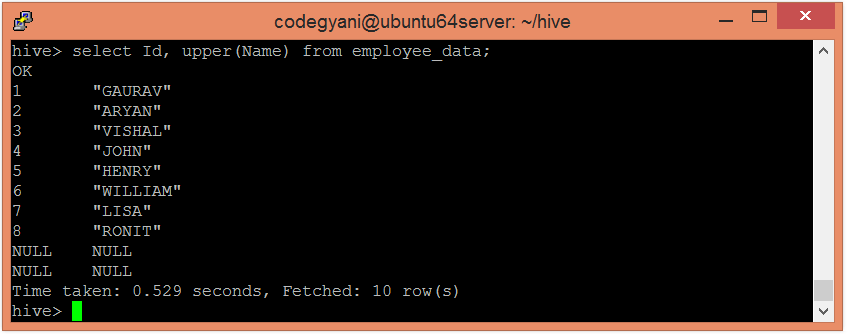
## **Other built-in Functions in Hive**

The following are some other commonly used in-built functions in the hive: -

### Examples of other in-built Functions in Hive

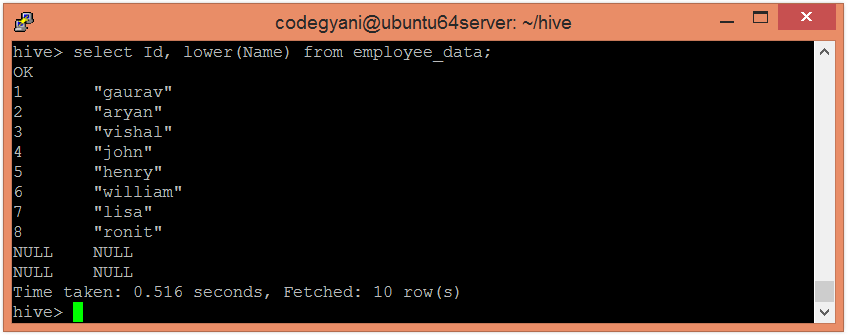
* Let's see an example to fetch the name of each employee in uppercase.

1. select Id, upper(Name) from employee\_data;



* Let's see an example to fetch the name of each employee in lowercase.

1. select Id, lower(Name) from employee\_data;



**Hive DDL Commands – Types of DDL Hive Commands**

Apache Hive DDL stands for (Data Definition Language) which is used to define or change the structure of Databases, Tables, indexes, and so on. The most commonly used DDL are CREATE, DROP, ALTER, SHOW, and so on.

The following is the list of DDL statements that are supported in Apache Hive.

1. CREATE
2. DROP
3. TRUNCATE
4. ALTER
5. SHOW
6. DESCRIBE
7. USE

| Commands | Commands Use With |
| --- | --- |
| CREATE | [DATABASE/SCHEMA](https://www.cloudduggu.com/hive/ddl-operations/#create_db), [TABLE](https://www.cloudduggu.com/hive/ddl-operations/#create_tb), VIEW, FUNCTION, INDEX |
| DROP | [DATABASE/SCHEMA](https://www.cloudduggu.com/hive/ddl-operations/#drop_db), [TABLE](https://www.cloudduggu.com/hive/ddl-operations/#drop_tb), VIEW, INDEX |
| TRUNCATE | [TABLE](https://www.cloudduggu.com/hive/ddl-operations/#truncate_tb) |
| ALTER | [DATABASE/SCHEMA](https://www.cloudduggu.com/hive/ddl-operations/#alter_db), [TABLE](https://www.cloudduggu.com/hive/ddl-operations/#alter_tb), VIEW |
| SHOW | [DATABASES/SCHEMAS](https://www.cloudduggu.com/hive/ddl-operations/#show_db), TABLES, TBLPROPERTIES, VIEWS, PARTITIONS, FUNCTIONS, INDEX[ES], COLUMNS, CREATE TABLE |
| DESCRIBE | DATABASE/SCHEMA, [TABLE\_NAME](https://www.cloudduggu.com/hive/ddl-operations/#describe_tb), VIEW\_NAME, MATERIALIZED\_VIEW\_NAME |
| USE | [DATABASE](https://www.cloudduggu.com/hive/ddl-operations/#use_db) |

Let us see each command in detail.

DDL Commands on Databases

1. Create Database

The Create Database command is useful in creating a database in Apache Hive. We can use the DATABASE as well as SCHEMA.

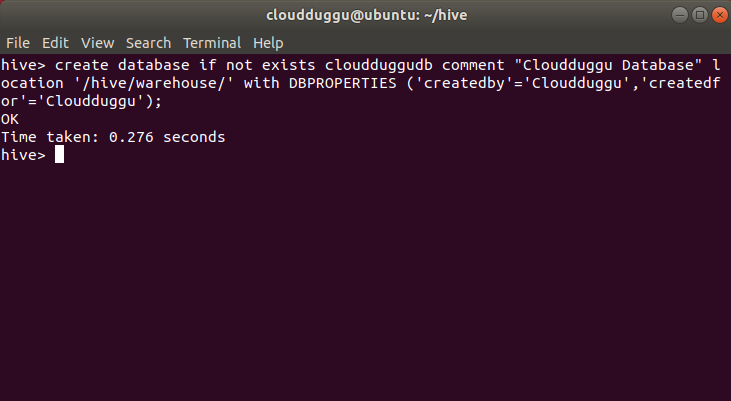
Create Database Syntax:

CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database\_name   
[COMMENT database\_comment]   
[LOCATION hdfs\_path]   
[MANAGEDLOCATION hdfs\_path]   
[WITH DBPROPERTIES (property\_name=property\_value, ...)];

Create Database Statement:

create database if not exists cloudduggudb   
comment "Cloudduggu Database"   
location '/hive/warehouse/' with   
DBPROPERTIES ('createdby'='Cloudduggu','createdfor'='Cloudduggu');

Command Output:



2. Drop Database

The Drop Database command is used to drop a database in Hive. The default mode is RESTRICT and the user will not able to drop a database unless the database is non-empty. If we want to delete a database then we will have to change the mode to CASCADE.

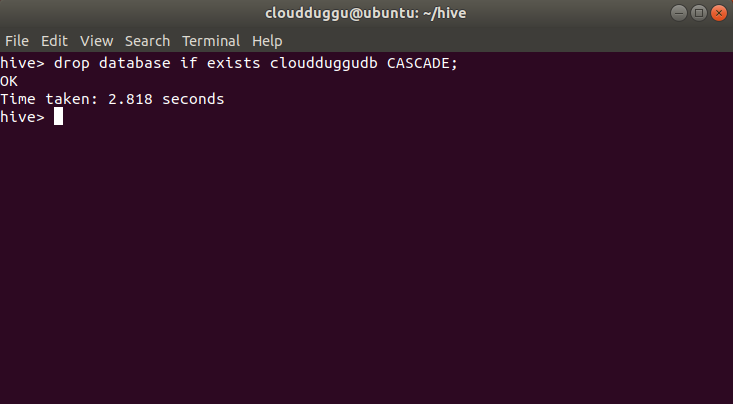
Drop Database Syntax:

DROP (DATABASE|SCHEMA) [IF EXISTS] database\_name [RESTRICT|CASCADE];

Drop Database Statement:

drop database if exists cloudduggudb CASCADE;

Command Output:



3. Alter Database

The Alter Database command is useful in altering the structure of the database.

Alter Database Syntax:

ALTER (DATABASE|SCHEMA) database\_name SET OWNER [USER|ROLE] user\_or\_role;

Alter Database Statement:

alter database cloudduggudb  set OWNER ROLE admin;

Command Output:

  
  
4. Use Database

The USE Database command is used in those case where we want to set a database and wants to operate on that particular database objects.

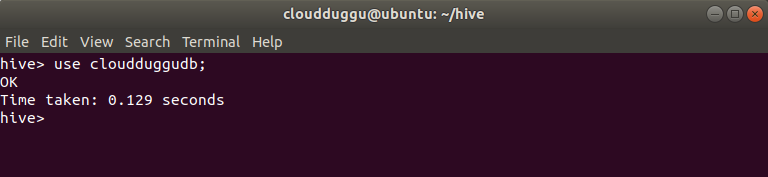
Use Database Syntax:

USE database\_name;

Use Database Statement:

USE cloudduggudb;

Command Output:

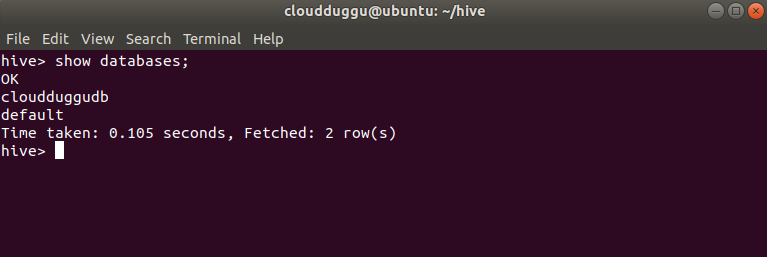
  
  
5. Show Database

The Show Database command is used to present the list of databases.

Show Database Syntax:

show databases;

Command Output:



DDL Commands on Tables

1. Create Table

The Create table command is used to create a table in the present database. In the below example we are creating a table named “serde\_example” in the cloudduggudb database. The table storage location in HDFS would be “/hive/warehouse/cloudduggudb.db/”.

Create Table Syntax:

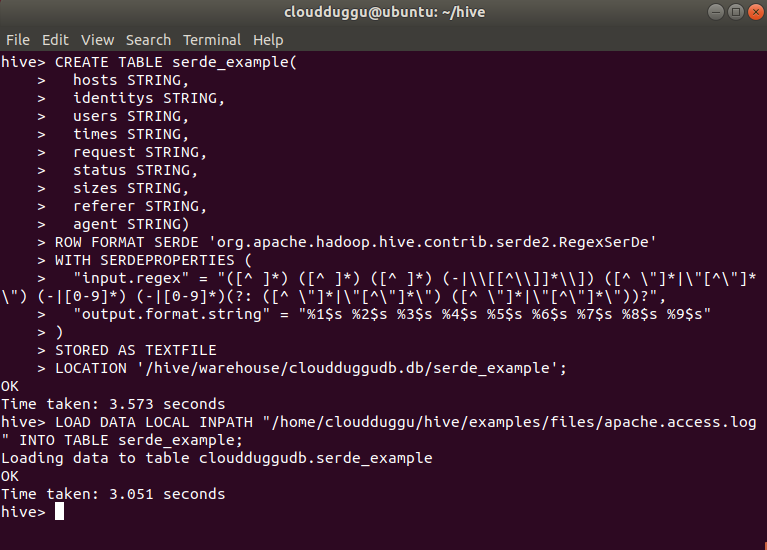
CREATE TABLE [IF NOT EXISTS] [db\_name.] table\_name   
[(col\_name data\_type [COMMENT col\_comment],   
[COMMENT col\_comment])]   
[COMMENT table\_comment]   
[ROW FORMAT row\_format]   
[STORED AS file\_format]   
[LOCATION hdfs\_path];

Create Table Statement:

CREATE TABLE serde\_example(   
  hosts STRING,   
  identitys STRING,   
  users STRING,   
  times STRING,   
  request STRING,   
  status STRING,   
  sizes STRING,   
  referer STRING,   
  agent STRING)

ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'   
WITH SERDEPROPERTIES (   
  "input.regex" = "([^ ]\*) ([^ ]\*) ([^ ]\*) (-|\\[[^\\]]\*\\]) ([^ \"]\*|\"[^\"]\*\") (-|[0-9]\*) (-|[0-9]\*)(?: ([^ \"]\*|\"[^\"]\*\") ([^ \"]\*|\"[^\"]\*\"))?",   
  "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s"   
)   
STORED AS TEXTFILE   
LOCATION '/hive/warehouse/cloudduggudb.db/serde\_example';

Command Output:



2. Create Table from Existing Table

In Apache Hive a new table can be created based on an existing table, in this process, the only table structure is created, table content is not copied.

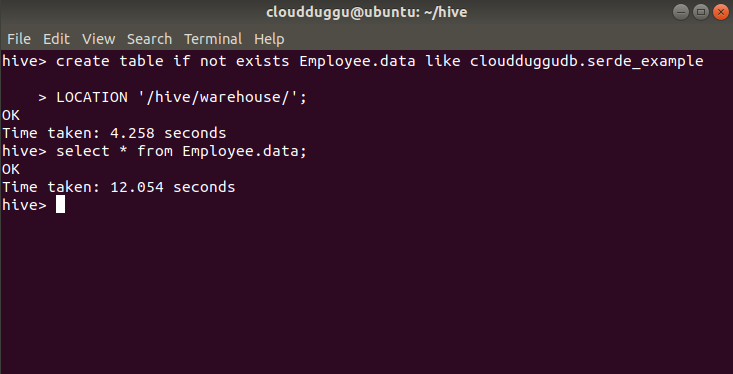
Create Table Syntax:

CREATE TABLE [IF NOT EXISTS] [db\_name.]table\_name   
Like [db\_name].existing\_table   
[LOCATION hdfs\_path]

Create Table Statement:

create table if not exists Employee.data   
like cloudduggudb.serde\_example   
LOCATION '/hive/warehouse/';

Command Output:

  
  
3. Truncate Table

The Truncate Table command is used to delete all rows from the table and no structure of the table. once the data is deleted it is pushed to the trash file system if that is enabled.

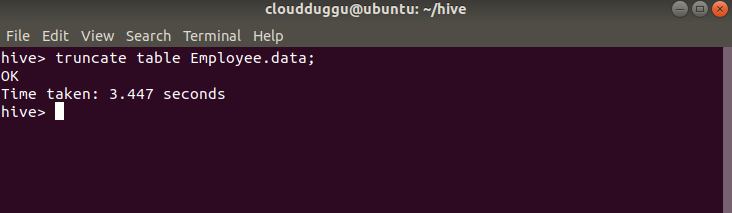
Truncate Table Syntax:

TRUNCATE [TABLE] table\_name [PARTITION partition\_spec];partition\_spec:   
: (partition\_column = partition\_col\_value, partition\_column = partition\_col\_value, …)

Truncate Table Statement:

truncate table Employee.data;

Command Output:



4. Alter Table

The Alter command is used to change the structure of the table.

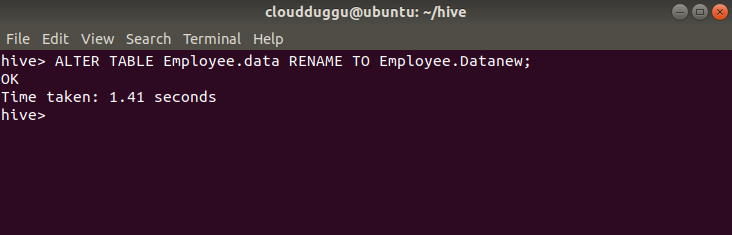
Rename Table Syntax:

ALTER TABLE table\_name RENAME TO new\_table\_name;

Rename Table Statement:

ALTER TABLE Employee.data RENAME TO Employee.Datanew;

Command Output:



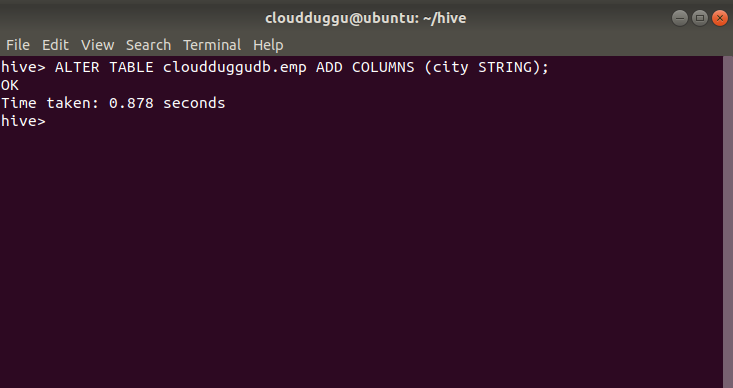
Add Column Syntax:

ALTER TABLE table\_name   
[PARTITION partition\_spec]   
ADD|REPLACE COLUMNS (col\_name data\_type [COMMENT col\_comment], ...)   
[CASCADE|RESTRICT]

Add Column Statement:

ALTER TABLE cloudduggudb.emp ADD COLUMNS (city STRING);

Command Output:



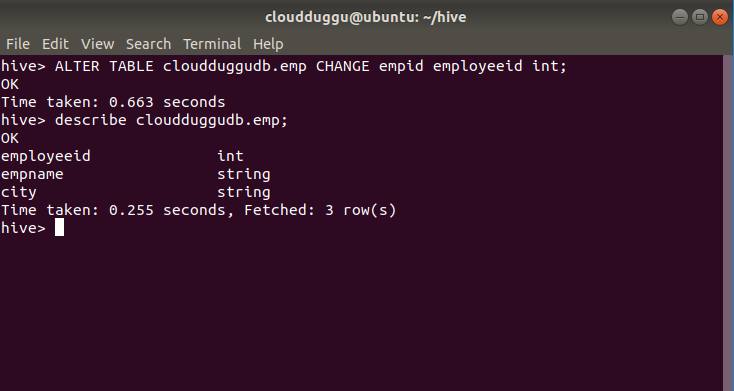
Rename Column Syntax:

ALTER TABLE table\_name [PARTITION partition\_spec] CHANGE [COLUMN] col\_old\_name col\_new\_name column\_type   
  [COMMENT col\_comment] [FIRST|AFTER column\_name] [CASCADE|RESTRICT];

Rename Column Statement:

ALTER TABLE cloudduggudb.emp CHANGE empid employeeid int;

Command Output:



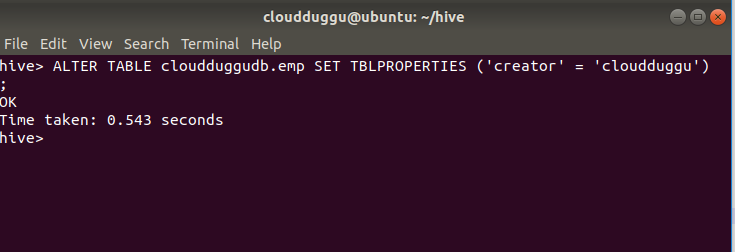
Alter Table Properties Syntax:

ALTER TABLE table\_name SET TBLPROPERTIES ('comment' = new\_comment);

Alter Table Properties Statement:

ALTER TABLE cloudduggudb.emp SET TBLPROPERTIES ('creator' = 'cloudduggu');

Command Output:



We can perform other operations using the Alter command.

| Name | Commands | Description |
| --- | --- | --- |
| Rename Table | ALTER TABLE table\_name1 RENAME TO new\_table\_name; | We can change the name of a table to a different name. |
| Alter Table Properties | ALTER TABLE table\_name2 SET TBLPROPERTIES table\_properties; table\_properties: : (property\_name = property\_value, property\_name = property\_value, ... )" | We can use this statement to add your own metadata to the tables. |
| Alter Table Comment | ALTER TABLE table\_name3 SET TBLPROPERTIES ('comment' = new\_comment); | We can change the comment for your table. |
| Add SerDe Properties | ALTER TABLE table\_name4 [PARTITION partition\_spec] SET SERDE serde\_class\_name [WITH SERDEPROPERTIES serde\_properties]; ALTER TABLE table\_name5 [PARTITION partition\_spec] SET SERDEPROPERTIES serde\_properties; serde\_properties: : (property\_name = property\_value, property\_name = property\_value, ... )" | We can change a table's SerDe or add user-defined metadata to the table's SerDe object. |
| Alter Table Storage Properties | ALTER TABLE table\_name6 CLUSTERED BY (col\_name, col\_name, ...) [SORTED BY (col\_name, ...)] INTO num\_buckets BUCKETS;" | We can change the table's physical storage properties. |
| Alter Table Constraints | ALTER TABLE table\_name7 ADD CONSTRAINT constraint\_name PRIMARY KEY (column, ...) DISABLE NOVALIDATE; | We can add or remove the table’s constraints using the Alter command. |
| Alter Partition(Add Partitions) | ALTER TABLE page\_view ADD PARTITION (dt='2010-09-08', country='ind') location '/path/to/user/part100908' PARTITION (dt='2010-08-09', country='ind') location '/path/to/user/part100809'; | We can use ALTER TABLE ADD PARTITION to add partitions to a table. |
| Alter Partition(Rename Partitions) | ALTER TABLE table\_name8 PARTITION partition\_spec RENAME TO PARTITION partition\_spec; | We can change the value of a partition column. |
| Alter Column | ALTER TABLE table\_name9 [PARTITION partition\_spec] CHANGE [COLUMN] col\_old\_name col\_new\_name column\_type [COMMENT col\_comment] [FIRST|AFTER column\_name] [CASCADE|RESTRICT]; | We can change a column's name, data type, comment, or position,or an arbitrary ombination of them. |

5. Describe Table

The Describe command shows the detailed structure of a table like its columns, data type, and so on.

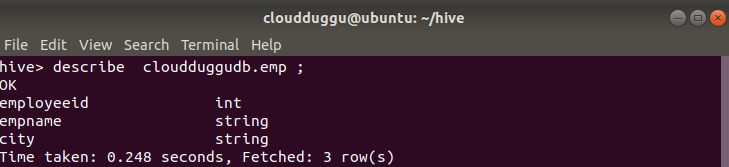
Describe Table Syntax:

DESCRIBE [EXTENDED|FORMATTED]table\_name[.col\_name ( [.field\_name] | [.'$elem$'] | [.'$key$'] | [.'$value$'] )\* ];

Describe Table Statement:

describe  cloudduggudb.emp;

Command Output:



6. Describe EXTENDED Table

The Describe usage with EXTENDED command shows the complete details of the table like table type, last access time, create time, last modification time, and so on.

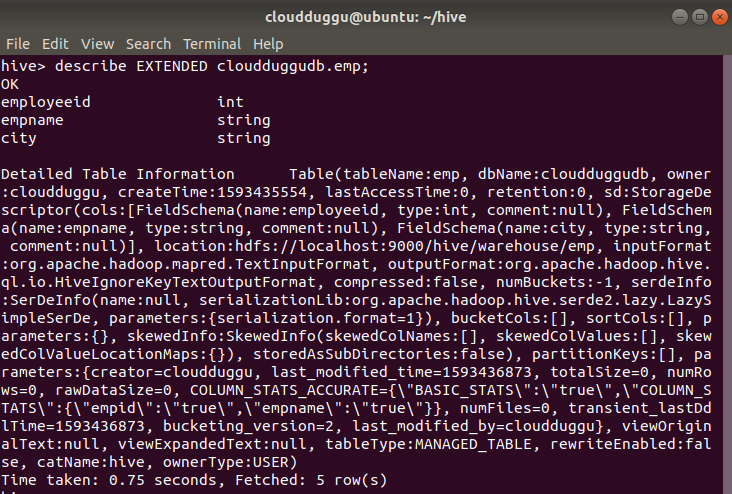
Describe EXTENDED Table Syntax:

DESCRIBE [EXTENDED|FORMATTED]table\_name[.col\_name ( [.field\_name] | [.'$elem$'] | [.'$key$'] | [.'$value$'] )\* ];

Describe EXTENDED Table Statement:

describe EXTENDED cloudduggudb.emp.

Command Output:



7. Describe FORMATTED Table

The Describe usage with FORMATTED command also shows similar details of the table like table type, last access time, create time, last modification time, and so on.

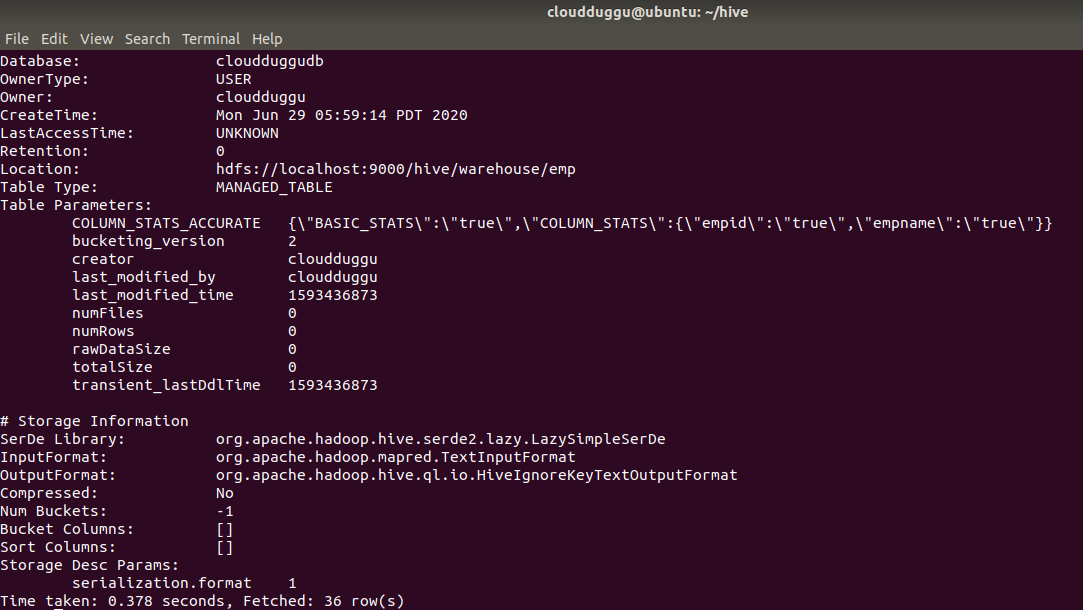
Describe FORMATTED Table Syntax:

DESCRIBE [EXTENDED|FORMATTED]table\_name[.col\_name ( [.field\_name] | [.'$elem$'] | [.'$key$'] | [.'$value$'] )\* ];

Describe FORMATTED Table Statement:

describe FORMATTED cloudduggudb.emp.

Command Output:



8. Drop-Table

The Drop-table command is used to drop the metadata and content of a table. When we perform the drop table command data is usually moved to the Trash/Current directory if Trash is configured.

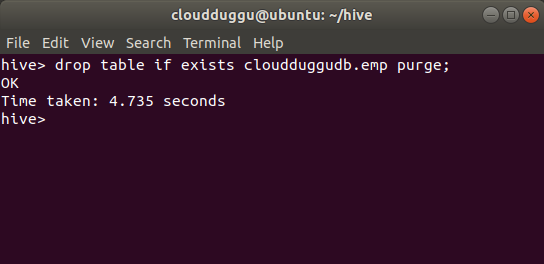
Drop-Table Syntax:

DROP TABLE [IF EXISTS] [db\_name.]table\_name [PURGE];

Drop-Table Statement:

drop table if exists cloudduggudb.emp purge;

**Command Output:**

****

**Apache Hive DML Operations**

Apache Hive DML stands for (Data Manipulation Language) which is used to insert, update, delete, and fetch data from Hive tables. Using DML commands we can load files into Apache Hive tables, write data into the filesystem from Hive queries, perform merge operation on the table, and so on.

The following list of DML statements is supported by Apache Hive.

* [LOAD](https://www.cloudduggu.com/hive/dml-operations/" \l "1load)
* [SELECT](https://www.cloudduggu.com/hive/dml-operations/" \l "2select)
* [INSERT](https://www.cloudduggu.com/hive/dml-operations/" \l "3insert)
* [DELETE](https://www.cloudduggu.com/hive/dml-operations/" \l "6delete)
* [UPDATE](https://www.cloudduggu.com/hive/dml-operations/" \l "7update)
* [EXPORT](https://www.cloudduggu.com/hive/dml-operations/" \l "8export)
* [IMPORT](https://www.cloudduggu.com/hive/dml-operations/" \l "9import)

Let us see each of the DML commands in detail.

**1. Load Command**

The load command is used to move datafiles into Hive tables. Load operations are pure copy/move operations.

* During the LOAD operation, if a LOCAL keyword is mentioned, then the LOAD command will check for the file path in the local filesystem.
* During the LOAD operation, if the LOCAL keyword is not mentioned, then the Hive will need the absolute URI of the file such as hdfs://namenode:9000/user/hive/project/data1.
* During LOAD operation, if the OVERWRITE keyword is mentioned, then the contents of the target table/partition will be deleted and replaced by the files referred by the file path.
* During LOAD operation, if the OVERWRITE keyword is not mentioned, then the files referred to by the file path will be appended to the table.

To perform load operation first we will create a table with the name “user\_data” and then we will load “u.data” data from the ml-100k.zip dataset.

You can download the ml-100k.zip dataset from the below link and save it in your HIVE\_HOME directory. We have saved this dataset under the below directory.

/home/cloudduggu/hive/examples/files/ml-100k/

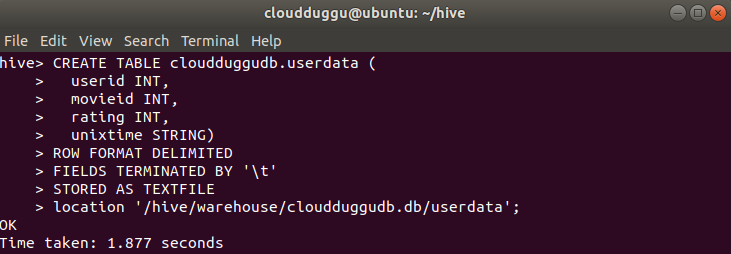
**Dataset Download Link:**

$wget http://files.grouplens.org/datasets/movielens/ml-100k.zip

**Create Table Statement:**

CREATE TABLE cloudduggudb.userdata (   
  userid INT,   
  movieid INT,   
  rating INT,   
  unixtime STRING)   
ROW FORMAT DELIMITED   
FIELDS TERMINATED BY '\t'   
STORED AS TEXTFILE   
location '/hive/warehouse/cloudduggudb.db/userdata';

**Command Output:**



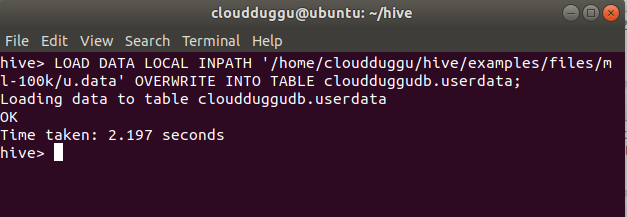
**Load Table Syntax:**

LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...)]

**Load Table Statement:**

LOAD DATA LOCAL INPATH '/home/cloudduggu/hive/examples/files/ml-100k/u.data' OVERWRITE INTO TABLE cloudduggudb.userdata;

**Command Output:**

  
  
**2. Select Command**

The Select statement project the records from the table.

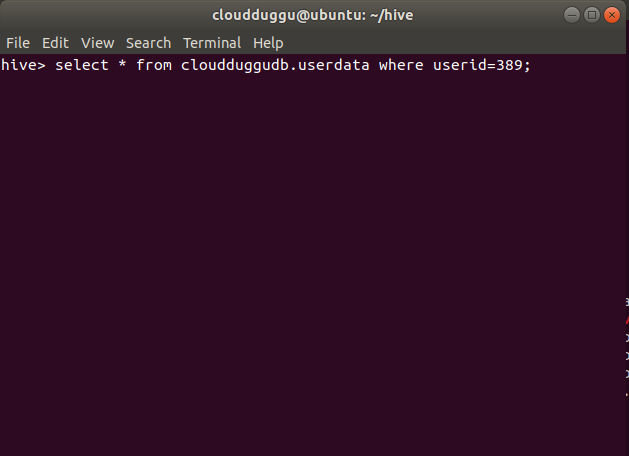
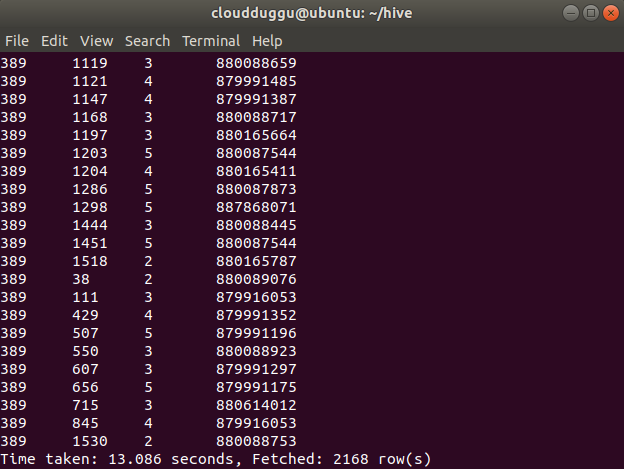
**Select Command Syntax:**

SELECT [ALL | DISTINCT] select\_expr, select\_expr, ...   
  FROM table\_reference   
  [WHERE where\_condition]   
  [GROUP BY col\_list]   
  [ORDER BY col\_list]   
  [CLUSTER BY col\_list   
    | [DISTRIBUTE BY col\_list] [SORT BY col\_list]   
  ]   
 [LIMIT [offset,] rows]

**Select Command Statement:**

SELECT \* FROM cloudduggudb.userdata WHERE userid=389;

**Command Output:**

  
  
  
  
**3. Insert into Command**

The Insert into command appends data from one table to another table.

We have created two tables to show this example. The name of the first table is “cloudduggudb.employee\_detail” and the name of the second table is “cloudduggudb. employee\_bkp”.

We have inserted data into cloudduggudb.employee\_detail using load command now we will copy the same data into cloudduggudb. employee\_bkp using INSERT INTO command.

**Create Table Statement:**

CREATE TABLE IF NOT EXISTS cloudduggudb.employee\_detail ( eid int, name String,   
salary String, destination String)   
COMMENT 'Employee details'   
ROW FORMAT DELIMITED   
FIELDS TERMINATED BY ' '   
LINES TERMINATED BY '\n'   
STORED AS TEXTFILE   
location '/hive/warehouse/cloudduggudb.db/employee\_detail';

**Load Data Statement:**

LOAD DATA LOCAL INPATH '/home/cloudduggu/hive/examples/files/customers.txt' OVERWRITE INTO TABLE cloudduggudb.employee\_detail;

**Create Table Statement:**

CREATE TABLE IF NOT EXISTS cloudduggudb.employee\_bkp ( eid int, name String,   
salary String, destination String)   
COMMENT 'Employee details'   
ROW FORMAT DELIMITED   
FIELDS TERMINATED BY ' '   
LINES TERMINATED BY '\n'   
STORED AS TEXTFILE   
location '/hive/warehouse/cloudduggudb.db/employee\_bkp';

After both table creations, let us use the INSERT INTO command to append data into cloudduggudb.employee\_bkp.

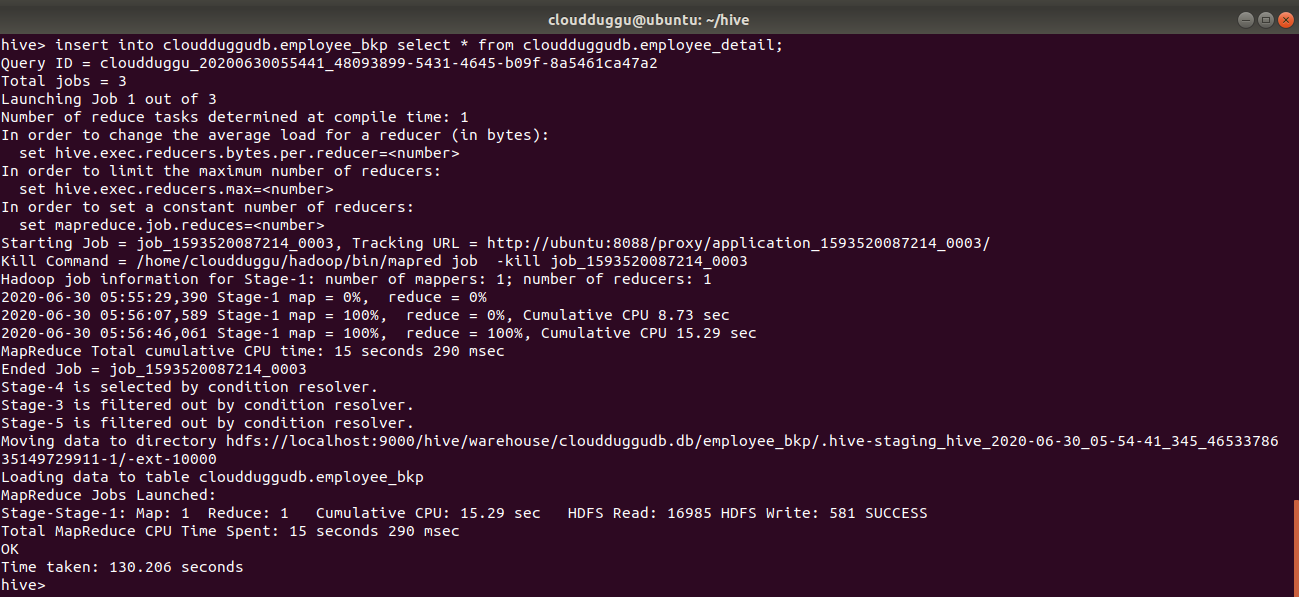
**Insert Into Syntax:**

INSERT INTO TABLE tablename1 [PARTITION (partcol1=val1, partcol2=val2 ...)] select\_statement1 FROM from\_statement;

**Insert Into Statement:**

INSERT INTO cloudduggudb.employee\_bkp SELECT \* FROM cloudduggudb.employee\_detail;

**Command Output:**

  
  
**4. Insert Overwrite Command**

The Insert overwrites perform the overwriting of the existing content of the table.

In this example we will use both tables which we used in the INSERT INTO section and overwrite the content of "cloudduggudb.employee\_bkp" with "cloudduggudb.employee\_detail".

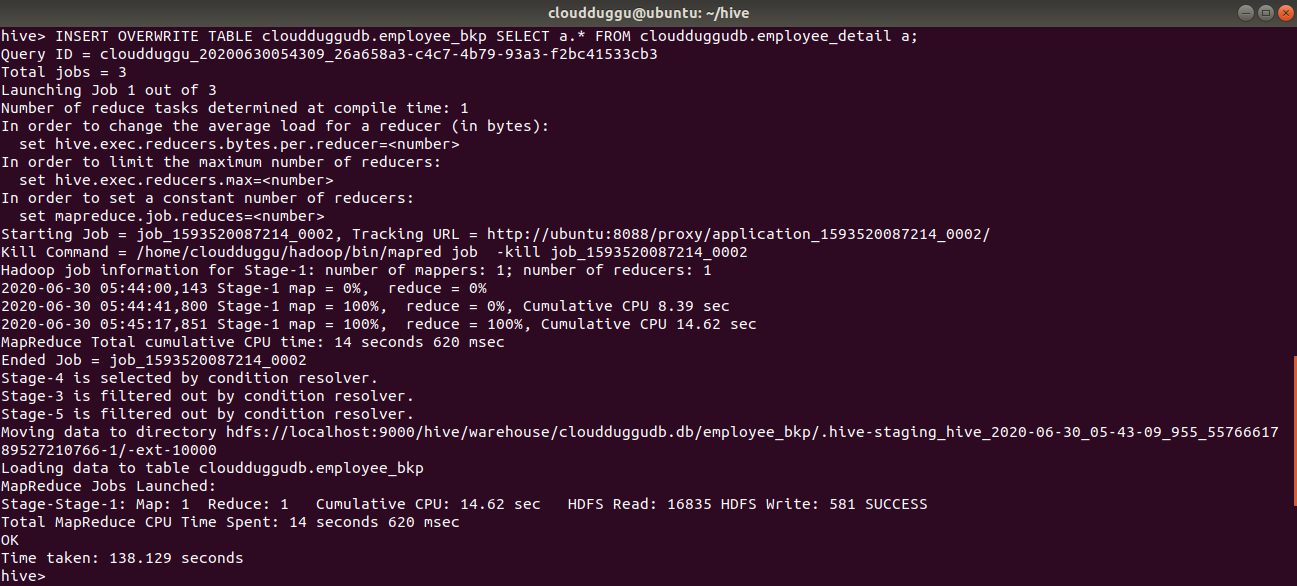
**Insert Overwrite Syntax:**

INSERT OVERWRITE TABLE tablename1 [PARTITION (partcol1=val1, partcol2=val2 ...) [IF NOT EXISTS]] select\_statement1 FROM from\_statement;

**Insert Overwrite Statement:**

INSERT OVERWRITE TABLE cloudduggudb.employee\_bkp SELECT a.\* FROM cloudduggudb.employee\_detail a;

**Command Output:**

  
  
  
**5. Insert Values Command**

By using the Insert Values command we can manually insert records in the existing table. We will use the “cloudduggudb.employee\_bkp” table and insert 2 records in that.

**Insert Values Syntax:**

INSERT INTO TABLE tablename [PARTITION (partcol1[=val1], partcol2[=val2] ...)] VALUES values\_row [, values\_row ...];

**Insert Values Statement:**

INSERT INTO cloudduggudb.employee\_bkp VALUES(1207,'Mahesh',70000,'Manager'),(1208,'Raj',70000,'Executive');

**Command Output:**

  
  
**6. Delete Command**

The Delete command is used to delete data from the table. If we supply where clause then it will delete that particular record only.

To perform Delete/Update operations in Apache Hive we need to follow the below points while creating a table otherwise delete/update statements with fail with error 10297.

**Note: In Apache Hive, We can perform a DELETE statement on those tables which follow the ACID property.**

Please follow the below points while creating the ACID enable table in Apache Hive.

1. Before performing the create, delete, update table we should enable the ACID property using the below parameters on Hive prompt.
   * hive>set hive.support.concurrency=true;
   * hive>set hive.enforce.bucketing=true;
   * hive>set hive.exec.dynamic.partition.mode=nonstrict;
   * hive>set hive.compactor.initiator.on=true;
   * hive>set hive.compactor.worker.threads=1;
   * hive>set hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
2. File format should be ORC which can be defined with TBLPROPERTIES(‘transactional’=’true’).
3. The table should be created with CLUSTERED BY followed by Buckets.

Now we will enable ACID property and create a table. After table creation, we will insert data.

**Create Table Statement:**

CREATE TABLE acidexample (key int, value int)   
PARTITIONED BY (load\_date date)   
CLUSTERED BY(key) INTO 3 BUCKETS   
STORED AS ORC TBLPROPERTIES ('transactional'='true');

**Insert Table Statement:**

INSERT INTO acidexample partition (load\_date='2016-03-01') VALUES (1, 1);

INSERT INTO acidexample partition (load\_date='2016-03-02') VALUES (2, 2);

INSERT INTO acidexample partition (load\_date='2016-03-03') VALUES (3, 3);

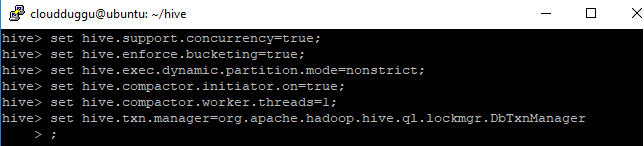
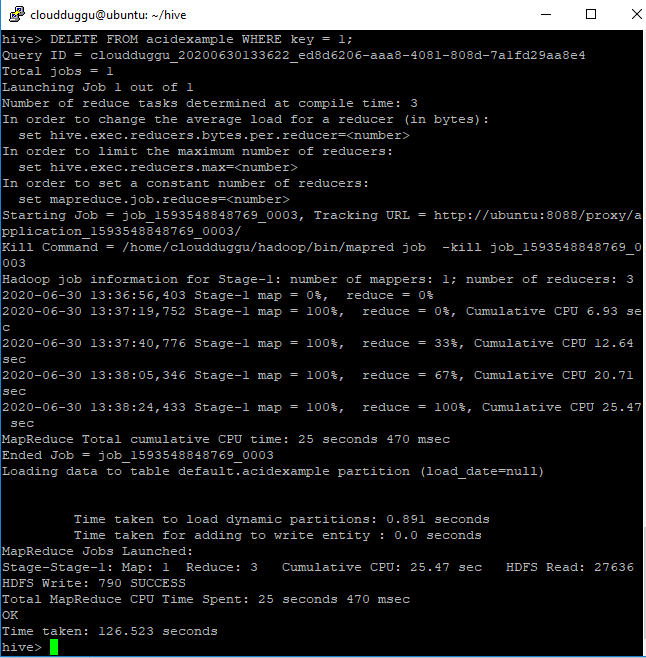
**Delete Command Syntax:**

DELETE FROM tablename [WHERE expression];

**Delete Command with where Statement:**

DELETE FROM acidexample WHERE key = 1;

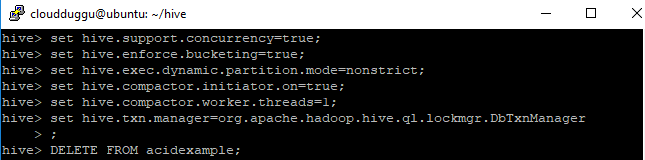
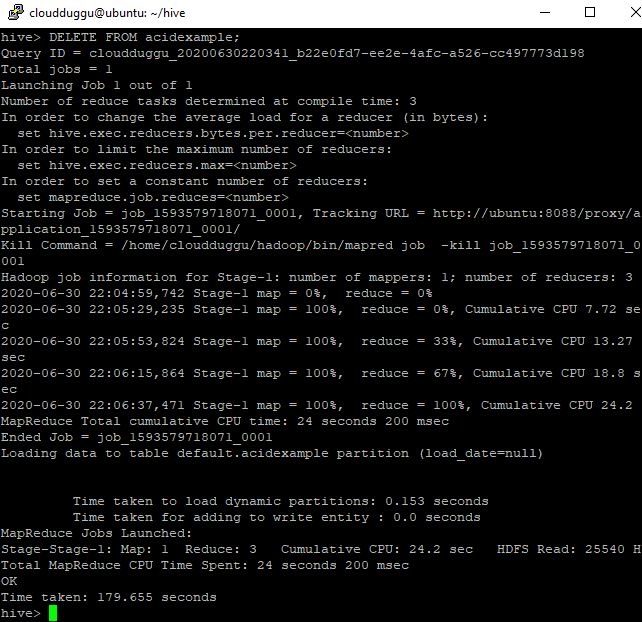
**Command Output:**

**Delete Statement:**

DELETE FROM acidexample;

**Command Output:**

  
  
  
  
  
  
**7. Update Command**

The Update command updates the existing records if where clause is supplied otherwise it will delete table data. We can’t perform update command on Partitioning and Bucketing columns.

**Note: In Apache Hive, We can perform an UPDATE statement on those tables which follow the ACID property.**

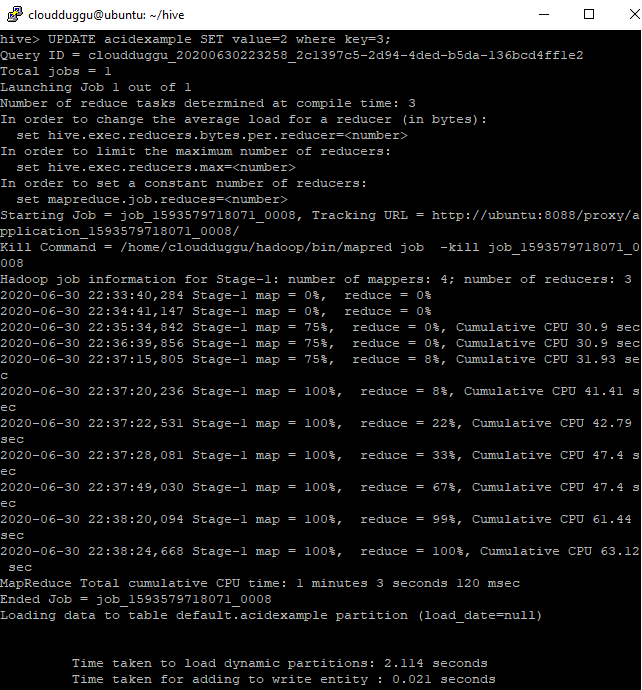
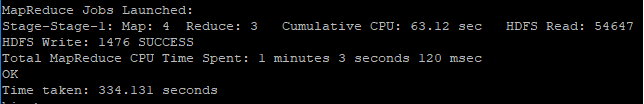
**Update Command Syntax:**

UPDATE tablename SET column = value [, column = value ...] [WHERE expression];

**Update Statement:**

UPDATE acidexample SET value=2 where key=3;

**Command Output:**

  
  
  
  
**8. Export Command**

The Apache Hive EXPORT command is used in case we need to exports the table's metadata to some other location.

To perform this activity we have created a directory “hive\_export\_location“in HDFS under /data/ hive\_export\_location and exporting table “acidexample”.

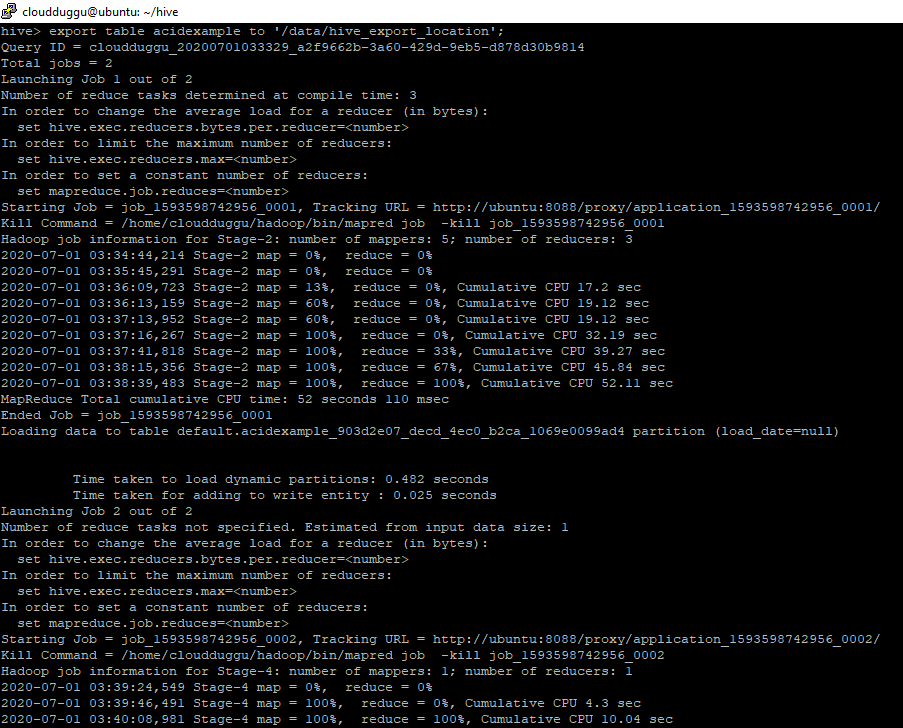
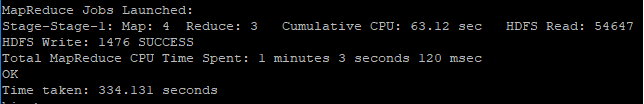
**Export Command Syntax:**

EXPORT TABLE tablename [PARTITION (part\_column="value"[, ...])]   
  TO 'export\_target\_path' [ FOR replication('eventid') ]

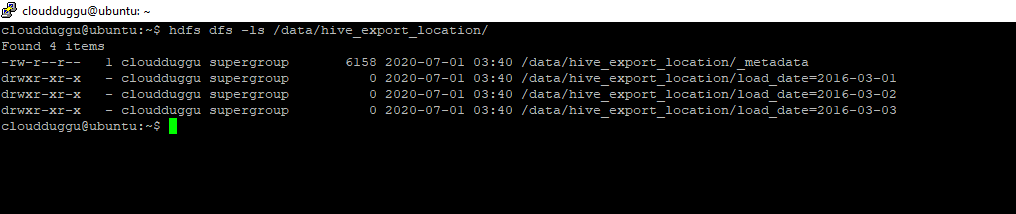
**Export Statement:**

export table acidexample to '/data/hive\_export\_location';

**Command Output:**

After export, we can verify \_metadata and data subdirectory of table “acidexample” in HDFS.



**9. Import Command**

Apache Hive IMPORT command imports the data from a specific location into Hive tables.

To perform this activity we will copy table “acidexample” data from HDFS location “data/ hive\_export\_location” into the “cloudduggudb” database.

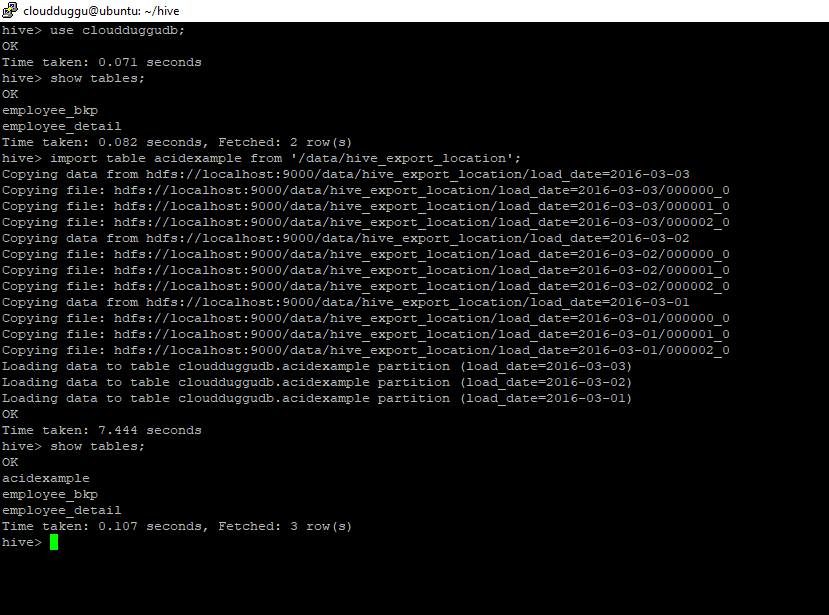
**Import Command Syntax:**

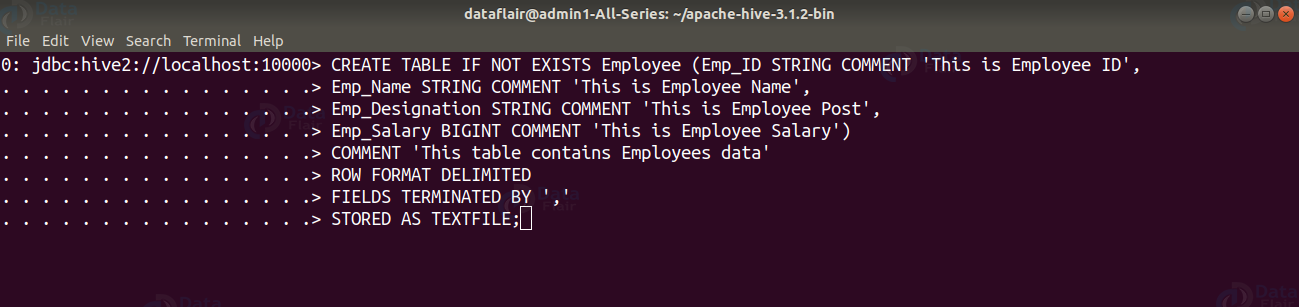
IMPORT [[EXTERNAL] TABLE new\_or\_original\_tablename [PARTITION (part\_column="value"[, ...])]]   
FROM 'source\_path'   
[LOCATION 'import\_target\_path']

**Import Statement:**

import table acidexample from '/data/hive\_export\_location';

**Command Output:**



[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/CREATETABLE-HiveDDLcommands14.png)

**ROW FORMAT DELIMITED** means we are telling the Hive that when it finds a new line character, that means a new record.

**FIELDS TERMINATED BY ‘,’** tells Hive what delimiter we are using in our files to separate each column.

**STORED AS TEXTFILE** is to tell Hive what type of file to expect.

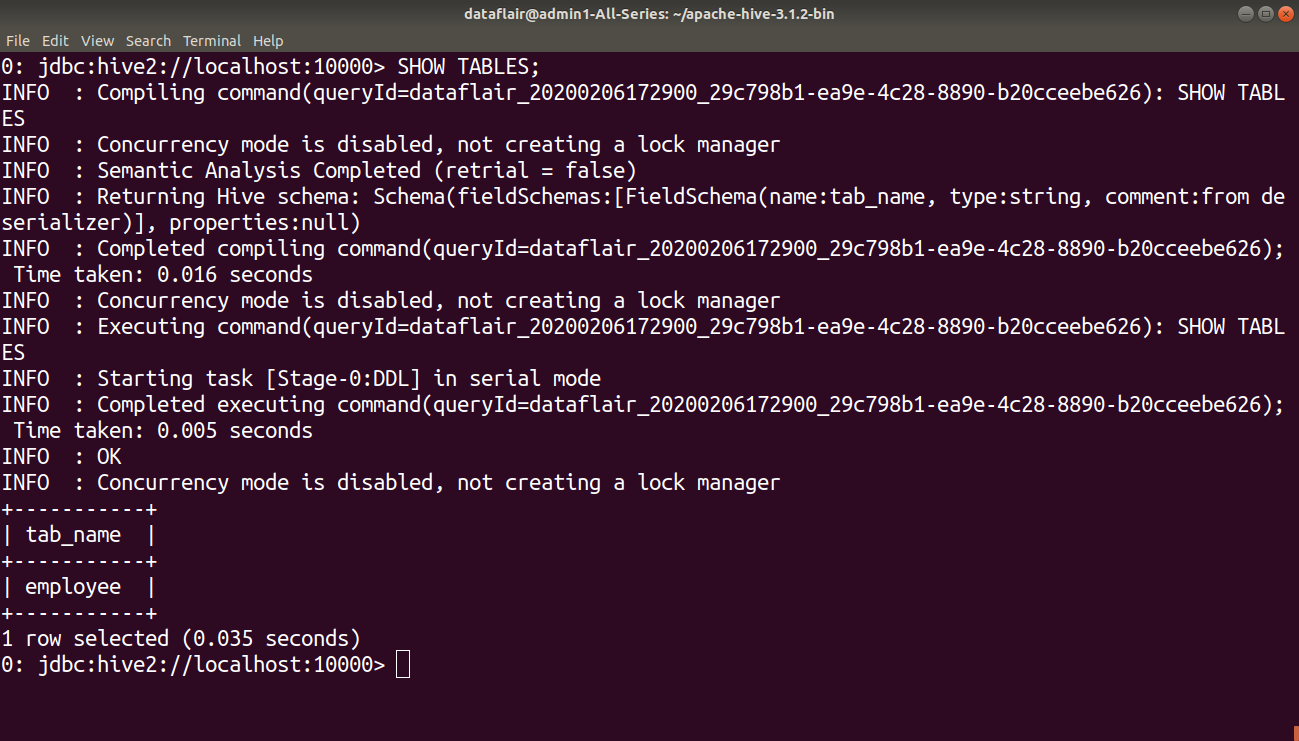
**2. SHOW TABLES in Hive**

The **SHOW TABLES** statement in Hive lists all the base tables and **views** in the current database.

**Syntax:**

SHOW TABLES [IN database\_name];

**DDL SHOW TABLES Example:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/SHOWTABLES-HiveDDLcommands15.png)

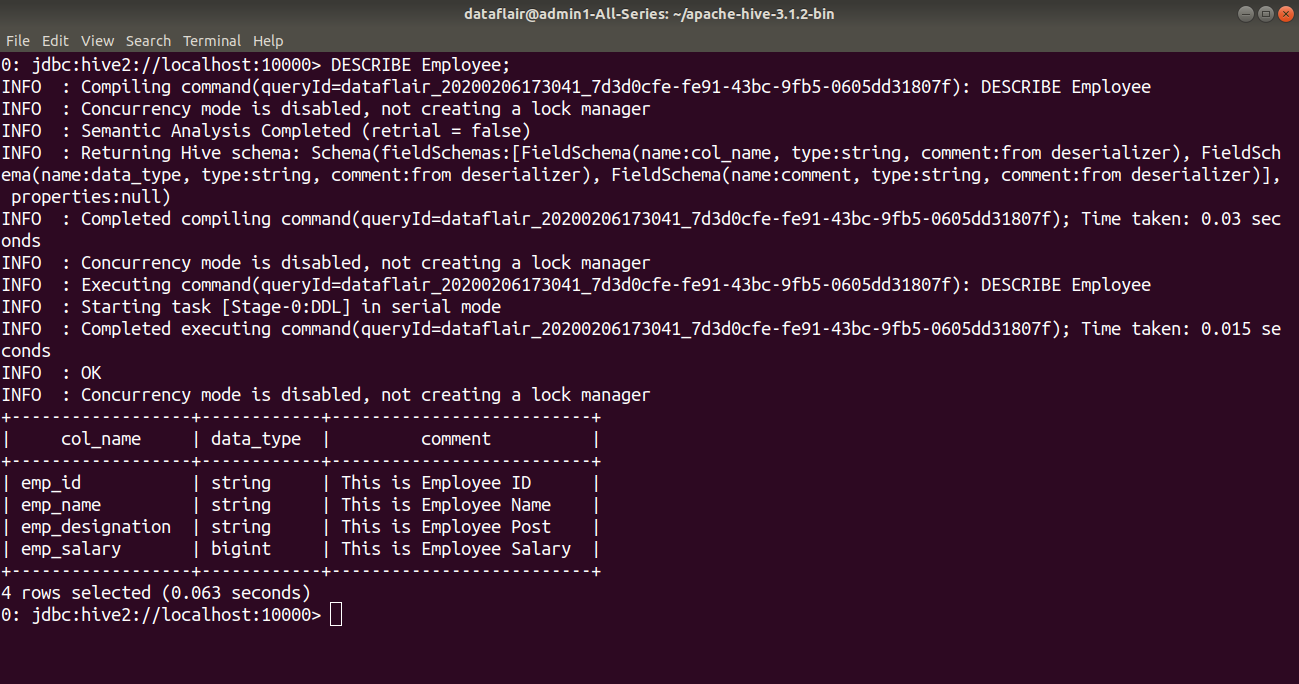
**3. DESCRIBE TABLE in Hive**

The **DESCRIBE** statement in Hive shows the lists of columns for the specified table.

**Syntax:**

DESCRIBE [EXTENDED|FORMATTED] [db\_name.] table\_name[.col\_name ( [.field\_name])];

**DDL DESCRIBE TABLE Example:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/DESCRIBETABLE-HiveDDLcommands16.png)

**4. DROP TABLE in Hive**

The **DROP TABLE** statement in Hive deletes the data for a particular table and remove all metadata associated with it from Hive metastore.

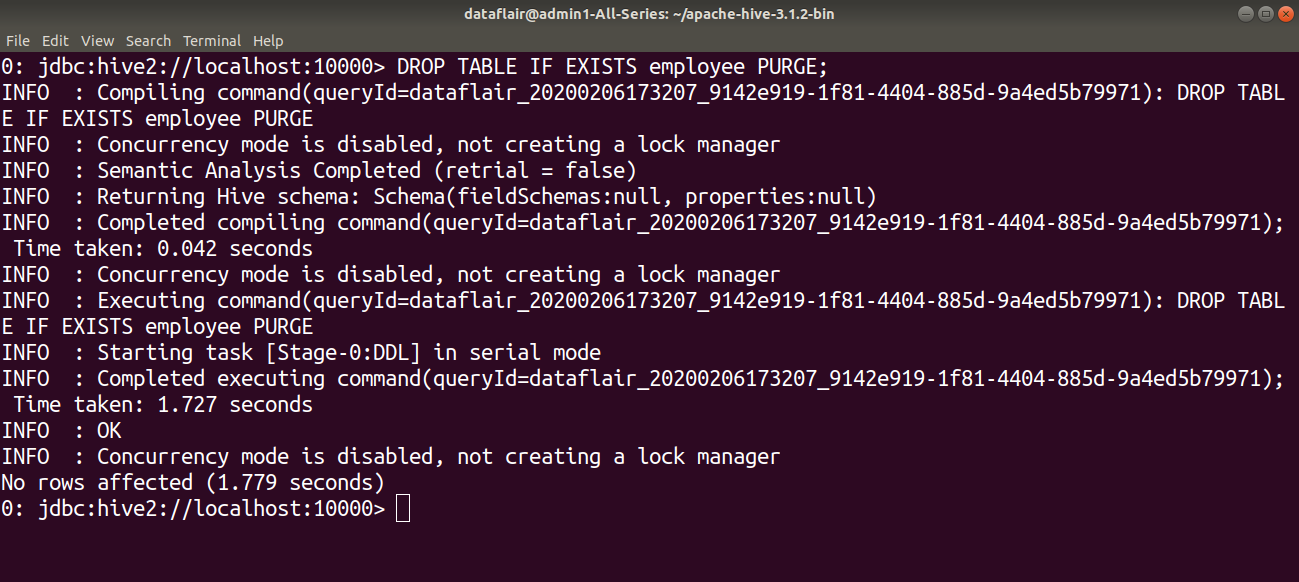
If **PURGE** is not specified then the data is actually moved to the .Trash/current directory. If **PURGE** is specified, then data is lost completely.

**Syntax:**

DROP TABLE [IF EXISTS] table\_name [PURGE];

**DDL DROP TABLE Example:**

In the below example, we are deleting the ‘employee’ table.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/DROPTABLE-HiveDDLcommands17.png)

**5. ALTER TABLE in Hive**

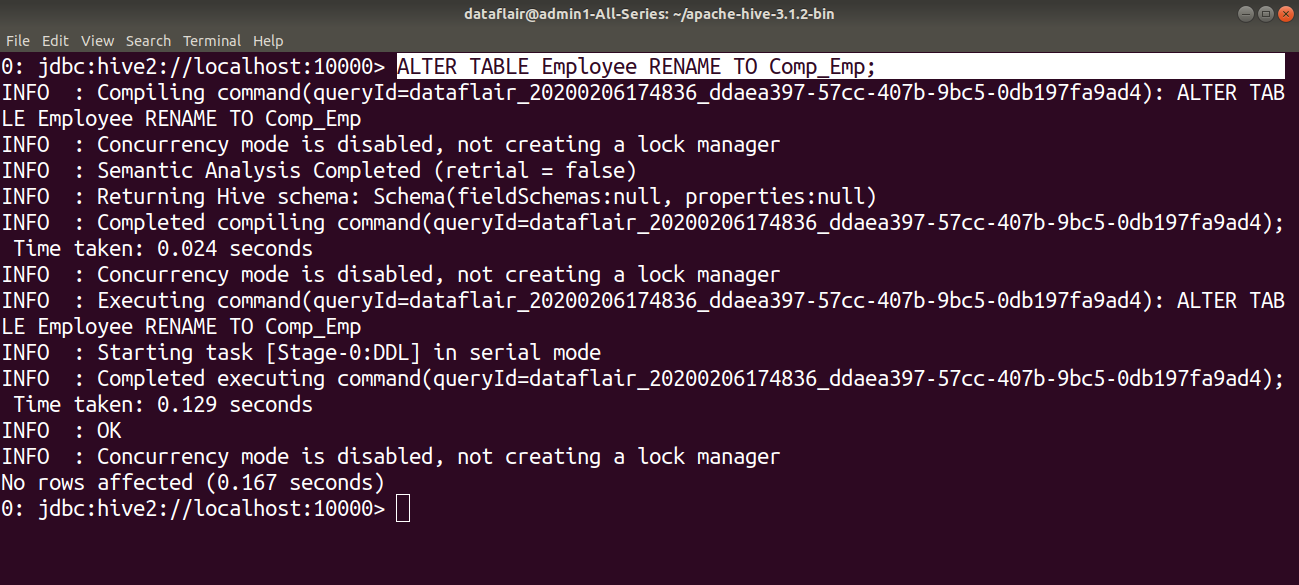
The **ALTER TABLE** statement in Hive enables you to change the structure of an existing table. Using the ALTER TABLE statement we can rename the table, add columns to the table, change the table properties, etc.

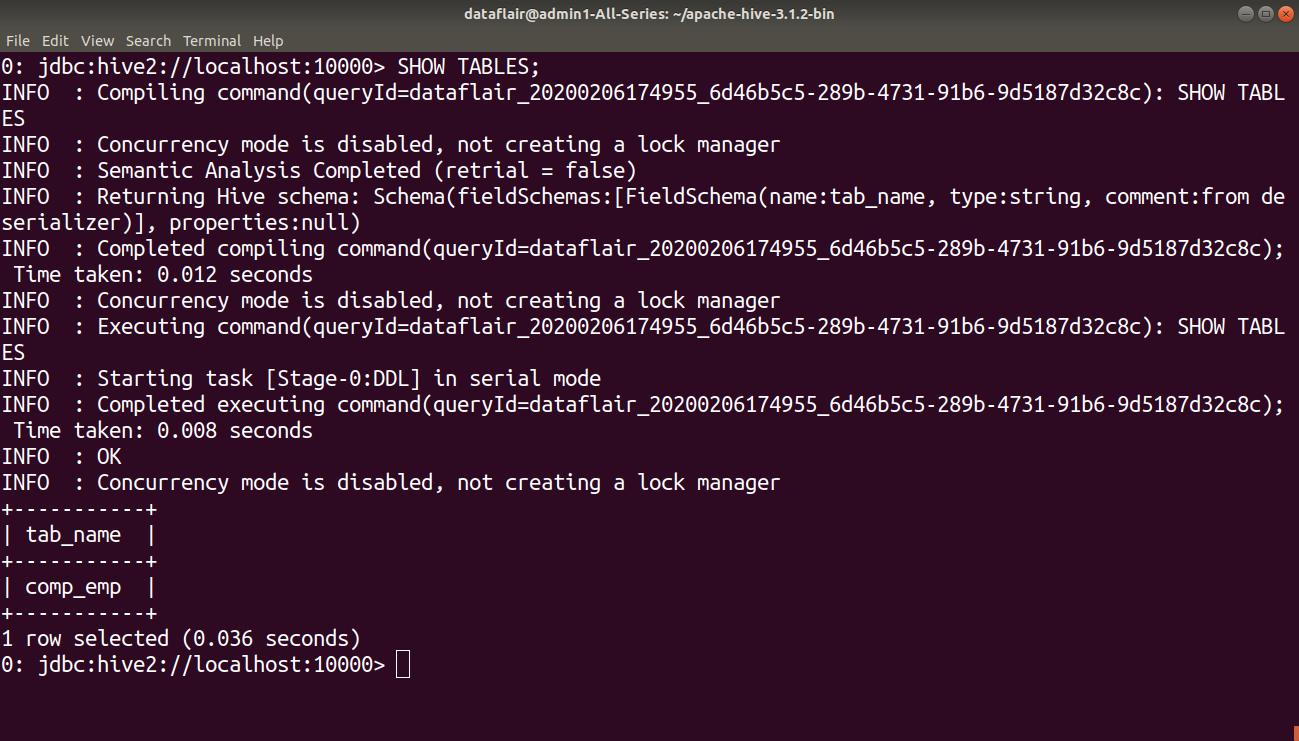
**Syntax to Rename a table:**

ALTER TABLE table\_name RENAME TO new\_table\_name;

**DDL ALTER TABLE name Example:**

In this example, we are trying to rename the ‘Employee’ table to ‘Com\_Emp’ using the ALTER statement.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/ALTERTABLERENAME-HiveDDLcommands18.png)

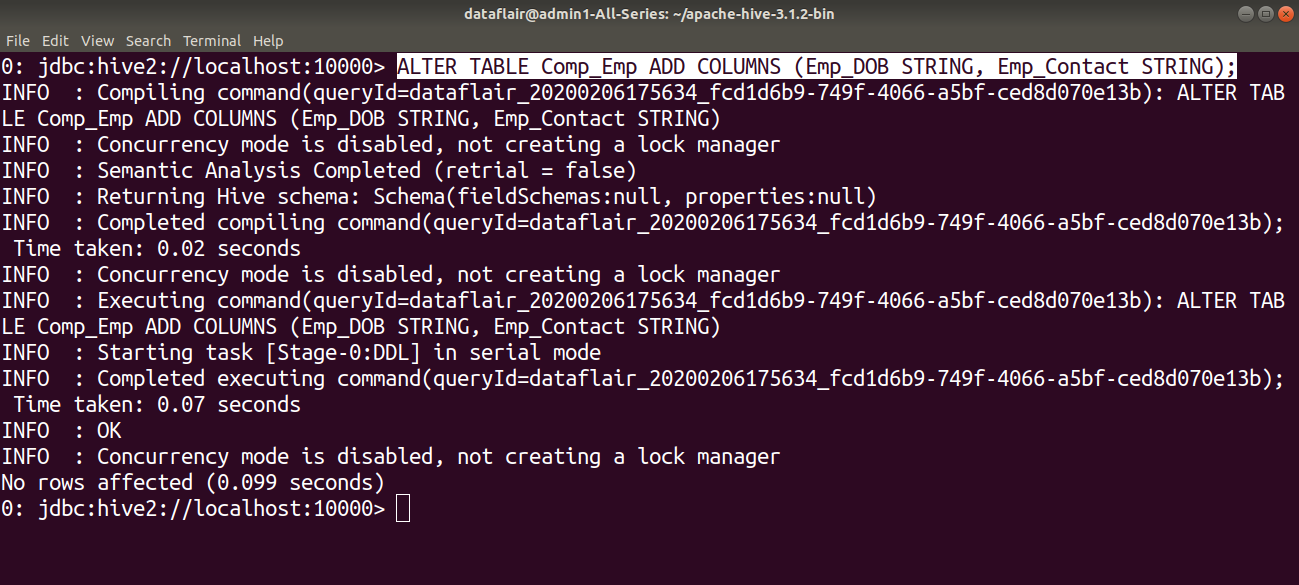
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/listingtableafterrename-HiveDDLcommands19.png)

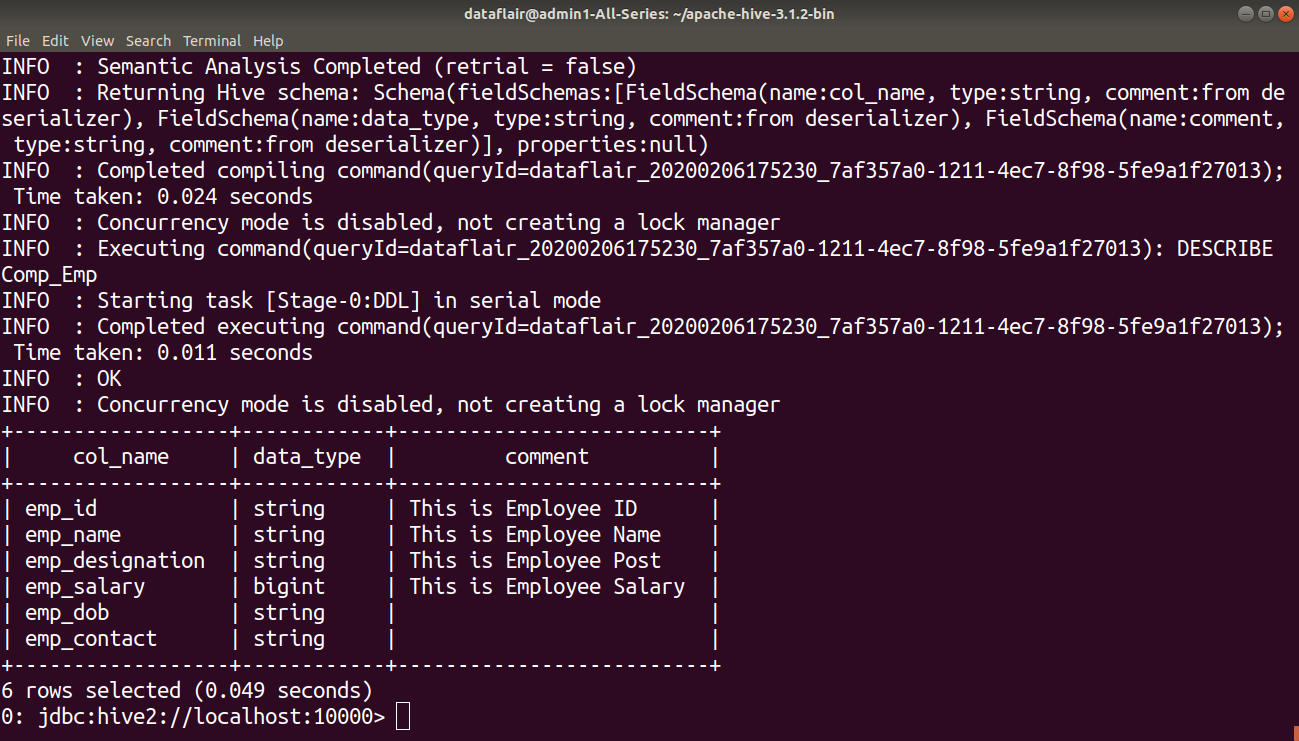
**Syntax to Add columns to a table:**

ALTER TABLE table\_name ADD COLUMNS (column1, column2) ;

**DDL ALTER TABLE columns Example:**

In this example, we are adding two columns ‘Emp\_DOB’ and ‘Emp\_Contact’ in the ‘Comp\_Emp’ table using the ALTER command.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/ALTERTABLEADDCOLUMN-HiveDDLcommands20.png)

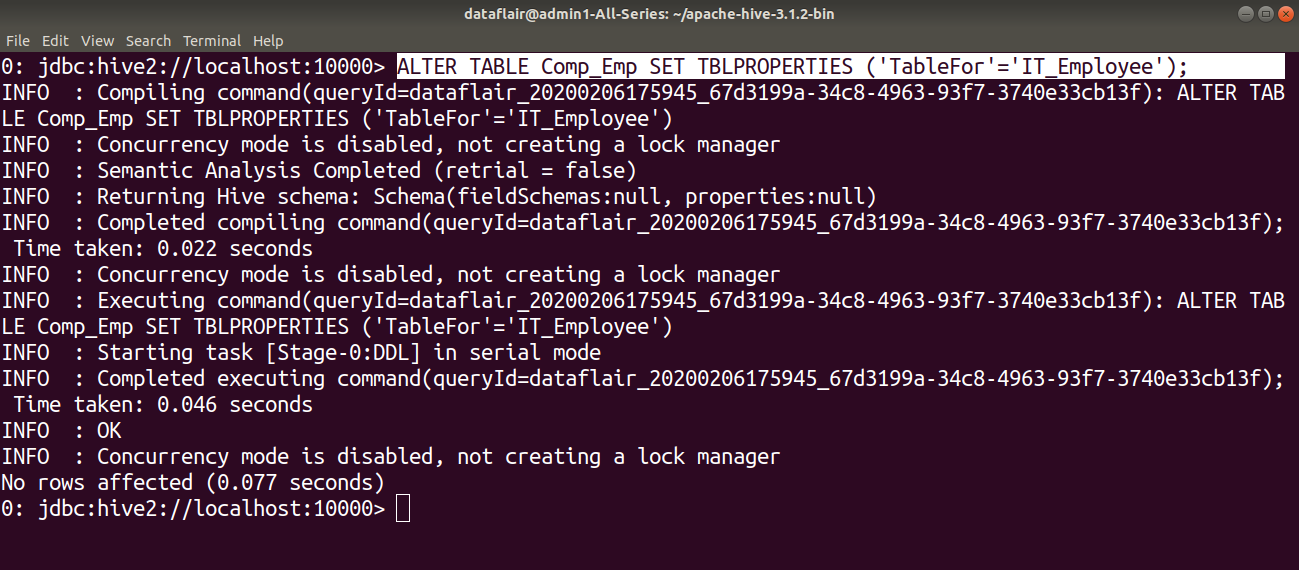
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/describingtableafterrename-HiveDDLcommands21.png)

**Syntax to set table properties:**

ALTER TABLE table\_name SET TBLPROPERTIES (‘property\_key’=’property\_new\_value’);

**DDL ALTER TABLE properties Example:**

In this example, we are setting the table properties after table creation by using ALTER command.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/ALTERTABLEPROPERTIES-HiveDDLcommands22.png)

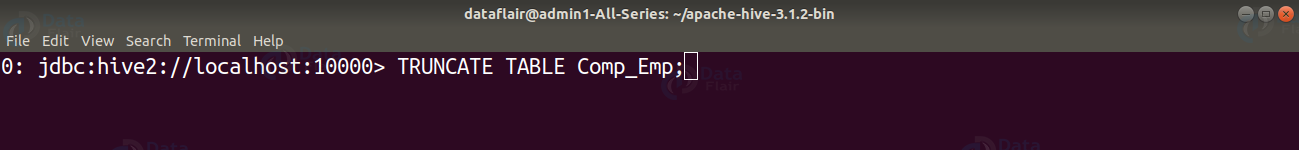
**6. TRUNCATE TABLE**

**TRUNCATE TABLE** statement in Hive removes all the rows from the table or partition.

**Syntax:**

TRUNCATE TABLE table\_name;

**DDL TRUNCATE TABLE Example:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/03/TRUNCATETABLE-HiveDDLcommands23.png)

**BDA**

**UNIT-IV**

**Introduction to Apache Pig**

Pig Represents Big Data as data flows. Pig is a high-level platform or tool which is used to process the large datasets. It provides a high-level of abstraction for processing over the MapReduce. It provides a high-level scripting language, known as Pig Latin which is used to develop the data analysis codes. First, to process the data which is stored in the HDFS, the programmers will write the scripts using the Pig Latin Language. Internally Pig Engine(a component of Apache Pig) converted all these scripts into a specific map and reduce task. But these are not visible to the programmers in order to provide a high-level of abstraction. Pig Latin and Pig Engine are the two main components of the Apache Pig tool. The result of Pig always stored in the HDFS.

**Need of Pig:** One limitation of MapReduce is that the development cycle is very long. Writing the reducer and mapper, compiling packaging the code, submitting the job and retrieving the output is a time-consuming task. Apache Pig reduces the time of development using the multi-query approach. Also, Pig is beneficial for programmers who are not from [Java](https://www.geeksforgeeks.org/java/) background. 200 lines of Java code can be written in only 10 lines using the Pig Latin language. Programmers who have SQL knowledge needed less effort to learn Pig Latin.

* It uses query approach which results in reducing the length of the code.
* Pig Latin is SQL like language.
* It provides many builtIn operators.
* It provides nested data types (tuples, bags, map).

**Evolution of Pig:** Earlier in 2006, Apache Pig was developed by Yahoo’s researchers. At that time, the main idea to develop Pig was to execute the MapReduce jobs on extremely large datasets. In the year 2007, it moved to Apache Software Foundation(ASF) which makes it an open source project. The first version(0.1) of Pig came in the year 2008. The latest version of Apache Pig is 0.18 which came in the year 2017.

**Features of Apache Pig:**

* For performing several operations Apache Pig provides rich sets of operators like the filtering, joining, sorting, aggregation etc.
* Easy to learn, read and write. Especially for SQL-programmer, Apache Pig is a boon.
* Apache Pig is extensible so that you can make your own process and  user-defined functions(UDFs) written in python, java or other programming languages .
* Join operation is easy in Apache Pig.
* Fewer lines of code.
* Apache Pig allows splits in the pipeline.
* By integrating with other components of the Apache Hadoop ecosystem, such as Apache Hive, Apache Spark, and Apache ZooKeeper, Apache Pig enables users to take advantage of these components’ capabilities while transforming data.
* The data structure is multivalued, nested, and richer.
* Pig can handle the analysis of both structured and unstructured data.

**Difference between Pig and MapReduce**

| **Apache Pig** | **MapReduce** |
| --- | --- |
| It is a scripting language. | It is a compiled programming language. |
| Abstraction is at higher level. | Abstraction is at lower level. |
| It have less line of code as compared to MapReduce. | Lines of code is more. |
| Less effort is needed for Apache Pig. | More development efforts are required for MapReduce. |
| Code efficiency is less as compared to MapReduce. | As compared to Pig efficiency of code is higher. |
| Pig provides built in functions for ordering, sorting and union. | Hard to perform data operations. |
| It allows nested data types like map, tuple and bag | It does not allow nested data types |

**Applications of Apache Pig:**

* For exploring large datasets Pig Scripting is used.
* Provides the supports across large data-sets for Ad-hoc queries.
* In the prototyping of large data-sets processing algorithms.
* Required to process the time sensitive data loads.
* For collecting large amounts of datasets in form of search logs and web crawls.
* Used where the analytical insights are needed using the sampling.

**Types of Data Models in Apache Pig:** It consist of the 4 types of data models as follows:

* **Atom**: It is a atomic data value which is used to store as a string. The main use of this model is that it can be used as a number and as well as a string.
* **Tuple**: It is an ordered set of the fields.
* **Bag**: It is a collection of the tuples.
* **Map**: It is a set of key/value pairs.

**Running Pig**

**Apache Pig Run Modes**

Apache Pig executes in two modes: Local Mode and MapReduce Mode.



**Local Mode**

* It executes in a single JVM and is used for development experimenting and prototyping.
* Here, files are installed and run using localhost.
* The local mode works on a local file system. The input and output data stored in the local file system.

The command for local mode grunt shell:

1. $ pig-x local

**MapReduce Mode**

* The MapReduce mode is also known as Hadoop Mode.
* It is the default mode.
* In this Pig renders Pig Latin into MapReduce jobs and executes them on the cluster.
* It can be executed against semi-distributed or fully distributed Hadoop installation.
* Here, the input and output data are present on HDFS.

The command for Map reduce mode:

1. $ pig

Or,

1. $ pig -x mapreduce

**Ways to execute Pig Program**

These are the following ways of executing a Pig program on local and MapReduce mode: -

* **Interactive Mode** - In this mode, the Pig is executed in the Grunt shell. To invoke Grunt shell, run the pig command. Once the Grunt mode executes, we can provide Pig Latin statements and command interactively at the command line.
* **Batch Mode** - In this mode, we can run a script file having a .pig extension. These files contain Pig Latin commands.
* **Embedded Mode** - In this mode, we can define our own functions. These functions can be called as UDF (User Defined Functions). Here, we use programming languages like Java and Python.

**Apache Pig Execution Modes**

You can run Apache Pig in two modes, namely, **Local Mode** and **HDFS mode**.

**Local Mode**

In this mode, all the files are installed and run from your local host and local file system. There is no need of Hadoop or HDFS. This mode is generally used for testing purpose.

**MapReduce Mode**

MapReduce mode is where we load or process the data that exists in the Hadoop File System (HDFS) using Apache Pig. In this mode, whenever we execute the Pig Latin statements to process the data, a MapReduce job is invoked in the back-end to perform a particular operation on the data that exists in the HDFS.

**Apache Pig Execution Mechanisms**

Apache Pig scripts can be executed in three ways, namely, interactive mode, batch mode, and embedded mode.

* **Interactive Mode** (Grunt shell) − You can run Apache Pig in interactive mode using the Grunt shell. In this shell, you can enter the Pig Latin statements and get the output (using Dump operator).
* **Batch Mode** (Script) − You can run Apache Pig in Batch mode by writing the Pig Latin script in a single file with **.pig** extension.
* **Embedded Mode** (UDF) − Apache Pig provides the provision of defining our own functions (**U**ser **D**efined **F**unctions) in programming languages such as Java, and using them in our script.

**Invoking the Grunt Shell**

You can invoke the Grunt shell in a desired mode (local/MapReduce) using the **−x** option as shown below.

|  |  |
| --- | --- |
| **Local mode** | **MapReduce mode** |
| **Command −**  $ ./pig –x local | **Command −**  $ ./pig -x mapreduce |
| **Output** −  Local Mode Output | **Output** −  MapReduce Mode Output |

Either of these commands gives you the Grunt shell prompt as shown below.

grunt>

You can exit the Grunt shell using **‘ctrl + d’.**

After invoking the Grunt shell, you can execute a Pig script by directly entering the Pig Latin statements in it.

grunt> customers = LOAD 'customers.txt' USING PigStorage(',');

**Executing Apache Pig in Batch Mode**

You can write an entire Pig Latin script in a file and execute it using the **–x command**. Let us suppose we have a Pig script in a file named **sample\_script.pig** as shown below.

**Sample\_script.pig**

student = LOAD 'hdfs://localhost:9000/pig\_data/student.txt' USING

PigStorage(',') as (id:int,name:chararray,city:chararray);

Dump student;

Now, you can execute the script in the above file as shown below.

|  |  |
| --- | --- |
| **Local mode** | **MapReduce mode** |
| $ pig -x local **Sample\_script.pig** | $ pig -x mapreduce **Sample\_script.pig** |

**Getting started with Apache Pig!**

**Introduction**

After reading the heading Apache Pig, the first question that hits every mind is, why the word Pig? Apache Pig is capable of working on any kind of data, similar to a pig who can eat anything. Pig is nothing but a high-level extensible programming language designed to analyze bulk data sets and to reduce the complexities of coding MapReduce programs. Yahoo developed Pig to analyze huge [unstructured data](https://www.analyticsvidhya.com/blog/2020/12/what-i-did-when-i-had-to-work-with-unstructured-data/) sets and minimize the writing time of Mapper and Reducer functions.

**What is Apache Pig?**

Apache Pig is an abstraction over MapReduce that is used to handle structured, semi-structured, and unstructured data. It is a high-level data flow tool developed to execute queries on large datasets that are stored in HDFS. Pig Latin is the high-level scripting language used by Apache Pig to write data analysis programs. For reading, writing, and processing data, it provides multiple operators that can easily be used by developers. These pig scripts get internally converted to Map and Reduce tasks and get executed on data available in HDFS. Generally, a component of Apache Pig called Pig Engine is responsible for converting the scripts into MapReduce jobs.

**Why do we Need Apache Pig?**

Usually, programmers struggle while performing any MapReduce tasks, as they are not so good at Java to work with Hadoop. In such cases, Pig works as a situation booster to all such programmers. Here are some reasons that make Pig a must-use platform:

* While using Pig Latin, programmers can skip typing the complex codes in Java and perform MapReduce tasks with ease.
* The multi-query approach is used by Pig to reduce the length of codes. For example, Instead of typing 200 lines of code (LoC) in Java, programmers can use Apache Pig to write just 10 LoC.
* Pig Latin is easy to understand as it is a SQL-like language.
* Pig is also known as an operator-rich language because it offers multiple built-in operators like joins, filters, ordering, etc.

**Features of Apache Pig**

Following are the features of Apache Pig:-

1. **Extensibility**

Pig is an extensible language that means, with the help of its existing operators, users can build their own functions to read, write, and process data.

1. **Optimization Opportunities**

The tasks encoded in Apache Pig allow the system to optimize their execution automatically, so the users can focus only on the semantics of the language rather than efficiency.

1. **UDFs**

UDFs stand for user-defined functions; Pig provides the facility to create them in other programming languages like Java and embed them in Pig Scripts.

1. **Ease of programming**

It is difficult for non-programmers to write the complex java programs for map-reduce, but using Pig Latin, they can easily perform queries.

1. **Rich operator set**

Pig Latin has multiple operator support like join, sort, filter, etc.

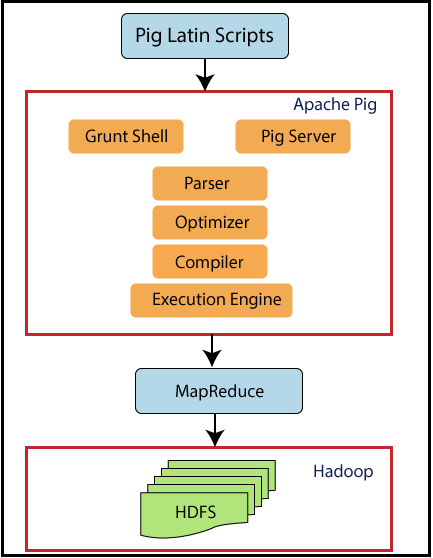
**Difference between Apache Pig and Map Reduce**

Although Apache Pig is an abstraction over MapReduce, their overlapping functions make it difficult to differentiate them. Apache Pig is related to the MapReduce tasks but works in an entirely different manner. Here are some major differences:

* Apache Pig is a user-friendly high-level data-flow language, while MapReduce is just a low-level paradigm for data processing.
* Apache Pig doesn’t require any compilation process, but MapReduce operations need a significant compilation process.
* Join task in Pig can be performed much more smoothly and efficiently than MapReduce.
* The multi-query functionality of Apache Pig enables to write very few lines of code and makes the operation more efficient, while MapReduce doesn’t support this feature. In comparison to Pig, MapReduce needs to write 20 times more lines of code to perform the same operation.
* Basic knowledge of SQL is enough for working with Pig, but a deep understanding of Java concepts is required to work with MapReduce.

**Architecture and Components of Apache Pig**

Firstly, we submit Pig scripts to the execution environment of Apache Pig which can be written in Pig Latin using in-built operators. The Pig scripts undergo various transformations in multiple stages to generate the desired output. Let’s discuss each phase separately.



source: https://www.tutorialandexample.com/apache-pig-architecture

**First Stage: Parser**

At first, when a Pig Latin script is sent to Hadoop Pig, it is handled by the Parser. Basically, the parser is responsible for various types of checks on the script, like type checks, syntax checks, and other miscellaneous checks. Afterwards, Parser gives an output in the form of a Directed Acyclic Graph (DAG), which carries the logical operators and Pig Latin statements. In the logical plan(DAG), the logical operators of the script are represented as the nodes, and the data flows are represented as edges.

**Second Stage: Optimizer**

After retrieving the output from the parser, a logical plan for DAG is submitted to a logical optimizer. The logical optimizations are carried out by the optimizer, which includes activities like transform, split, merge, reorder operators, etc. The optimizer basically aims to reduce the quantity of data in the pipeline when it processes the extracted data. This optimizer performs automatic optimization of the data and uses various functions like

* **PushUpFilter:** If multiple conditions are available in a filter and the filter can be split, then Pig pushes up each condition individually and splits those conditions. An earlier selection of these conditions is helpful by resulting in the reduction of the number of records left in the pipeline.
* **LimitOptimizer:** If the limit operator is applied just after a load or sort operator, then Pig converts these operators into a limit-sensitive implementation, which omits the processing of the whole data set.
* **ColumnPruner:** This function will omit the columns that are never used; hence, it reduces the size of the record. This function can be applied after each operator to prune the fields aggressively and frequently.
* **MapKeyPruner:** This function will omit the map keys that are never used, hence, reducing the size of the record.

**Third Stage: Compiler**

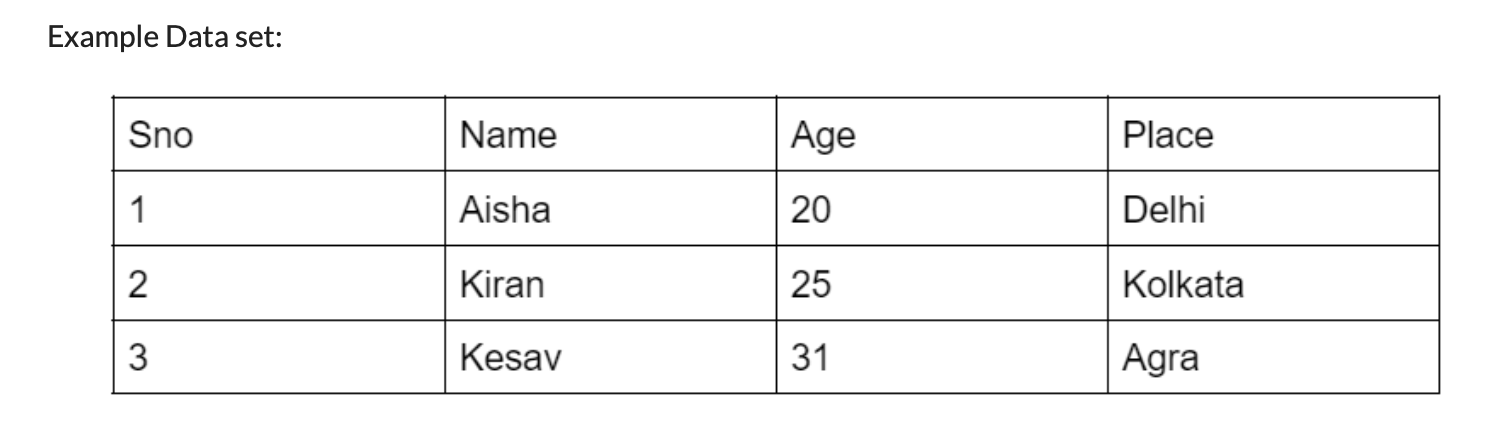
After receiving the optimizer’s output, the Compiler compiles the resultant code into a series of MapReduce tasks. The Compiler is responsible for the conversion of Pig Script into MapReduce jobs.

**Fourth Stage: Execution Engine**

At last, come to the Execution Engine, where the MapReduce jobs are transferred for execution to the Hadoop. Then the MapReduce jobs get executed, and Hadoop provides the required results. Output can be displayed on the screen by using the ‘DUMP’ statement and can be stored in the HDFS by the ‘STORE’ statement.

**Pig Latin Data Model**

The data model of Pig Latin allows it to handle a variety of data. Pig Latin can handle simple atomic data types such as int, float, long, double, etc., as well as complex non-atomic data types such as map, tuple, and bag.

Example Data set:

**Atom**

Atom is a scaler primitive data type that can be any single value in Pig Latin, irrespective of their data type. The atomic values of Pig can be string, int, long, float, double, char array, and byte array. A simple atomic value or a byte of data is known as a field.

**Example of an atom** − ‘2’, ‘Kiran,’ ‘25’, ‘Kolkata,’ etc.

**Tuple**

A tuple is a record that is formed by an ordered set of fields that may carry different data types for each field. A tuple can be compared with the records stored in a row in an RDBMS. It is not mandatory to have a schema attached with the elements present inside a tuple. Small brackets ‘()’ are used to represent the tuples.

**Example of tuple** − (2, Kiran, 25, Kolkata)

**Bag**

An unordered set of tuples is known as a bag. Basically, a bag is a collection of tuples that are not mandatory to be unique. Curly braces ‘{}’ are used to represent the bag in a data model. Bag supports a flexible schema, i.e., each tuple can have any number of fields. A bag is much more similar to a table in RDBMS, but unlike a table in RDBMS, it is not mandatory that the same number of fields are present in a tuple or that the fields in the same position have the same data type.

**Example of a bag** − {(Kiran, 25, Kolkata), (Aisha, 20), (Ketki, Agra)}

**Map**

A map is nothing but a set of key-value pairs used to represent the data elements. The key should be unique and must be of the type char array, whereas the value can be of any type. Square brackets ‘[]’ are used to represent the Map, and the hash ‘#’ symbol is used to separate the key-value pair.

**Example of maps**− [name#Kiran, age#25 ], [name#Aisha, age#20 ]

**Introduction to Pig Latin commands**

To communicate with Pig, a very powerful tool is used called Grunt Shell. A grunt shell is an interactive shell that establishes an interaction of the shell with HDFS and the local file system. We can open any remote client access software like putty to start the Cloudera and type Pig to enter in the Grunt Shell. Grunt Shell allows you to write Pig Latin statements and queries the structured/unstructured data.

To start Pig Grunt type:

Following are the basic and intermediate Pig Latin operations:

**Reading Data**

To read data in Pig, we need to put the data from the local file system to Hadoop. Let’s see the steps:

**Step 1:-** Create a file using the cat command in the local file system.

**Step 2:-** Transfer the file into the HDFS using the put command.

**Step 3:-** Read the data from the Hadoop to the Pig Latin using the load command.

**Syntax:-**

Relation = LOAD 'Input file path information' USING load\_function AS schema;

Where,

* Relation − We have to provide the relation name where we want to load the file content.
* Input file path information − We have to provide the path of the Hadoop directory where the file is stored.
* load\_function − Apache Pig provides a variety of load functions like BinStorage, JsonLoader, PigStorage, TextLoader. Here, we need to choose a function from this set. PigStorage is the commonly used function as it is suited for loading structured text files.
* Schema − We need to define the schema of the data or the passing files in parenthesis.

**Running Pig Statements**

To run the Pig Latin statements, we use the Dump operator. It will display the results on the screen and can be used for debugging purposes.

**Syntax:**

grunt> Dump Relation\_Name

**Describe Operator**

To display the schema of the relation, describe operator is used.

Syntax:

grunt> Describe Relation\_name

**Illustrate Operator**

To get the step-by-step execution of a sequence of statements in Pig command, illustrate operator is used.

Syntax:

grunt> illustrate Relation\_name;

**Explain**

To review the logical, physical, and map-reduce execution plans of a relation, explain operator is used.

Syntax:

grunt> explain Relation\_name;

**Filter**

To select the required tuples from a relation depending upon a condition, the FILTER operator is used.

Syntax:

grunt> Relation2 = FILTER Relation1 BY (condition);

**Limit**

To get a limited number of tuples from a relation, Pig supports the limit operator.

Syntax:

grunt> Output = LIMIT Relation\_name number of tuples required ;

**Distinct**

To remove duplicate tuples from a relation, the distinct operator is used.

Syntax:-

grunt> Relation2 = DISTINCT Relation1;

**Foreach**

To generate specified data transformations based on the column data, the Foreach operator is required.

Syntax:

grunt> Relation2 = FOREACH Relation1 GENERATE (required data);

**Group**

To group the data in one or more relations, the Group operator is used. It groups the data with the same key.

Syntax:

grunt> Group\_data = GROUP Relation\_name BY key\_column;

Grouping by multiple columns

Group operators can also group the data in one or more relations using multiple columns.

Syntax:

grunt> Group\_data = GROUP Relation\_name BY (column1, column2,column3,..);

**Group All**

To group a relation by all the columns, the Group All operator is used.

Syntax:

grunt> group\_all\_data = GROUP Relation\_name All;

**Cogroup**

The Cogroup operator is much more similar to the group operator. The major difference is the cogroup is more suited for multiple relations, whereas the group is more suitable for single relations.

Syntax:

grunt> cogroup\_data = COGROUP Relation1 by column1, Relation2 by column2;

**Join**

The purpose of the Join operator is to combine data from two or more relations. Firstly, we have two declare the keys, which are nothing but a tuple from each relation. If the keys are matched with each other, then we consider that two particular tuples are matched and can be displayed in the output; otherwise, the unmatched records are dropped. Following are the types of Joins−

* Self Join
* Inner Join
* Left outer join
* Right outer join
* Full outer join

Let’s understand each type of join.

**Self Join**

When we have to join a table with itself, or we treat a single table as two separate relations, then self-join is used. We load the same data multiple times with different alias names to perform the join operation.

Syntax:

grunt> Relation3 = JOIN Relation1 BY key, Relation2 BY key;

**Inner Join**

The most commonly used join is the inner join(also known as equijoin). An inner join compares both the tables(say A and B) and returns rows when there is a match. Using the join predicate combines the column values and creates a new relation.

Syntax:

grunt> Output = JOIN Relation1 BY column1, Relation2 BY column2;

**Left Outer Join**

The left outer join compares two relations, the left and the right relation, and returns all the rows from the left relation, even if doesn’t match with the right relation.

Syntax:

grunt> Relation3 = JOIN Relation1 BY id LEFT OUTER, Relation2 BY column;

**Right Outer Join**

The right outer join compares two relations, the left and the right relations, and returns all the rows from the right relation, even if doesn’t match with the left relation.

Syntax:

grunt> Relation3 = JOIN Relation1 BY id RIGHT OUTER, Relation2 BY column;

**Full Outer Join**

The full outer Join compares two relations, the left and the right relation, and returns rows when there is a match in one of the relations.

Syntax:

grunt> Relation3 = JOIN Relation1 BY id FULL OUTER, Relation2 BY column;

**Cross**

To calculate the cross-product of two or more relations, the cross operator is used.

Syntax:

grunt> Relation3 = CROSS Relation1, Relation2;

**Union**

To merge the content of two relations, a union operator is used. The necessary condition to perform merging is the columns and domains of both the relations must be identical.

Syntax:

grunt> Relation3 = UNION Relation1, Relation2;

**Split**

To divide a relation into two or more relations, the split operator is required.

Syntax:

grunt> SPLIT Relation1 INTO Relation2 IF (condition1), Relation2 (condition2);

**Conclusion**

In this guide, we learned about the Apache Pig, which analyzes any type of data present in HDFS.

* We discussed Pig, its features, architecture, and its components.
* In this guide, we also discussed how to interact with the Grunt shell and perform various Linux-based commands.
* We also made a comparison of Pig with MapReduce.
* We don’t have to install the Grunt shell explicitly. Instead, we can open the Cloudera to run these Linux-based Pig commands.

**Working with Operators in Pig**

Pig Latin is the language used to analyze data in Hadoop using Apache Pig. In this chapter, we are going to discuss the basics of Pig Latin such as Pig Latin statements, data types, general and relational operators, and Pig Latin UDF’s.

**Pig Latin – Data Model**

As discussed in the previous chapters, the data model of Pig is fully nested. A **Relation** is the outermost structure of the Pig Latin data model. And it is a **bag** where −

* A bag is a collection of tuples.
* A tuple is an ordered set of fields.
* A field is a piece of data.

**Pig Latin – Statemets**

While processing data using Pig Latin, **statements** are the basic constructs.

* These statements work with **relations**. They include **expressions** and **schemas**.
* Every statement ends with a semicolon (;).
* We will perform various operations using operators provided by Pig Latin, through statements.
* 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 MapReduce job for loading the data into the file system will be carried out.

**Example**

Given below is a Pig Latin statement, which loads data to Apache Pig.

grunt> Student\_data = LOAD 'student\_data.txt' USING PigStorage(',')as

( id:int, firstname:chararray, lastname:chararray, phone:chararray, city:chararray );

**Pig Latin – Data types**

Given below table describes the Pig Latin data types.

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Data Type** | **Description & Example** |
| 1 | Int | Represents a signed 32-bit integer.  **Example** : 8 |
| 2 | Long | Represents a signed 64-bit integer.  **Example** : 5L |
| 3 | Float | Represents a signed 32-bit floating point.  **Example** : 5.5F |
| 4 | Double | Represents a 64-bit floating point.  **Example** : 10.5 |
| 5 | Chararray | Represents a character array (string) in Unicode UTF-8 format.  **Example** : ‘tutorials point’ |
| 6 | Bytearray | Represents a Byte array (blob). |
| 7 | Boolean | Represents a Boolean value.  **Example** : true/ false. |
| 8 | Datetime | Represents a date-time.  **Example** : 1970-01-01T00:00:00.000+00:00 |
| 9 | Biginteger | Represents a Java BigInteger.  **Example** : 60708090709 |
| 10 | Bigdecimal | Represents a Java BigDecimal  **Example** : 185.98376256272893883 |
| **Complex Types** | | |
| 11 | Tuple | A tuple is an ordered set of fields.  **Example** : (raja, 30) |
| 12 | Bag | A bag is a collection of tuples.  **Example** : {(raju,30),(Mohhammad,45)} |
| 13 | Map | A Map is a set of key-value pairs.  **Example** : [ ‘name’#’Raju’, ‘age’#30] |

**Null Values**

Values for all the above data types can be NULL. Apache Pig treats null values in a similar way as SQL does.

A null can be an unknown value or a non-existent value. It is used as a placeholder for optional values. These nulls can occur naturally or can be the result of an operation.

**Pig Latin – Arithmetic Operators**

The following table describes the arithmetic operators of Pig Latin. Suppose a = 10 and b = 20.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | **Addition** − Adds values on either side of the operator | a + b will give 30 |
| − | **Subtraction** − Subtracts right hand operand from left hand operand | a − b will give −10 |
| \* | **Multiplication** − Multiplies values on either side of the operator | a \* b will give 200 |
| / | **Division** − Divides left hand operand by right hand operand | b / a will give 2 |
| % | **Modulus** − Divides left hand operand by right hand operand and returns remainder | b % a will give 0 |
| ? : | **Bincond** − Evaluates the Boolean operators. It has three operands as shown below.  variable **x** = (expression) ? **value1** if true : **value2** if false. | b = (a == 1)? 20: 30;  if a = 1 the value of b is 20.  if a!=1 the value of b is 30. |
| CASE  WHEN  THEN  ELSE END | **Case** − The case operator is equivalent to nested bincond operator. | CASE f2 % 2  WHEN 0 THEN 'even'  WHEN 1 THEN 'odd'  END |

**Pig Latin – Comparison Operators**

The following table describes the comparison operators of Pig Latin.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | **Equal** − Checks if the values of two operands are equal or not; if yes, then the condition becomes true. | (a = b) is not true |
| != | **Not Equal** − Checks if the values of two operands are equal or not. If the values are not equal, then condition becomes true. | (a != b) is true. |
| > | **Greater than** − Checks if the value of the left operand is greater than the value of the right operand. If yes, then the condition becomes true. | (a > b) is not true. |
| < | **Less than** − Checks if the value of the left operand is less than the value of the right operand. If yes, then the condition becomes true. | (a < b) is true. |
| >= | **Greater than or equal to** − Checks if the value of the left operand is greater than or equal to the value of the right operand. If yes, then the condition becomes true. | (a >= b) is not true. |
| <= | **Less than or equal to** − Checks if the value of the left operand is less than or equal to the value of the right operand. If yes, then the condition becomes true. | (a <= b) is true. |
| Matches | **Pattern matching** − Checks whether the string in the left-hand side matches with the constant in the right-hand side. | f1 matches '.\*tutorial.\*' |

**Pig Latin – Type Construction Operators**

The following table describes the Type construction operators of Pig Latin.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| () | **Tuple constructor operator** − This operator is used to construct a tuple. | (Raju, 30) |
| {} | **Bag constructor operator** − This operator is used to construct a bag. | {(Raju, 30), (Mohammad, 45)} |
| [] | **Map constructor operator** − This operator is used to construct a tuple. | [name#Raja, age#30] |

**Pig Latin – Relational Operations**

The following table describes the relational operators of Pig Latin.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| **Loading and Storing** | |
| LOAD | To Load the data from the file system (local/HDFS) into a relation. |
| STORE | To save a relation to the file system (local/HDFS). |
| **Filtering** | |
| FILTER | To remove unwanted rows from a relation. |
| DISTINCT | To remove duplicate rows from a relation. |
| FOREACH, GENERATE | To generate data transformations based on columns of data. |
| STREAM | To transform a relation using an external program. |
| **Grouping and Joining** | |
| JOIN | To join two or more relations. |
| COGROUP | To group the data in two or more relations. |
| GROUP | To group the data in a single relation. |
| CROSS | To create the cross product of two or more relations. |
| **Sorting** | |
| ORDER | To arrange a relation in a sorted order based on one or more fields (ascending or descending). |
| LIMIT | To get a limited number of tuples from a relation. |
| **Combining and Splitting** | |
| UNION | To combine two or more relations into a single relation. |
| SPLIT | To split a single relation into two or more relations. |
| **Diagnostic Operators** | |
| DUMP | To print the contents of a relation on the console. |
| DESCRIBE | To describe the schema of a relation. |
| EXPLAIN | To view the logical, physical, or MapReduce execution plans to compute a relation. |
| ILLUSTRATE | To view the step-by-step execution of a series of statements. |

**Working with Functions in Pig.**

In addition to the built-in functions, Apache Pig provides extensive support for **U**ser **D**efined **F**unctions (UDF’s). Using these UDF’s, we can define our own functions and use them. The UDF support is provided in six programming languages, namely, Java, Jython, Python, JavaScript, Ruby and Groovy.

For writing UDF’s, complete support is provided in Java and limited support is provided in all the remaining languages. Using Java, you can write UDF’s involving all parts of the processing like data load/store, column transformation, and aggregation. Since Apache Pig has been written in Java, the UDF’s written using Java language work efficiently compared to other languages.

In Apache Pig, we also have a Java repository for UDF’s named **Piggybank**. Using Piggybank, we can access Java UDF’s written by other users, and contribute our own UDF’s.

**Types of UDF’s in Java**

While writing UDF’s using Java, we can create and use the following three types of functions −

* **Filter Functions** − The filter functions are used as conditions in filter statements. These functions accept a Pig value as input and return a Boolean value.
* **Eval Functions** − The Eval functions are used in FOREACH-GENERATE statements. These functions accept a Pig value as input and return a Pig result.
* **Algebraic Functions** − The Algebraic functions act on inner bags in a FOREACHGENERATE statement. These functions are used to perform full MapReduce operations on an inner bag.

**Writing UDF’s using Java**

To write a UDF using Java, we have to integrate the jar file **Pig-0.15.0.jar**. In this section, we discuss how to write a sample UDF using Eclipse. Before proceeding further, make sure you have installed Eclipse and Maven in your system.

Follow the steps given below to write a UDF function −

* Open Eclipse and create a new project (say **myproject**).
* Convert the newly created project into a Maven project.
* Copy the following content in the pom.xml. This file contains the Maven dependencies for Apache Pig and Hadoop-core jar files.
* Apache Pig provides various built-in functions namely **eval, load, store, math, string, bag** and **tuple** functions.

## **Eval Functions**

* Given below is the list of **eval** functions provided by Apache Pig.

|  |  |
| --- | --- |
| **S.N.** | **Function & Description** |
| 1 | [AVG()](https://www.tutorialspoint.com/apache_pig/apache_pig_avg.htm)  To compute the average of the numerical values within a bag. |
| 2 | [BagToString()](https://www.tutorialspoint.com/apache_pig/apache_pig_bagtostring.htm)  To concatenate the elements of a bag into a string. While concatenating, we can place a delimiter between these values (optional). |
| 3 | [CONCAT()](https://www.tutorialspoint.com/apache_pig/apache_pig_concat.htm)  To concatenate two or more expressions of same type. |
| 4 | [COUNT()](https://www.tutorialspoint.com/apache_pig/apache_pig_count.htm)  To get the number of elements in a bag, while counting the number of tuples in a bag. |
| 5 | [COUNT\_STAR()](https://www.tutorialspoint.com/apache_pig/apache_pig_count_star.htm)  It is similar to the **COUNT()** function. It is used to get the number of elements in a bag. |
| 6 | [DIFF()](https://www.tutorialspoint.com/apache_pig/apache_pig_diff.htm)  To compare two bags (fields) in a tuple. |
| 7 | [IsEmpty()](https://www.tutorialspoint.com/apache_pig/apache_pig_isempty.htm)  To check if a bag or map is empty. |
| 8 | [MAX()](https://www.tutorialspoint.com/apache_pig/apache_pig_max.htm)  To calculate the highest value for a column (numeric values or chararrays) in a single-column bag. |
| 9 | [MIN()](https://www.tutorialspoint.com/apache_pig/apache_pig_min.htm)  To get the minimum (lowest) value (numeric or chararray) for a certain column in a single-column bag. |
| 10 | [PluckTuple()](https://www.tutorialspoint.com/apache_pig/apache_pig_plucktuple.htm)  Using the Pig Latin **PluckTuple()** function, we can define a string Prefix and filter the columns in a relation that begin with the given prefix. |
| 11 | [SIZE()](https://www.tutorialspoint.com/apache_pig/apache_pig_size.htm)  To compute the number of elements based on any Pig data type. |
| 12 | [SUBTRACT()](https://www.tutorialspoint.com/apache_pig/apache_pig_subtract.htm)  To subtract two bags. It takes two bags as inputs and returns a bag which contains the tuples of the first bag that are not in the second bag. |
| 13 | [SUM()](https://www.tutorialspoint.com/apache_pig/apache_pig_sum.htm)  To get the total of the numeric values of a column in a single-column bag. |
| 14 | [TOKENIZE()](https://www.tutorialspoint.com/apache_pig/apache_pig_tokenize.htm)  To split a string (which contains a group of words) in a single tuple and return a bag which contains the output of the split operation. |

**Apache Pig - Load & Store Functions**

The **Load** and **Store** functions in Apache Pig are used to determine how the data goes ad comes out of Pig. These functions are used with the load and store operators. Given below is the list of load and store functions available in Pig.

|  |  |
| --- | --- |
| **S.N.** | **Function & Description** |
| 1 | [PigStorage()](https://www.tutorialspoint.com/apache_pig/apache_pig_pigstore.htm)  To load and store structured files. |
| 2 | [TextLoader()](https://www.tutorialspoint.com/apache_pig/apache_pig_textloader.htm)  To load unstructured data into Pig. |
| 3 | [BinStorage()](https://www.tutorialspoint.com/apache_pig/apache_pig_binstorage.htm)  To load and store data into Pig using machine readable format. |
| 4 | [Handling Compression](https://www.tutorialspoint.com/apache_pig/apache_pig_handling_compression.htm)  In Pig Latin, we can load and store compressed data. |

**Apache Pig - Bag & Tuple Functions**

Given below is the list of Bag and Tuple functions.

|  |  |
| --- | --- |
| **S.N.** | **Function & Description** |
| 1 | [TOBAG()](https://www.tutorialspoint.com/apache_pig/apache_pig_tobag.htm)  To convert two or more expressions into a bag. |
| 2 | [TOP()](https://www.tutorialspoint.com/apache_pig/apache_pig_top.htm)  To get the top **N** tuples of a relation. |
| 3 | [TOTUPLE()](https://www.tutorialspoint.com/apache_pig/apache_pig_totuple.htm)  To convert one or more expressions into a tuple. |
| 4 | [TOMAP()](https://www.tutorialspoint.com/apache_pig/apache_pig_tomap.htm)  To convert the key-value pairs into a Map. |

**Apache Pig - String Functions**

We have the following String functions in Apache Pig.

|  |  |
| --- | --- |
| **S.N.** | **Functions & Description** |
| 1 | [ENDSWITH(string, testAgainst)](https://www.tutorialspoint.com/apache_pig/apache_pig_endswith.htm)  To verify whether a given string ends with a particular substring. |
| 2 | [STARTSWITH(string, substring)](https://www.tutorialspoint.com/apache_pig/apache_pig_startswith.htm)  Accepts two string parameters and verifies whether the first string starts with the second. |
| 3 | [SUBSTRING(string, startIndex, stopIndex)](https://www.tutorialspoint.com/apache_pig/apache_pig_substring.htm)  Returns a substring from a given string. |
| 4 | [EqualsIgnoreCase(string1, string2)](https://www.tutorialspoint.com/apache_pig/apache_pig_equalsignorecase.htm)  To compare two stings ignoring the case. |
| 5 | [INDEXOF(string, ‘character’, startIndex)](https://www.tutorialspoint.com/apache_pig/apache_pig_indexof.htm)  Returns the first occurrence of a character in a string, searching forward from a start index. |

**Apache Pig - Date-time Functions**

Apache Pig provides the following Date and Time functions −

|  |  |
| --- | --- |
| **S.N.** | **Functions & Description** |
| 1 | [ToDate(milliseconds)](https://www.tutorialspoint.com/apache_pig/apache_pig_todate.htm)  This function returns a date-time object according to the given parameters. The other alternative for this function are ToDate(iosstring), ToDate(userstring, format), ToDate(userstring, format, timezone) |
| 2 | [CurrentTime()](https://www.tutorialspoint.com/apache_pig/apache_pig_currenttime.htm)  returns the date-time object of the current time. |
| 3 | [GetDay(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_getday.htm)  Returns the day of a month from the date-time object. |
| 4 | [GetHour(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_gethour.htm)  Returns the hour of a day from the date-time object. |
| 5 | [GetMilliSecond(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_getmillisecond.htm)  Returns the millisecond of a second from the date-time object. |
| 6 | [GetMinute(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_getminute.htm)  Returns the minute of an hour from the date-time object. |
| 7 | [GetMonth(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_getmonth.htm)  Returns the month of a year from the date-time object. |
| 8 | [GetSecond(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_getsecond.htm)  Returns the second of a minute from the date-time object. |
| 9 | [GetWeek(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_getweek.htm)  Returns the week of a year from the date-time object. |
| 10 | [GetWeekYear(datetime)](https://www.tutorialspoint.com/apache_pig/apache_pig_getweekyear.htm)  Returns the week year from the date-time object. |
|  |  |

**Introduction to NoSQL**

NoSQL is a type of database management system (DBMS) that is designed to handle and store large volumes of unstructured and semi-structured data. Unlike traditional relational databases that use tables with pre-defined schemas to store data, NoSQL databases use flexible data models that can adapt to changes in data structures and are capable of scaling horizontally to handle growing amounts of data.

The term NoSQL originally referred to “non-SQL” or “non-relational” databases, but the term has since evolved to mean “not only SQL,” as NoSQL databases have expanded to include a wide range of different database architectures and data models.

**NoSQL databases are generally classified into four main categories:**

1. **Document databases:** These databases store data as semi-structured documents, such as JSON or XML, and can be queried using document-oriented query languages.
2. **Key-value stores:** These databases store data as key-value pairs, and are optimized for simple and fast read/write operations.
3. **Column-family stores:** These databases store data as column families, which are sets of columns that are treated as a single entity. They are optimized for fast and efficient querying of large amounts of data.
4. **Graph databases:** These databases store data as nodes and edges, and are designed to handle complex relationships between data.

NoSQL databases are often used in applications where there is a high volume of data that needs to be processed and analyzed in real-time, such as social media analytics, e-commerce, and gaming. They can also be used for other applications, such as content management systems, document management, and customer relationship management.

However, NoSQL databases may not be suitable for all applications, as they may not provide the same level of data consistency and transactional guarantees as traditional relational databases. It is important to carefully evaluate the specific needs of an application when choosing a database management system.

**NoSQL** originally referring to non SQL or non relational is a database that provides a mechanism for storage and retrieval of data. This data is modeled in means other than the tabular relations used in relational databases. Such databases came into existence in the late 1960s**,** but did not obtain the NoSQL moniker until a surge of popularity in the early twenty-first century. NoSQL databases are used in real-time web applications and big data and their use are increasing over time.

* NoSQL systems are also sometimes called Not only SQL to emphasize the fact that they may support SQL-like query languages. A NoSQL database includes simplicity of design, simpler horizontal scaling to clusters of machines,**has** and finer control over availability. The data structures used by NoSQL databases are different from those used by default in relational databases which makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it should solve.
* NoSQL databases, also known as “not only SQL” databases, are a new type of database management system that has**,** gained popularity in recent years. Unlike traditional relational databases, NoSQL databases are designed to handle large amounts of unstructured or semi-structured data, and they can accommodate dynamic changes to the data model. This makes NoSQL databases a good fit for modern web applications, real-time analytics, and big data processing.
* Data structures used by NoSQL databases are sometimes also viewed as more flexible than relational database tables. Many NoSQL stores compromise consistency in favor of availability, speed,**,** and partition tolerance. Barriers to the greater adoption of NoSQL stores include the use of low-level query languages, lack of standardized interfaces, and huge previous investments in existing relational databases.
* Most NoSQL stores lack true ACID(Atomicity, Consistency, Isolation, Durability) transactions but a few databases, such as MarkLogic, Aerospike, FairCom c-treeACE, Google Spanner (though technically a NewSQL database), Symas LMDB, and OrientDB have made them central to their designs.
* Most NoSQL databases offer a concept of eventual consistency in which database changes are propagated to all nodes so queries for data might not return updated data immediately or might result in reading data that is not accurate which is a problem known as stale reads. Also,**has** some NoSQL systems may exhibit lost writes and other forms of data loss. Some NoSQL systems provide concepts such as write-ahead logging to avoid data loss.
* One simple example of a NoSQL database is a document database. In a document database, data is stored in documents rather than tables. Each document can contain a different set of fields, making it easy to accommodate changing data requirements
* For example, “Take, for instance, a database that holds data regarding employees.”. In a relational database, this information might be stored in tables, with one table for employee information and another table for department information. In a document database, each employee would be stored as a separate document, with all of their information contained within the document.
* NoSQL databases are a relatively new type of database management system that has**a** gained popularity in recent years due to their scalability and flexibility. They are designed to handle large amounts of unstructured or semi-structured data and can handle dynamic changes to the data model. This makes NoSQL databases a good fit for modern web applications, real-time analytics, and big data processing.

**Key Features of NoSQL:**

1. **Dynamic schema:** NoSQL databases do not have a fixed schema and can accommodate changing data structures without the need for migrations or schema alterations.
2. **Horizontal scalability:** NoSQL databases are designed to scale out by adding more nodes to a database cluster, making them well-suited for handling large amounts of data and high levels of traffic.
3. **Document-based:** Some NoSQL databases, such as MongoDB, use a document-based data model, where data is stored in a schema-less semi-structured format, such as JSON or BSON.
4. **Key-value-based:** Other NoSQL databases, such as Redis, use a key-value data model, where data is stored as a collection of key-value pairs.
5. **Column-based:** Some NoSQL databases, such as Cassandra, use a column-based data model, where data is organized into columns instead of rows.
6. **Distributed and high availability:** NoSQL databases are often designed to be highly available and to automatically handle node failures and data replication across multiple nodes in a database cluster.
7. **Flexibility:** NoSQL databases allow developers to store and retrieve data in a flexible and dynamic manner, with support for multiple data types and changing data structures.
8. **Performance:** NoSQL databases are optimized for high performance and can handle a high volume of reads and writes, making them suitable for big data and real-time applications.

**Advantages of NoSQL:** There are many advantages of working with NoSQL databases such as MongoDB and Cassandra. The main advantages are high scalability and high availability.

1. **High scalability:** NoSQL databases use sharding for horizontal scaling. Partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved is sharding. Vertical scaling means adding more resources to the existing machine whereas horizontal scaling means adding more machines to handle the data. Vertical scaling is not that easy to implement but horizontal scaling is easy to implement. Examples of horizontal scaling databases are MongoDB, Cassandra, etc. NoSQL can handle a huge amount of data because of scalability, as the data grows NoSQL scales**The auto** itself to handle that data in an efficient manner.
2. **Flexibility:** NoSQL databases are designed to handle unstructured or semi-structured data, which means that they can accommodate dynamic changes to the data model. This makes NoSQL databases a good fit for applications that need to handle changing data requirements.
3. **High availability:** The auto**,** replication feature in NoSQL databases makes it highly available because in case of any failure data replicates itself to the previous consistent state.
4. **Scalability:** NoSQL databases are highly scalable, which means that they can handle large amounts of data and traffic with ease. This makes them a good fit for applications that need to handle large amounts of data or traffic
5. **Performance:** NoSQL databases are designed to handle large amounts of data and traffic, which means that they can offer improved performance compared to traditional relational databases.
6. **Cost-effectiveness:** NoSQL databases are often more cost-effective than traditional relational databases, as they are typically less complex and do not require expensive hardware or software.
7. **Agility:** Ideal for agile development.

**Disadvantages of NoSQL:** NoSQL has the following disadvantages.

1. **Lack of standardization:**  There are many different types of NoSQL databases, each with its own unique strengths and weaknesses. This lack of standardization can make it difficult to choose the right database for a specific application
2. **Lack of ACID compliance:** NoSQL databases are not fully ACID-compliant, which means that they do not guarantee the consistency, integrity, and durability of data. This can be a drawback for applications that require strong data consistency guarantees.
3. **Narrow focus:** NoSQL databases have a very narrow focus as it is mainly designed for storage but it provides very little functionality. Relational databases are a better choice in the field of Transaction Management than NoSQL.
4. **Open-source:** NoSQL is an **database**open-source database. There is no reliable standard for NoSQL yet. In other words, two database systems are likely to be unequal.
5. **Lack of support for complex queries:** NoSQL databases are not designed to handle complex queries, which means that they are not a good fit for applications that require complex data analysis or reporting.
6. **Lack of maturity:** NoSQL databases are relatively new and lack the maturity of traditional relational databases. This can make them less reliable and less secure than traditional databases.
7. **Management challenge:** The purpose of big data tools is to make the management of a large amount of data as simple as possible. But it is not so easy. Data management in NoSQL is much more complex than in a relational database. NoSQL, in particular, has a reputation for being challenging to install and even more hectic to manage on a daily basis.
8. **GUI is not available:** GUI mode tools to access the database are not flexibly available in the market.
9. **Backup:** Backup is a great weak point for some NoSQL databases like MongoDB. MongoDB has no approach for the backup of data in a consistent manner.
10. **Large document size:** Some database systems like MongoDB and CouchDB store data in JSON format. This means that documents are quite large (BigData, network bandwidth, speed), and having descriptive key names actually hurts since they increase the document size.

**Types of NoSQL database:** Types of NoSQL databases and the name of the database system that falls in that category are:

1. **Graph Databases**: Examples – Amazon Neptune, Neo4j
2. **Key value store:** Examples – Memcached, Redis, Coherence
3. **Column:** Examples – Hbase, Big Table, Accumulo
4. **Document-based:** Examples – MongoDB, CouchDB, Cloudant

**When should NoSQL be used:**

1. When a huge amount of data needs to be stored and retrieved.
2. The relationship between the data you store is not that important
3. The data changes over time and is not structured.
4. Support of Constraints and Joins is not required at the database level
5. The data is growing continuously and you need to scale the database regularly to handle the data.

In conclusion, NoSQL databases offer several benefits over traditional relational databases, such as scalability, flexibility, and cost-effectiveness. However, they also have several drawbacks, such as a lack of standardization, lack of ACID compliance, and lack of support for complex queries. When choosing a database for a specific application, it is important to weigh the benefits and drawbacks carefully to determine the best fit.

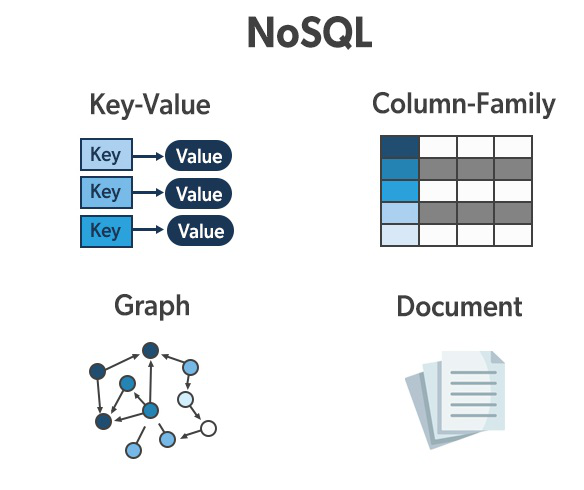
**Types of NoSQL Databases**

A database is a collection of structured data or information which is stored in a computer system and can be accessed easily. A database is usually managed by a Database Management System (DBMS).

NoSQL is a non-relational database that is used to store the data in the nontabular form. NoSQL stands for Not only SQL. The main types are documents, key-value, wide-column, and graphs.

**Types of NoSQL Database:**

* Document-based databases
* Key-value stores
* Column-oriented databases
* Graph-based databases



**Document-Based Database:**

The document-based database is a nonrelational database. Instead of storing the data in rows and columns (tables), it uses the documents to store the data in the database. A document database stores data in JSON, BSON, or XML documents.

Documents can be stored and retrieved in a form that is much closer to the data objects used in applications which means less translation is required to use these data in the applications. In the Document database, the particular elements can be accessed by using the index value that is assigned for faster querying.

Collections are the group of documents that store documents that have similar contents. Not all the documents are in any collection as they require a similar schema because document databases have a flexible schema.

Key features of documents database:

* Flexible schema: Documents in the database has a flexible schema. It means the documents in the database need not be the same schema.
* Faster creation and maintenance: the creation of documents is easy and minimal maintenance is required once we create the document.
* No foreign keys: There is no dynamic relationship between two documents so documents can be independent of one another. So, there is no requirement for a foreign key in a document database.
* Open formats: To build a document we use XML, JSON, and others.

**Key-Value Stores:**

A key-value store is a nonrelational database. The simplest form of a NoSQL database is a key-value store. Every data element in the database is stored in key-value pairs. The data can be retrieved by using a unique key allotted to each element in the database. The values can be simple data types like strings and numbers or complex objects.

A key-value store is like a relational database with only two columns which is the key and the value.

Key features of the key-value store:

* Simplicity.
* Scalability.
* Speed.

**Column Oriented Databases:**

A column-oriented database is a non-relational database that stores the data in columns instead of rows. That means when we want to run analytics on a small number of columns, you can read those columns directly without consuming memory with the unwanted data.

Columnar databases are designed to read data more efficiently and retrieve the data with greater speed. A columnar database is used to store a large amount of data. Key features of columnar oriented database:

* Scalability.
* Compression.
* Very responsive.

**Graph-Based databases:**

Graph-based databases focus on the relationship between the elements. It stores the data in the form of nodes in the database. The connections between the nodes are called links or relationships.

Key features of graph database:

* In a graph-based database, it is easy to identify the relationship between the data by using the links.
* The Query’s output is real-time results.
* The speed depends upon the number of relationships among the database elements.
* Updating data is also easy, as adding a new node or edge to a graph database is a straightforward task that does not require significant schema changes.

# BDA

# UNIT-V Understanding Analytics in Big Data: Comparing, Reporting and Analysis, Types of Analytics, Points to Consider during Analysis and Understanding text Analytics. Applications: Social Media Analytics and Text Mining Mobile Analytics

**Difference between Big Data and Data Analytics**

**1.** [**Big Data**](https://www.geeksforgeeks.org/what-is-big-data/)**:** Big data refers to the large volume of data and also the data is increasing with a rapid speed with respect to time. It includes structured and unstructured and semi-structured data which is so large and complex and it cant not be managed by any traditional data management tool. Specialized big data management tools are required to store and process the data. Volume, velocity, and variety represents the primary characteristics of big data. Stock exchanges, [Data warehouses](https://www.geeksforgeeks.org/data-warehousing/), Sensors, social media sites, jet engines, etc are the different sources of Big data.

**Application of Big data :**

* Big Data for Financial Services
* Big Data in Communications
* Communications, Media and Entertainment
* Big Data for Retail
* Banking and Securities

**Benefits of Big Data :**

* Diversify revenue streams
* Big Data is Secure
* Authoritive and Actionable
* Product price optimization
* Greater innovations

**2.** [**Data Analytics**](https://www.geeksforgeeks.org/data-analytics-and-its-type/) **:** Data Analytics refers to the process of analyzing the raw data and finding out conclusions about that information. It helps in taking raw data and uncovering patterns by examining it to extract valuable insights from it. The aim behind data analytics is to enhance productivity and business gain. It helps companies to better understand their customers, planning strategies accordingly and develop products. Descriptive, Diagnostic, Predictive, and Prescriptive are the four basic types of data analytics.

**Application of Data Analytics:**

* Healthcare
* For travelling
* Gaming
* Energy Managementand aRisk detection and management

**Benefits of Data Analytics :**

* Improved Performance
* Better decision making
* Maintaining quality and consistency
* Data-driven marketing
* Real-time forecasting and monitoring

**How does big data fit into data analysis solutions?**

* Big data plays a crucial role in data analysis solutions by providing organizations with large amounts of data that can be used to uncover insights and support decision-making.
* Data analytics solutions, such as Hadoop, Spark, and NoSQL databases, are specifically designed to handle big data and can be used to process, store, and analyze large amounts of data in a distributed and parallel manner.
* Additionally, big data can be integrated with other data sources such as structured data, semi-structured data, and unstructured data, to form a holistic view of the organization’s data landscape, which can lead to more accurate predictions, better decision-making, and more effective outcomes.
* In summary, big data is a crucial input for data analysis solutions, and data analytics solutions are specifically designed to handle big data and extract valuable insights from it.

**Difference between Big Data and Data Analytics:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **BIG DATA** | **DATA ANALYTICS** |
| **01.** | Big data refers to a large volume of data and also the data is increasing at, modeling rapid speed with respect to time. | Data Analytics refers to the process of analyzing the raw data and finding out conclusions about that information. |
| **02.** | Big data includes Structured, Unstructured and Semi-structured the three types of data. | Descriptive, Diagnostic, Predictive, Prescriptive are the four basic types of data analytics. |
| **03.** | The purpose of big data is to store huge volume of data and to process it. | The purpose of data analytics is to analyze the raw data and find out insights for the information. |
| **04.** | Parallel computing and other complex automation tools are used to handle big data. | Predictive and statistical modelling with relatively simple tools used to handle data analytics. |
| **05.** | Big data operations are handled by big data professionals. | Data analytics is performed by skilled data analysts. |
| **06.** | Big data analysts need the knowledge of programming, NoSQL databases, distributed systems, and frameworks. | Data Analysts need the knowledge of programming, statistics, and mathematics. |
| **07.** | Big data is mainly found in financial services, Media and Entertainment, communication, Banking, information technology, retail, etc. | Data analytics is mainly used in business for risk detection and management, science, travelling, health care, Gaming, energy management, and information technology. |
| **08.** | It supports in dealing with huge volumes of data. | It supports in examining raw data and recognizing useful information. |
| **09.** | It is considered as the first step as first big data generated and then stored. | It is considered as second step as it performs analysis on the large data sets. |
| **10.** | Some of the big data tools are Apache Hadoop, Cloudera Distribution for Hadoop, Cassandra, MongoDB etc. | Some of the data analytics tools are Tableau Public, Python, Apache Spark, Excel, RapidMiner, KNIME etc. |

**Reporting and Analysis,**

**Why do you need to know the difference between reporting and analytics?**

Yes, although these words are often mistakenly used interchangeably, the truth is that the difference between reporting and analytics can play a big role for your company.

Knowing what these terms have and don’t have in common can help you identify what you need best and on what occasion, for example:

1. **Choose the right people:** When you want to staff the company with the right people, you need to know each person’s expertise.
2. **Choose the right tool:** Some tools are built for reporting, while others are focused on analytics.

Choosing the right people and the right toolset is the fastest way to success, so let’s see what is what in terms of analytics vs. reporting.

**What is reporting?**

When you have a set of data stored somewhere and you are happy with the structure (e.g. you have already cleaned or enhanced your dataset), it is time to make something out of it. So here comes the “reporting” part. Data reporting is about taking the available information (e.g. your dataset), organizing it, and displaying it in a well-structured and digestible format we call “reports”. You can present data from various sources, making it available for anyone to analyze it.

Reporting is a great way to help the internal teams and experts answer the question of **what** ishappening.

**What is analytics?**

Analytics is a much wider term. It actually contains reporting as you cannot actually talk about analytics without proper reporting. Having said that, for proper decision-making, you will need much more than that.

Analytics is about diving deeper into your data and reports in order to look for insights. It’s actually an attempt to answer **why** something is happening. Analytics powers up decision-making as the main goal is to make sense of the data explaining the reason behind the reported numbers. Last but not least, in the context of reporting vs. analytics, you will find that analytics

includes recommendations as well. After you analyze your data and know why something is happening, your aim is to determine a course of action to either improve something or provide a solution.

As discussed, to do a proper analysis you will need well-designed reports. As opposed to reporting where data is just grouped up and presented, analytics rests on dashboards that allow you to dive deeper into existing numbers and look for insights.

**Analytics vs. reporting: Key differences**

Hopefully the difference between reporting and analytics will be clearer now and you will have a better idea of the purpose and benefits of each. As understanding of the ins and outs of reporting vs analytics can play a critical role in your company and decisions.

Let’s summarize the key differences across three pillars:

|  | **Reporting** | **Analytics** |
| --- | --- | --- |
| Purpose | Focuses on **what** is happening | Focuses on **why** something is happening |
| Tasks | Cleaning, organizing and summarizing your data | Exploring, analyzing, and questioning your data |
| Value | Transforms your data into information | Transforms the information into insights & recommendations. |

The ideal scenario is to combine both reports & analytics for better results, just like we did with [no-code agency XRay. Tech](https://www.coupler.io/case-studies/xray-tech) helping them to track and analyze their time.

**Types of reports vs. analytics**

**What types of reports can you use?**

Like with many things, there isn’t just one report type you can use for every case. Depending on your particular case study, you may need to use a specific type of report to better present data.  
  
The type of the report is defined by many things, such as who will be receiving the report, how often, the size of the report and what it is actually reporting. Below you may find the most common types of reports you can use:

* **Long Reports:** Long reports are usually longer than 10 pages and are typically used on formal occasions.
* **Short Reports:** Short reports are the exact opposite. They are less than 10 pages and tend to withhold less data, usually shared in informal occasions (e.g. quickly sharing a set of data).
* **Internal Reports:** Internal reports are created and shared either within the same organization or even the same department.
* **External Reports:** External reports are built with the aim of being shared outside the organization.
* **Vertical Reports:** Vertical reports are typically internal reports that are shared across different levels of the hierarchy of the organization (e.g. sharing a report with your manager or stakeholders).
* **Lateral Reports:** Lateral reports are the ones that are shared horizontally within the organization. Take, for example, a report shared between two different departments (e.g. HR and finance).
* **Periodic Reports:** We call periodic reports the ones that are created periodically (e.g. on a monthly basis). The report will have the exact same format, but the data within are changing based on the interval.

Besides the above categories, we can divide the report types based on their content as well. This way, we can have different types of reports, four of the most common are:

* **Financial Reports:** Financial reports provide an overview of the most important financial metrics of the organization (e.g. profits and losses, expenses and revenue).
* **Marketing Reports:** Marketing reports are used to evaluate marketing efforts. Usually containing how much money was invested and what the return was in terms of traffic.
* **Sales Reports:** Sales reports focus more on the revenue side of things. Usually, they report the number of sales, the revenue, and items sold.
* **Management Reports:** The main goal of [management reporting](https://blog.coupler.io/management-reporting/) is to provide the information investors need. This may include the return on investment, the share price, and profits and losses for a given time frame.

**Types of analytics**

Analytics is what we do when we want to understand **why** something is happening. Having said that, there is no one-fit-all approach. Like with reports, there are different types of analytics you can perform and each one serves its own purpose.  
  
 the most common types of analytics (usually presented in the following order):

* **Descriptive:** Descriptive analytics is when you assess historical data and try to identify specific patterns. The main goal is to answer what happened and if it was expected or not, making comparisons with other timeframes.
* **Diagnostic:** When we know what’s going on, the next step is to understand why. So you may have performed some descriptive analytics techniques and you were able to identify that sales went up by 12%. Diagnostic analytics is there to help identify why this happened and what actually worked for your business.
* **Predictive:** Predictive analytics involves sophisticated techniques that can help you use the patterns observed and make forecasts about future performance, e.g., [financial data analytics](https://blog.coupler.io/financial-data-analytics/). While this may require specific expertise, it’s extremely useful in order to be better prepared for the future.
* **Prescriptive:** Last but not least, prescriptive analytics techniques can help you identify the best course of action. This type of analytics is frequently used by marketers to draft their strategies and achieve better results.

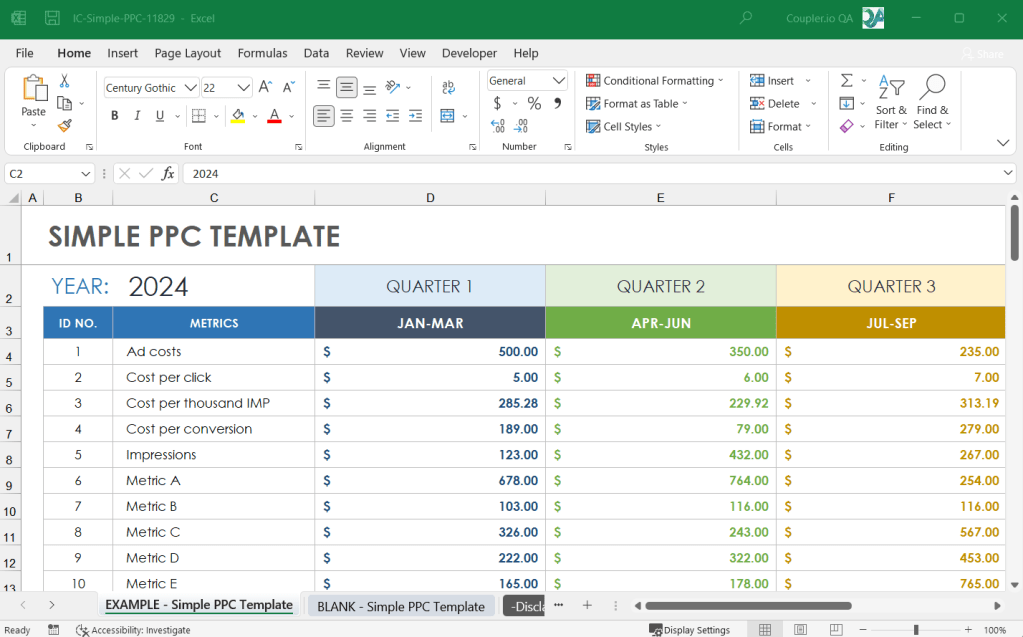
**The difference between reporting and analytics dashboards in examples**

Reports can answer critical questions like “How many products did we sell yesterday?” or “How many hours did we spend on this project?”. Using reports you can actually see the data, usually in but not limited to a tabular format, and use that to understand how your business is performing.

Analytical dashboards are a kind of report. They have a fixed structure and usually provide anewswwse overview of data metrics, so you can derive insights for making important decisions. For analytics, you usually need to perform one-time deep research on what is going on, why, and how it’s connected, sometimes even on each individual user level.

**Examples of reports**

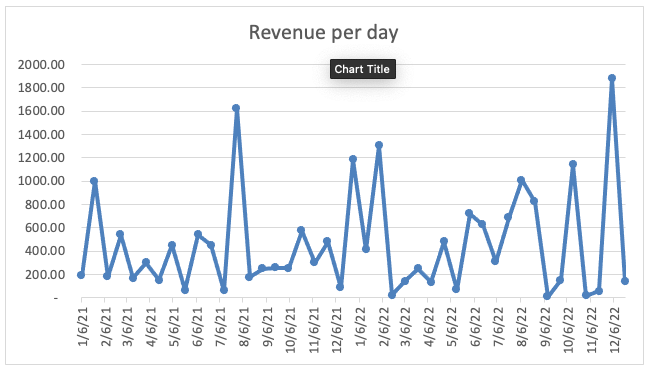
You’ve probably already used an Excel spreadsheet to store and manipulate or just read your exported data. This is an example of a report in Microsoft Excel:



Don’t get confused, reports are not just tables with data. You can also visualize your dataset and still call it a report. The below examples are part of reports showing what’s going on with the business using a bar chart



or a line chart.



**Introduction to Data Analysis**

Data Analysis Libraries

Data Visulization Libraries

Exploratory Data Analysis (EDA)

Data Preprocessing

Data Transformation

Time Series Data Analysis

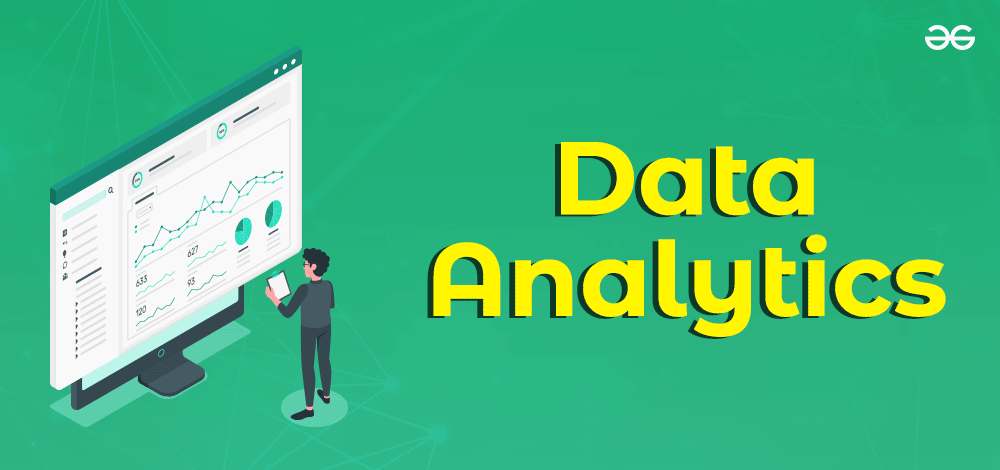
Case Studies and Projects

* [Machine Learning & Data Science Course](https://www.geeksforgeeks.org/courses/data-science-live?utm_source=geeksforgeeks&utm_medium=leftbar_lcta&utm_campaign=courses)

**Data Analytics and its type**

**Data analytics** is an important field that involves the process of collecting, processing, and interpreting data to uncover insights and help in making decisions. Data analytics is the practice of examining raw data to identify trends, draw conclusions, and extract meaningful information. This involves various techniques and tools to process and transform data into valuable insights that can be used for decision-making.

In this article, we will learn about **Data analytics, data** which will help businesses and individuals that can help them to enhance and solve complex problems, **Types of Data Analytics, Techniques** , **Tools** , and **the Importance of Data Analytics** .



Data Analytics

**What is Data Analytics?**

In this new digital world, data is being generated in an enormous amount which opens new paradigms. As we have high computing power and a large amount of data we can use this data to help us make data-driven decision making. The main benefits of data-driven decisions are that they are made up by observing past trends which have resulted in beneficial results.

In short, we can say that data analytics is the process of manipulating data to extract useful trends and hidden patterns that can help us derive valuable insights to make business predictions.

To gain expertise in data analytics and learn how to apply these techniques in real-world scenarios, consider enrolling in the [**Data Science Live course**](https://gfgcdn.com/tu/R8t/) . This course offers in-depth training on data analysis, machine learning, and statistical methods, equipping you with the skills needed to excel in the field of data science. Learn from industry experts and take your data analytics capabilities to the next level with practical, hands-on experience.

**Understanding Data Analytics**

Data analytics encompasses a wide array of techniques for analyzing data to gain valuable insights that can enhance various aspects of operations. By scrutinizing information, businesses can uncover patterns and metrics that might otherwise go unnoticed, enabling them to optimize processes and improve overall efficiency.

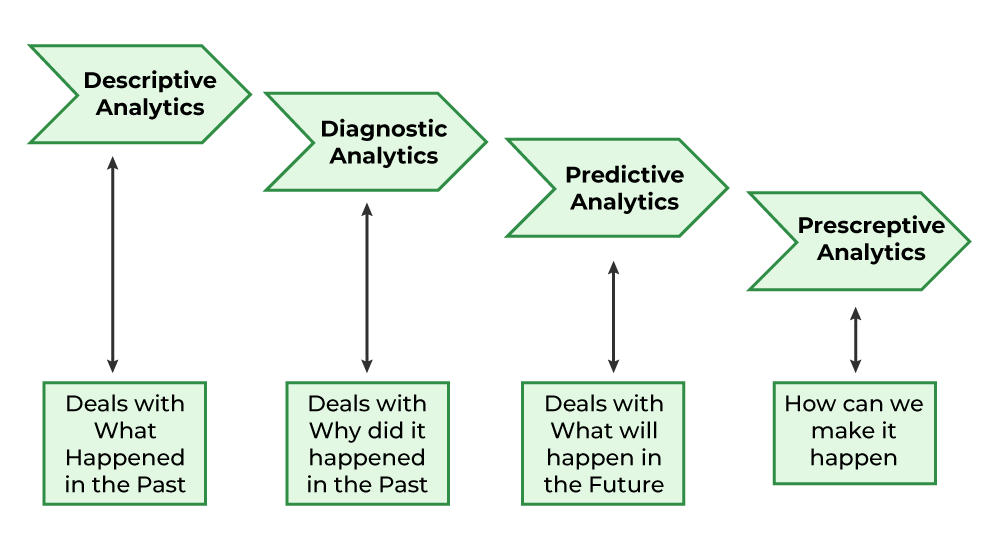
For instance, in manufacturing, companies collect data on machine runtime, downtime, and work queues to analyze and improve workload planning, ensuring machines operate at optimal levels.

Beyond production optimization, data analytics is utilized in diverse sectors. Gaming firms utilize it to design reward systems that engage players effectively, while content providers leverage analytics to optimize content placement and presentation, ultimately driving user engagement.

**Types of Data Analytics**

There are four major types of data analytics:

1. **Predictive (forecasting)**
2. **Descriptive (business intelligence and data mining)**
3. **Prescriptive (optimization and simulation)**
4. **Diagnostic analytics**



Data Analytics and its Types

**Predictive Analytics**

Predictive analytics turn the data into valuable, actionable information. predictive analytics uses data to determine the probable outcome of an event or a likelihood of a situation occurring. Predictive analytics holds a variety of statistical techniques from modeling, [machine learning](https://www.geeksforgeeks.org/machine-learning/) , [data mining](https://www.geeksforgeeks.org/data-mining/) , and [game theory](https://www.geeksforgeeks.org/game-theory/) that analyze current and historical facts to make predictions about a future event. Techniques that are used for predictive analytics are:

* Linear Regression
* Time Series Analysis and Forecasting
* Data Mining

**Basic Cornerstones of Predictive Analytics**

* Predictive modeling
* Decision Analysis and optimization
* Transaction profiling

**Descriptive Analytics**

Descriptive analytics looks at data and analyze past event for insight as to how to approach future events. It looks at past performance and understands the performance by mining historical data to understand the cause of success or failure in the past. Almost all management reporting such as sales, marketing, operations, and finance uses this type of analysis.

The descriptive model quantifies relationships in data in a way that is often used to classify customers or prospects into groups. Unlike a predictive model that focuses on predicting the behavior of a single customer, [Descriptive analytics](https://www.geeksforgeeks.org/descriptive-analysis-in-r-programming/) identifies many different relationships between customer and product.

**Common examples of Descriptive analytics are company reports that provide historic reviews like:**

* Data Queries
* Reports
* Descriptive Statistics
* Data dashboard

**Prescriptive Analytics**

Prescriptive Analytics automatically synthesize big data, mathematical science, business rule, and machine learning to make a prediction and then suggests a decision option to take advantage of the prediction.

Prescriptive analytics goes beyond predicting future outcomes by also suggesting action benefits from the predictions and showing the decision maker the implication of each decision option. Prescriptive Analytics not only anticipates what will happen and when to happen but also why it will happen. Further, Prescriptive Analytics can suggest decision options on how to take advantage of a future opportunity or mitigate a future risk and illustrate the implication of each decision option.

For example, [Prescriptive Analytics](https://www.geeksforgeeks.org/what-is-prescriptive-analytics-in-data-science/) can benefit healthcare strategic planning by using analytics to leverage operational and usage data combined with data of external factors such as economic data, population demography, etc.

**Diagnostic Analytics**

In this analysis, we generally use historical data over other data to answer any question or for the solution of any problem. We try to find any dependency and pattern in the historical data of the particular problem.

For example, companies go for this analysis because it gives a great insight into a problem, and they also keep detailed information about their disposal otherwise data collection may turn out individual for every problem and it will be very time-consuming.  Common techniques used for Diagnostic Analytics are:

* Data discovery
* Data mining
* Correlations

**The Role of Data Analytics**

Data analytics plays a pivotal role in enhancing operations, efficiency, and performance across various industries by uncovering valuable patterns and insights. Implementing data analytics techniques can provide companies with a competitive advantage. The process typically involves four fundamental steps:

* [**Data Mining**](https://www.geeksforgeeks.org/data-mining/) : This step involves gathering data and information from diverse sources and transforming them into a standardized format for subsequent analysis. Data mining can be a time-intensive process compared to other steps but is crucial for obtaining a comprehensive dataset.
* [**Data Management**](https://www.geeksforgeeks.org/what-is-data-management/) : Once collected, data needs to be stored, managed, and made accessible. Creating a database is essential for managing the vast amounts of information collected during the mining process. SQL (Structured Query Language) remains a widely used tool for database management, facilitating efficient querying and analysis of relational databases.
* **Statistical Analysis** : In this step, the gathered data is subjected to statistical analysis to identify trends and patterns. Statistical modeling is used to interpret the data and make predictions about future trends. Open-source programming languages like Python, as well as specialized tools like R, are commonly used for statistical analysis and graphical modeling.
* **Data Presentation** : The insights derived from data analytics need to be effectively communicated to stakeholders. This final step involves formatting the results in a manner that is accessible and understandable to various stakeholders, including decision-makers, analysts, and shareholders. Clear and concise data presentation is essential for driving informed decision-making and driving business growth.

**Steps in Data Analysis**

* **Define Data Requirements** : This involves determining how the data will be grouped or categorized. Data can be segmented based on various factors such as age, demographic, income, or gender, and can consist of numerical values or categorical data.
* **Data Collection** : Data is gathered from different sources, including computers, online platforms, cameras, environmental sensors, or through human personnel.
* **Data Organization** : Once collected, the data needs to be organized in a structured format to facilitate analysis. This could involve using spreadsheets or specialized software designed for managing and analyzing statistical data.
* [**Data Cleaning**](https://www.geeksforgeeks.org/data-cleansing-introduction/) : Before analysis, the data undergoes a cleaning process to ensure accuracy and reliability. This involves identifying and removing any duplicate or erroneous entries, as well as addressing any missing or incomplete data. Cleaning the data helps to mitigate potential biases and errors that could affect the analysis results.

**Usage of Data Analytics**

There are some key domains and strategic planning techniques in which Data Analytics has played a vital role:

* **Improved** [**Decision-Making**](https://www.geeksforgeeks.org/decision-making-meaning-nature-role-and-relationship-between-planning-and-decision-making/) – If we have supporting data in favour of a decision, then we can implement them with even more success probability. For example, if a certain decision or plan has to lead to better outcomes then there will be no doubt in implementing them again.
* **Better Customer Service –** Churn modeling is the best example of this in which we try to predict or identify what leads to customer churn and change those things accordingly so, that the attrition of the customers is as low as possible which is a most important factor in any organization.
* **Efficient Operations –** Data Analytics can help us understand what is the demand of the situation and what should be done to get better results then we will be able to streamline our processes which in turn will lead to efficient operations.
* **Effective Marketing –** Market segmentation techniques have been implemented to target this important factor only in which we are supposed to find the marketing techniques which will help us increase our sales and leads to effective marketing strategies.

**Future Scope of Data Analytics**

* **Retail** : To study sales patterns, consumer behavior, and inventory management, data analytics can be applied in the retail sector. Data analytics can be used by retailers to make data-driven decisions regarding what products to stock, how to price them, and how to best organize their stores.
* **Healthcare** : Data analytics can be used to evaluate patient data, spot trends in patient health, and create individualized treatment regimens. Data analytics can be used by healthcare companies to enhance patient outcomes and lower healthcare expenditures.
* **Finance** : In the field of finance, data analytics can be used to evaluate investment data, spot trends in the financial markets, and make wise investment decisions. Data analytics can be used by financial institutions to lower risk and boost the performance of investment portfolios.
* **Marketing** : By analyzing customer data, spotting trends in consumer behavior, and creating customized marketing strategies, data analytics can be used in marketing. Data analytics can be used by marketers to boost the efficiency of their campaigns and their overall impact.
* **Manufacturing** : Data analytics can be used to examine production data, spot trends in production methods, and boost production efficiency in the manufacturing sector. Data analytics can be used by manufacturers to cut costs and enhance product quality.
* **Transportation** : To evaluate logistics data, spot trends in transportation routes, and improve transportation routes, the transportation sector can employ data analytics. Data analytics can help transportation businesses cut expenses and speed up delivery times.

**5 Text Analytics Approaches: A Comprehensive Review**

Are you receiving more feedback than you could ever read, let alone summarize? Maybe

Here is my summary to break down these methods into 5 key approaches that are commonly used today.

**What is Text Analytics?**

Text analytics is the process of extracting meaning out of text. For example, this can be analyzing text written by customers in a customer survey, with the focus on finding common themes and trends. The idea is to be able to examine the customer feedback to inform the business on taking strategic action, in order to improve customer experience.

**What is Text Analytic software?**

To make text analytics the most efficient, organisations can use [text analytics software](https://getthematic.com/insights/the-best-text-analytics-software/), leveraging machine learning and natural language processing algorithms to find meaning in enormous amounts of text.

**How is Text Analytics used by companies?**

To take [Thematic](https://getthematic.com) as an example, we analyze the free-text feedback submitted in customer feedback forms, which was previously difficult to analyze, as companies spend time and resource struggling to do this manually.

Subsequently, we use text analytics to help companies find hidden customer insights and be able to easily answer questions about their existing customer data. In addition, with the help of text analytics software such as Thematic, companies can find recurrent and emerging themes, tracking trends and issues, and create visual reports for managers to track whether they are closing the loop with the end customer.

**Some Text Analytics background…**

Some try to reinvent the wheel by writing their own algorithms from scratch, others believe that Google and IBM APIs are the saviours, others again are stuck with technologies from the late 90’s that vendors pitch as “advanced Text Analytics”.

I’ve spent the last 15 years in Natural Language Processing, specifically in the area of making sense of text using algorithms: researching, creating, applying and selling the technology behind it.

My academic research resulted in algorithms used by hundreds of organizations (I’m the author of [KEA and Maui](http://www.medelyan.com/software)). The highlight of my text analytics career was at Google, where I wrote an algorithm that can analyse text in languages I don’t speak.

**5 Text Analytics Methods and Examples**

Here is my summary to break down these methods into 5 key approaches that are commonly used today.

**Text Analytics Approach 1: Word Spotting**

Let’s start with **word spotting**. First off, it’s not a thing!

The academic Natural Language Processing community does not register such an approach, and rightly so. In fact, in the academic world, [word spotting](http://ciir.cs.umass.edu/irdemo/hw-demo/wordspot_retr.html) refers to handwriting recognition (spotting which word a person, a doctor perhaps, has written).

There is also [keyword spotting](https://en.wikipedia.org/wiki/Keyword_spotting), which focuses on speech processing.

But to my knowledge, **word spotting is not a used for any type of text analysis**.

But I’ve heard frequently enough about it in meetings to include in this review. It’s loved by DIY analysts and Excel wizards and is a popular approach among many customer insights professionals.

The main idea behind text word spotting is this: If a word appears in text, we can assume that this piece of text is “about” that particular word. For example, if words like “price” or “cost” are mentioned in a review, this means that this review is about “Price”.

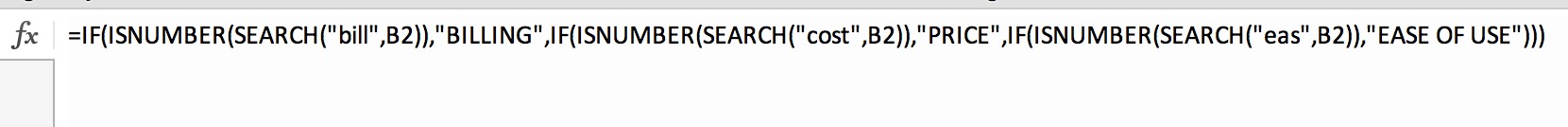
The beauty of the word spotting approach is its simplicity.

You can implement word spotting in an Excel spreadsheet in less than 10 minutes.

Or, you could write a script in Python or R. Here ’s how.

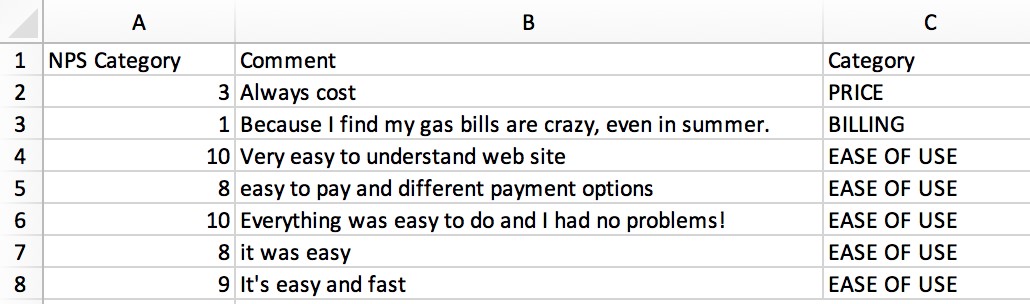
**How to build a Text Analytics solution in 10 minutes**

You can type in a formula, like this one, in Excel to categorize comments into “Billing”, “Pricing” and “Ease of use”:



And voilà!

Here it is applied to a Net Promoter Score survey where column B contains open-ended answers to questions “Why did you give us this score”:

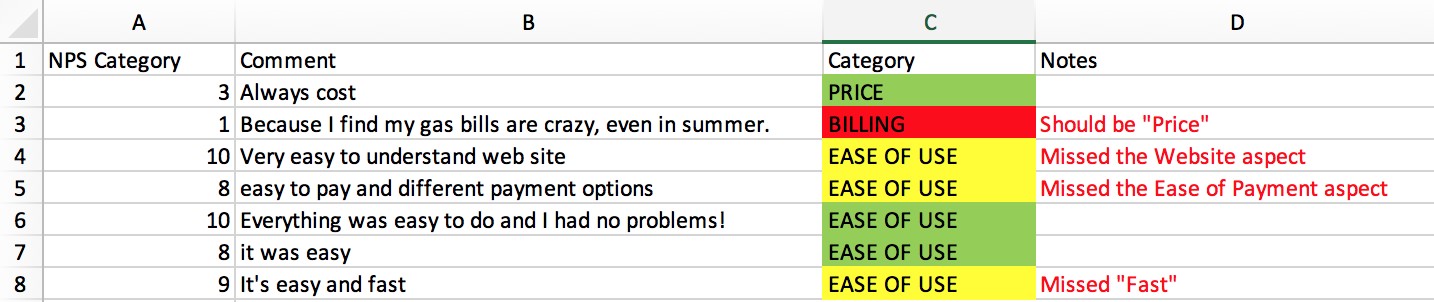


It probably took me less than 10 minutes to create this, and the result is so encouraging! But wait…

**Everyone loves simplicity. But in this case, simplicity sucks**

Various issues can easily crop up with this approach.

Here, annotated them for you.



Out of 7 comments, here only 3 were categorized correctly. “Billing” is actually about “Price”, and three other comments missed additional themes. Would you bet your customer insights on something that’s at best 50 accurate?

**When word spotting is OK**

You can imagine that the formula above can be tweaked further. And indeed, I’ve talked to companies who hand-crafted massive custom spreadsheets and are very happy with the results.

If you have a dataset with a couple of hundred responses that you only need to analyze once or twice, you can use this approach. If the dataset is small, you can review the results and ensure high accuracy very quickly.

**When word spotting fails**

As for the downside? Please don’t use word spotting:

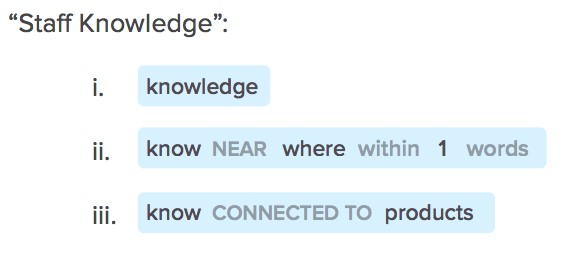
* If you have any substantial amount of data, more than several hundred responses
* If you won’t have time to review and correct the accuracy of each piece of text
* If you need to visualize the results (Excel will hear you swearing)
* If you need to share the results with your colleagues
* If you need to maintain the data consistently over time

There are also many other disadvantages to DIY word spotting, that we’ll discuss in the next post. I’ll also talk about what actually does work and is a good approach.

**Text Analytics Approach 2. Manual Rules**

The Manual Rules approach is closely related to word spotting. Both approaches operate on the same principle of creating a match pattern, but these patterns can also get quite complex.

For example, a manual rule could involve the use of regular expressions – something you can’t easily implement in Excel. Here is a rule for assigning the category “Staff Knowledge” from a popular enterprise solution Medallia:



Majority of Text Analytics providers as well as many other smaller players, who sell Text Analytics as an add-on to their main offering, provide an interface that makes it easy to create and manage such rules. They also sometimes offer professional services to help with the creation of these rules.

The best thing about Manual Rules is that they can be understood by a person. They are explainable, and therefore can be tweaked and adjusted when needed.

But the bottom line is that creating these rules takes a lot of effort. You also need to ensure that they are accurate and maintain them over time.

To get you started, some companies come with pre-packaged rules, already organized into a taxonomy. For example, they would have a category “Price”, with hundreds of words and phrases already pre-set, and underneath they might have sub-categories such as “Cheap” and “Expensive”.

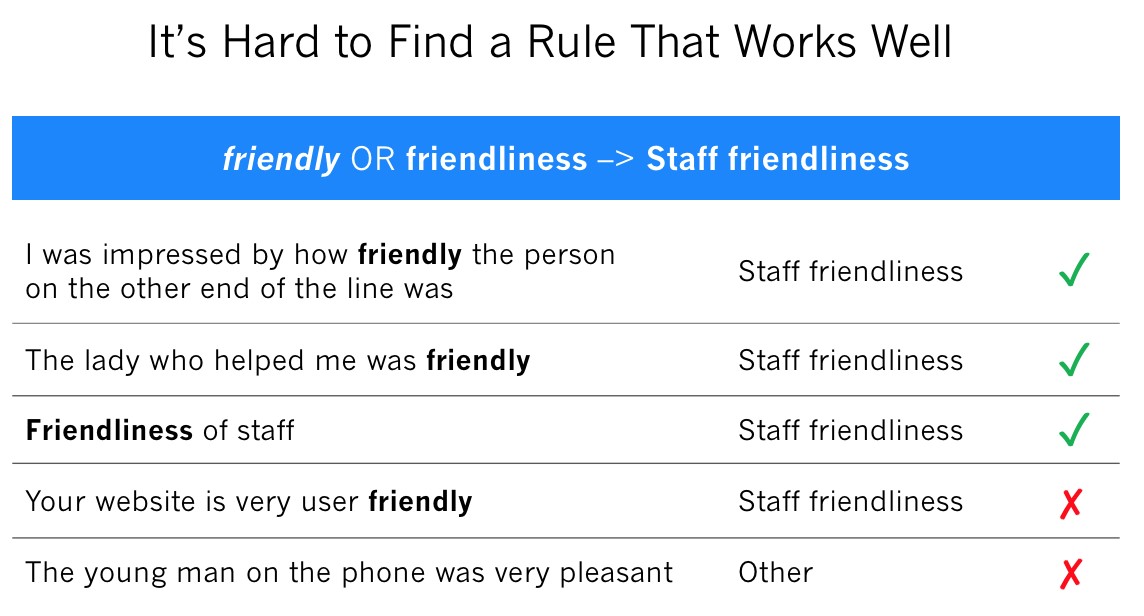
They may also have specific categories setup for certain industries, e.g. banks. And if you are a bank, you just need to add your product names into this taxonomy, and you’re good to go.

The benefit of this approach is that once set up, you can run millions of feedback pieces and get a good overview of the core categories mentioned in the text.

But, there are plenty of disadvantages for this approach, and in fact any manual rules and word spotting technique:

**1. Multiple word meanings make it hard to create rules**

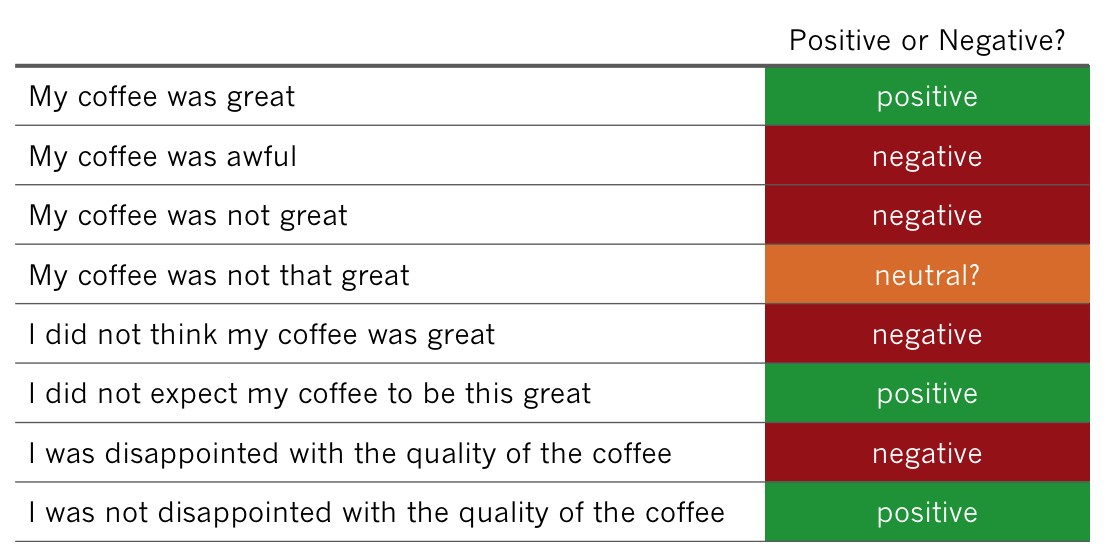
The most common reason why rules fail stems from **polysemy**, when the same word can have different meanings:



**2. Mentioned word != core topic**

Just because a word or a phrase is mentioned in text, it doesn’t always mean that the text is about that topic. For example, when a customer is explaining the situation that leads to an issue: “My credit card got declined and the cashier was super helpful, waiting patiently while I searched for cash in my bag.” This comment is not about credit cards or cash, it’s about the behavior of the staff.

**3. Rules cannot capture sentiment**



Knowing the general category alone isn’t enough. How do people think about “Price”, are they happy or not? Capturing sentiment with manually pre-set rules is impossible. People often do not realize how diverse and varied our language is.

So, a sub-category like “expensive” is actually extremely difficult to model. A person could say something like “I did not think this product was expensive”. To categorize this comment into a category like “good price”, you would need a complex algorithm to detect negation and its scope. A simple regular expression won’t cut it.

**4. Taxonomies don’t exist for software products and many other businesses**

The pre-set taxonomies with rules won’t exist for non-standard products or services. This is particularly problematic for the software industry, where each product is unique and the customer feedback talks about very specific issues

**5. Not everyone can maintain rules**

In any industry, even if you have a working rule-based taxonomy, someone with good linguistic knowledge would need to constantly maintain the rules to make sure all of the feedback is categorized accurately. This person would need to constantly scan for new expressions that people create so easily on the fly, and for any emerging themes that weren’t considered previously. It’s a never-ending process which is highly expensive.

And yet, despite these disadvantages, this approach is the most widely used commercial application of Text Analytics, with its roots in the 90s, and no clear path for fixing these issues.

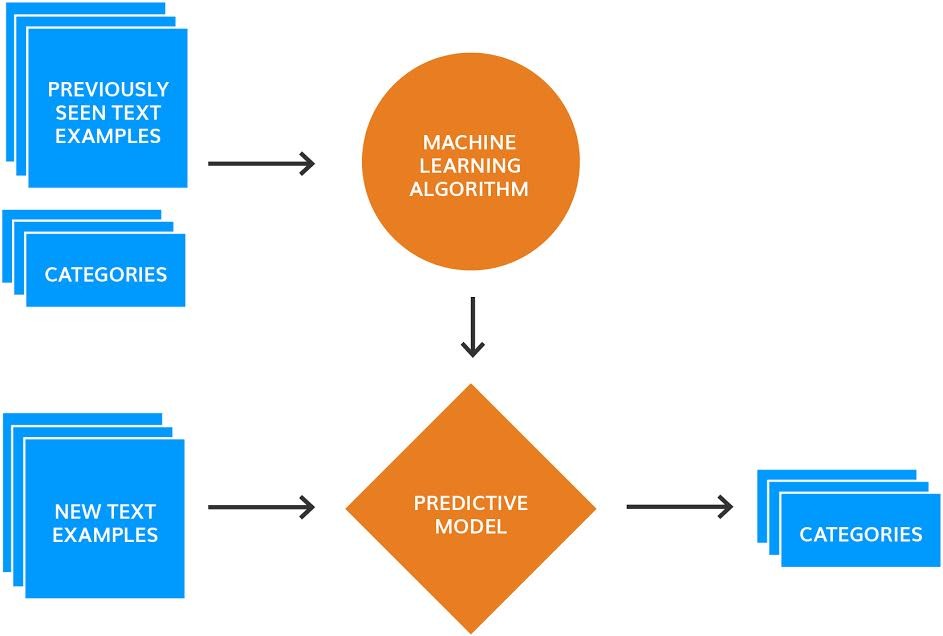
**Text Analytics Approach 3. Text Categorization**

Let’s bring some clarity to the messy subject of **Advanced Text Analytics**, the way it’s pitched by various vendors and data scientists.

Here, we’ll be looking at **Text Categorization**, the first of the three approaches that are actually automated and use algorithms.

**What is text categorization?**

This approach is powered by machine learning. The basic idea is that a machine learning algorithm (there are many) analyzes previously manually categorized examples (the training data) and figures out the rules for categorizing new examples. It’s a supervised approach.



The beauty of text categorization is that you simply need to provide examples, no manual creation of patterns or rules needed, unlike in the two previous approaches.

Another advantage of text categorization is that, theoretically, it should be able to capture the relative importance of a word occurrence in text. Let’s revisit the example from earlier posts. A customer may be explaining the situation that leads to an issue: “My credit card got declined and the cashier was super helpful, waiting patiently while I searched for cash in my bag.” This comment is not about credit cards or cash, it’s about the behaviour of the staff. The theme “credit card” mentioned in the comment isn’t important, but “helpfulness” and “patience” is. A text categorization approach can capture it with the right training.

It all comes down seeing similar examples in the training data.

**Near perfect accuracy… but only with the right training data**

There are academic research papers that show that text categorization can achieve near perfect accuracy. Deep Learning algorithms are even more powerful than the old naïve ones (one older algorithm is actually called Naïve Bayes).

And yet, all researchers agree that **the algorithm isn’t as important as the training data**.

The quality and the amount of the training data is the deciding factor in how successful this approach is for dealing with feedback. So, how much is enough? Well, it depends on the number of categories and the algorithm used to create a categorization model.

The more categories you have and the more closely related they are, the more training data is needed to help the algorithm to differentiate between them.

Some of the newer Text Analytics startups that rely on text categorization provide tools that make it easy for people to train the algorithms, so that they get better over time. **But do you have time to wait for the algorithm to get better, or do you need to act on customer feedback today?**

**Four issues with text categorization**

Apart from needing to train the algorithm, here are four other problems with using text categorization for analyzing people’s feedback:

1. **You won’t notice emerging themes**

You will only learn insights about categories that you trained for and will miss the unknown unknowns. This is the same disadvantage as manual rules and word spotting has: The need to continuously monitor the incoming feedback for emerging themes, and miscategorized items.

1. **Lack of transparency**

While the algorithm gets better over time, it is impossible to understand why it works the way it works and therefore easily tweak the results. Qualitative researchers have told me that the lack of transparency is the main reason why text categorization did not take off in their world. For example, if there is suddenly poor accuracy on differentiating between two themes “wait time to install fiber” and “wait time on the phone to set up fiber”, how much training data does one need to add, until the algorithm stops making these mistakes?

1. **Preparing and managing training data is hard**

The lack of training data is a real issue. It’s hard to start from scratch and most companies don’t have enough or accurate enough data to train the algorithms. In fact, companies always overestimate how much training data they have, which makes implementation fall below expectations. And finally, if you need to refine one specific category, you will need to re-label all of the data from scratch.

1. **Re-training for each new dataset**

Transferability can be really problematic! Imagine you have a working text categorization solution for one of your departments, e.g. support, and now want to analyse feedback that comes through customer surveys, like NPS or CSAT. Again, you would need to re-train the algorithm.

I just got off the phone with a subject matter expert on survey analysis, who told me this story: A team of data scientists spent many months and created a solution that she ultimately had to dismiss due to lack of accuracy. The company did not have time to wait for the algorithm to get better over time.

**Approach  4: Topic Modelling**

**Topic modelling** is also a Machine Learning approach, but an unsupervised one, which means that this approach learns from raw text. Sounds exciting, right?

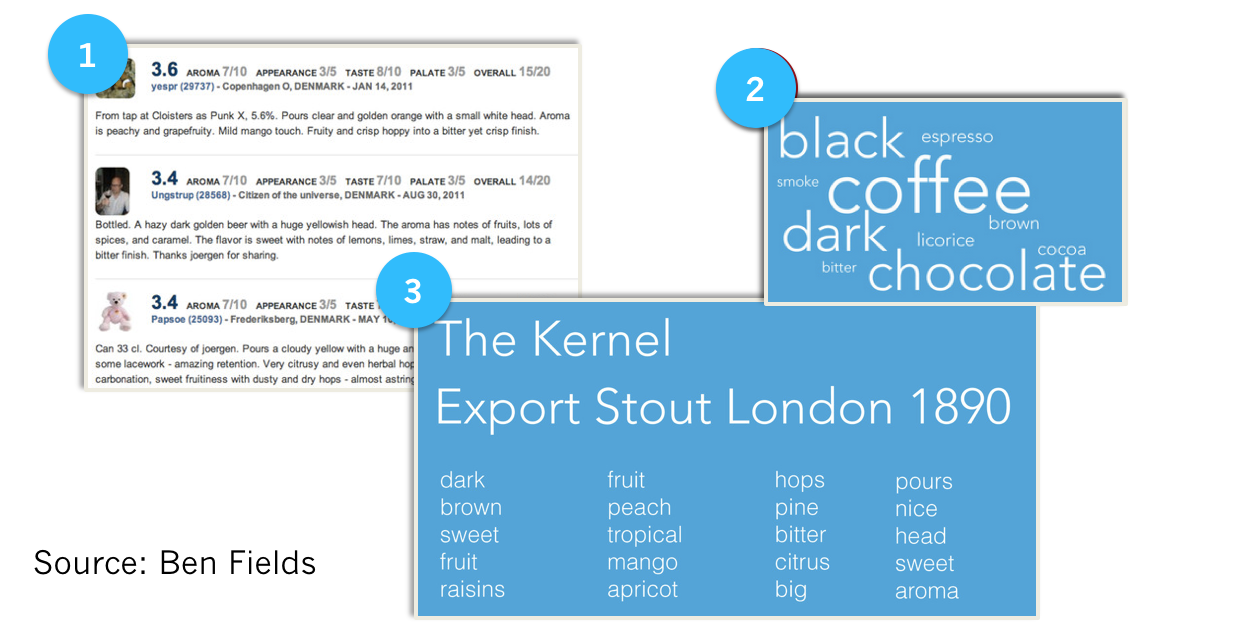
Occasionally, I hear insights professionals refer to any Machine Learning approach as “topic modelling”, but data scientists usually mean a specific algorithm when they say topic modelling.

It’s called LDA, an acronym for the tongue-twisting Latent Dirichlet Allocation. It’s an elegant mathematical model of language that captures topics (lists of similar words) and how they span across various texts.

**Example of topic modelling in action**

Here is an example of [applying topic modelling to beer reviews](https://speakerdeck.com/bfields/sweaty-horse-blanket-processing-the-natural-language-of-beer):

1. The input are reviews of various beers
2. A topic is a collection of similar words like coffee, dark, chocolate, black, espresso
3. Each review is assigned a list of topics. In this example, The Kernel Export stout London has 4 topics assigned to it.

The topics can also be weighted. For example, a customer comment like “your customer support is awful, please get a phone number”, could have weights and topics as following:

* 40% support, service, staff
* 30% bad, poor, awful
* 28% number, phone, email, call

**What’s great about topic modelling**

The best thing about topic modelling is that it needs no input other than the raw customer feedback. As mentioned, unlike text categorization, it’s unsupervised. In simple words, the learning happens by observing which words appear alongside other words in which reviews, and capturing this information using probability statistics. If you are into maths, you will love the concept, explained thoroughly in the corresponding [Wikipedia article](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation), and if those formulas are a bit too much, I recommend [Joyce Xu’s explanation](https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05).

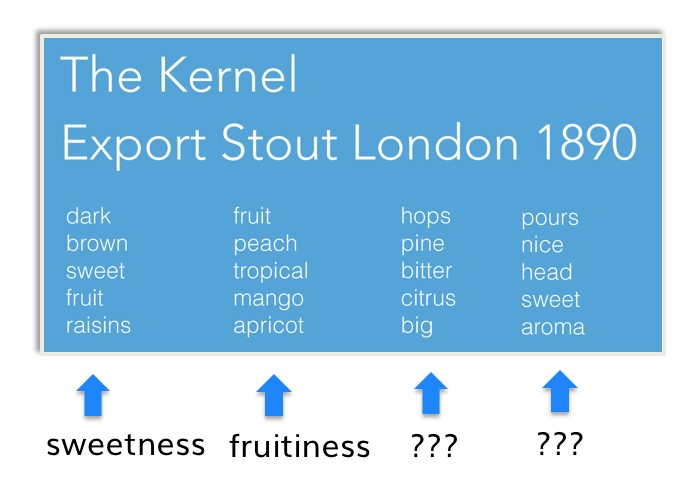
There are Text Analytics startups that use topic modelling to provide analysis of feedback and other text datasets. Other companies, like [StitchFix for example](https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec), use topic modelling to drive product recommendations. They extended traditional topic modelling with a Deep Learning technique called word embeddings. It allows to capture semantics in a more accurate way (more on this in our Part 5).

**Why is topic modelling an inadequate technique for feedback analysis**

When used for feedback analysis, topic modelling has one main disadvantage:

The meaning of the topics is really difficult to interpret

Each topic does capture some aspect of language, but in a non-transparent algorithmic way, which is different from how people understand language. For instance, how would you interpret the second and the fourth topics for the stout beer in the above example:



Whereas the first and the second topic can be somehow “named” as sweetness and fruitiness, the other two topics are just a collection of words.

Any data scientist can put together a solution using public libraries that can quickly spit out a somewhat meaningful output. However, turning this output into charts and graphs that can underpin business decisions is hard. Monitoring how a particular topic changes over time to establish whether the actions taken are working is even harder.

To sum up, because topic modelling produces results that are hard to interpret because it lacks transparency just like text categorization algorithms do, I don’t recommend this approach for analysing feedback. However, I stand by the algorithm as one that can capture language properties fairly well, and one that works really well in other tasks that require Natural Language Understanding.

**Approach 5. Thematic Analysis (plus our secret sauce on how to make it work even better)**

All of the former approaches mentioned have disadvantages. In the best case, you’ll get OK results only after spending many months setting things up. And you may miss out on the unknown unknowns.

The cost of acting late or missing out on crucial insights is huge! It can lead to losing customers and stagnant growth. This is why, according to [YCombinator](http://ycombinator.com) (the startup accelerator that produced more billion dollar companies than any other), “whenever you aren’t working on your product you should be speaking to your users”.

After [Thematic](https://getthematic.com/product) participated in their programme, we’ve been asked for advice three times via a survey, once via a personal email, and also in person. YCombinator also use Thematic to make sense of all the feedback they collect.

**When it comes to customer feedback, three things matter:**

1. Accurate, specific and actionable analysis
2. Ability to see emerging themes fast, without the need of setting things up
3. Transparency in how results are created, to bring in domain expertise and common sense knowledge

In my research, I’ve learned that the only approach that can achieve all three requirements is Thematic Analysis, combined with an interface for easily editing the results.

**Thematic Analysis: How it works**

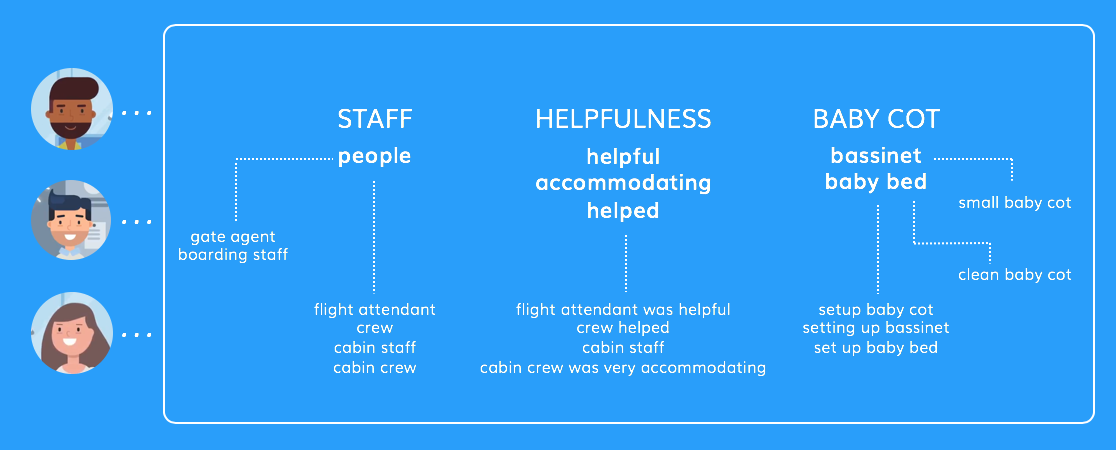
**Thematic Analysis** approaches extract themes from text, rather than categorize text. In other words, it’s a bottom-up analysis. Given a piece of feedback such as “The flight attendant was helpful when I asked to set up a baby cot”, they would extract themes such as “flight attendant”, “flight attendant was helpful”, “helpful”, “asked to set up a baby cot”, and “baby cot”.

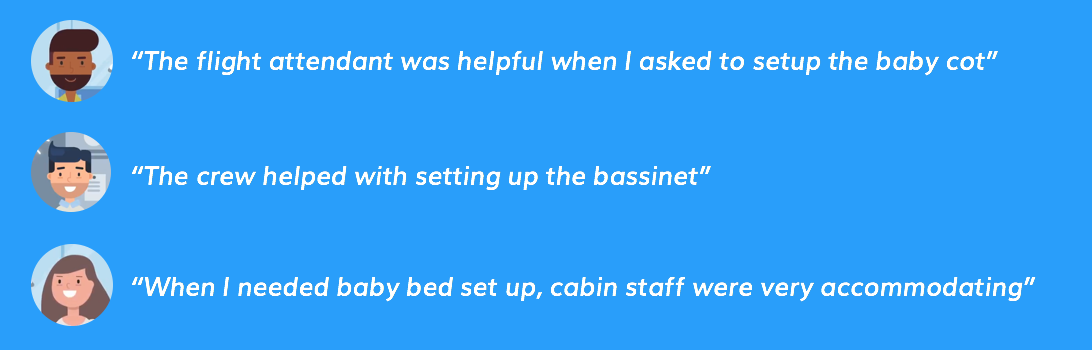


These are all meaningful phrases that can potentially be insightful when analyzing the entire dataset.

However, the most crucial step in a Thematic Analysis approach is merging phrases that are similar into themes and organizing them in a way that’s easy for people to review and edit. We achieve this by using our custom [word embeddings](https://en.wikipedia.org/wiki/Word_embedding) implementation, but there are different ways to achieve this.

For example, here is how three people talk about the same thing, and how we at Thematic group the results into themes and sub-themes:



****

**Advantages and disadvantages of Thematic Analysis**

The advantage of [Thematic Analysis](https://getthematic.com/insights/thematic-analysis-software) is that this approach is unsupervised, meaning that you don’t need to set up these categories in advance, don’t need to train the algorithm, and therefore can easily capture the unknown unknowns.

The disadvantages of this approach are that it’s difficult to implement correctly. A perfect approach must be able to merge and organize themes in a meaningful way, producing a set of themes that are not too generic and not too large. Ideally, the themes must capture at least 80% of verbatims (people’s comments). And the themes extraction must handle complex negation clauses, e.g. “I did not think this was a good coffee”.

**Who does Thematic Analysis?**

Some of the established bigger players have implemented Thematic Analysis to enhance their Manual Rules approaches but tend to produce a laundry list of terms that are hard to review.

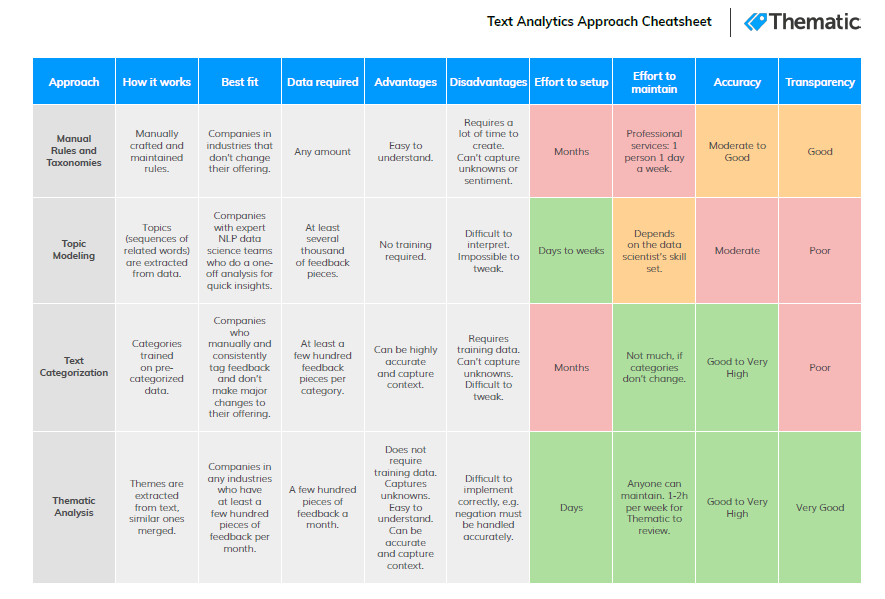
Traditional Text Analytics APIs designed by NLP experts also use this approach. However, they are rarely designed with customer feedback in mind and try to solve this problem in a generic way. For example, when we tested Google and Microsoft’s APIs we found that they aren’t grouping themes out of the box.

As a result, only 20 to 40% of feedback is linked to top 10 themes: only when there are strong similarities in how people talk about specific things. The vast majority of feedback is uncategorized meaning that you can’t slice the data for deeper insights.

At Thematic, we have developed a Thematic Analysis approach that can easily analyze feedback from customers of pizza delivery services, music app creators, real estate brokers and many more. We achieved this by focusing on a specific type of text: customer feedback, unlike NLP APIs that are designed to work on any type of text. We have implemented complex negation algorithms that separate positive from negative themes, to provide better insight.

**Our secret sauce: Human in the loop**

Each dataset, and sometimes even each survey question, gets its own set of themes, and by using our Themes Editor, insights professionals can refine the themes to suit their business. For example, Thematic might find themes such as “fast delivery”, “quick and easy”, “an hour wait”, “slow service”, “delays in delivery” and group them under “speed of service”. One insight professional might re-group these into “slow” and “fast” under “speed of service”, another into “fast service” > “quick and easy”, and “slow service” -> “an hour wait”, “delays in delivery”. It’s a subjective task.



**Applications: Social Media Analytics and Text Mining Mobile Analytics**

**What is social media analytics**

Social media analytics is the ability to gather and find meaning in data gathered from social channels to support business decisions—and measure the performance of actions based on those decisions through social media.

Practitioners and analysts alike know social media by its many websites and channels: Facebook, YouTube, Instagram, Twitter, LinkedIn, Reddit and many others.

Social media analytics is broader than metrics such as likes, follows, retweets, previews, clicks, and impressions gathered from individual channels. It also differs from reporting offered by services that support marketing campaigns such as LinkedIn or Google Analytics.  
  
Social media analytics uses specifically designed software platforms that work similarly to web search tools. Data about keywords or topics is retrieved through search queries or web ‘crawlers’ that span channels. Fragments of text are returned, loaded into a database, categorized and analyzed t derive meaningful in sights. Social media analytics includes the concept of social listening. Listening is monitoring social channels for problems and opportunities. Social media analytics tools typically incorporate listening into more comprehensive reporting that involves listening and performance analysis.

Ebook

Generative AI and ML for the enterprise

Learn key benefits of generative AI and how organizations can incorporate generative AI and

**Why is social media analytics important?**

IBM points out that with the prevalence of social media: “News of a great product can spread like wildfire. And news about a bad product — or a bad experience with a customer service rep — can spread just as quickly. Consumers are now holding organizations to account for their brand promises and sharing their experiences with friends, co-workers and the public at large.”  
  
Social media analytics helps companies address these experiences and use them to:

* Spot trends related to offerings and brands
* Understand conversations — what is being said and how it is being received
* Derive customer sentiment towards products and services
* Gauge response to social media and other communications
* Identify high-value features for a product or service
* Uncover what competitors are saying and its effectiveness
* Map how third-party partners and channels may affect performance

These insights can be used to not only make tactical adjustments, like addressing an angry tweet, they can help drive strategic decisions. In fact, IBM finds social media analytics is now “being brought into the core discussions about how businesses develop their strategies.”  
  
**These strategies affect a range of business activity:**

* **Product development** - Analyzing an aggregate of Facebook posts, tweets and Amazon product reviews can deliver a clearer picture of customer pain points, shifting needs and desired features. Trends can be identified and tracked to shape the management of existing product lines as well as guide new product development.
* **Customer experience** - An IBM study discovered “organizations are evolving from product-led to experience-led businesses.” Behavioral analysis can be applied across social channels to capitalize on micro-moments to delight customers and increase loyalty and lifetime value.  
  Branding - Social media may be the world’s largest focus group. Natural language processing and sentiment analysis can continually monitor positive or negative expectations to maintain brand health, refine positioning and develop new brand attributes.
* **Competitive Analysis** - Understanding what competitors are doing and how customers are responding is always critical. For example, a competitor may indicate that they are foregoing a niche market, creating an opportunity. Or a spike in positive mentions for a new product can alert organizations to market disruptors.
* **Operational efficiency** – Deep analysis of social media can help organizations improve how they gauge demand. Retailers and others can use that information to manage inventory and suppliers, reduce costs and optimize resources.

**Key capabilities of effective social media analytics**

The first step for effective social media analytics is developing a goal. Goals can range from increasing revenue to pinpointing service issues. From there, topics or keywords can be selected and parameters such as date range can be set. Sources also need to be specified — responses to YouTube videos, Facebook conversations, Twitter arguments, Amazon product reviews, comments from news sites. It is important to select sources pertinent to a given product, service or brand.

Typically, a data set will be established to support the goals, topics, parameters and sources. Data is retrieved, analysed and reported through visualizations that make it easier to understand and manipulate.

These steps are typical of a general social media analytics approach that can be made more effective by capabilities found in social media analytics platforms.

* **Natural language processing and machine learning technologies** identify entities and relationships in unstructured data — information not pre-formatted to work with data analytics. Virtually all social media content is unstructured. These technologies are critical to deriving meaningful insights.
* **Segmentation** is a fundamental need in social media analytics. It categorizes social media participants by geography, age, gender, marital status, parental status and other demographics. It can help identify influencers in those categories. Messages, initiatives and responses can be better tuned and targeted by understanding who is interacting on key topics.
* **Behaviour analysis** is used to understand the concerns of social media participants by assigning behavioural types such as user, recommender, prospective user and detractor. Understanding these roles helps develop targeted messages and responses to meet, change or deflect their perceptions.
* **Sentiment analysis** measures the tone and intent of social media comments. It typically involves natural language processing technologies to help understand entities and relationships to reveal positive, negative, neutral or ambivalent attributes.
* **Share of voice** analyses prevalence and intensity in conversations regarding brand, products, services, reputation and more. It helps determine key issues and important topics. It also helps classify discussions as positive, negative, neutral or ambivalent.
* **Clustering analysis** can uncover hidden conversations and unexpected insights. It makes associations between keywords or phrases that appear together frequently and derives new topics, issues and opportunities. The people that make baking soda, for example, discovered new uses and opportunities using clustering analysis.
* **Dashboards and visualization** charts, graphs, tables and other presentation tools summarize and share social media analytics findings — a critical capability for communicating and acting on what has been learned. They also enable users to grasp meaning and insights more quickly and look deeper into specific findings without advanced technical skills.

**Text mining applications**

Text mining applications When combined with text analytics tools, feedback systems, such as chatbots, customer surveys, NPS (net-promoter scores), online reviews, support tickets, and social media profiles, enable companies to improve their customer experience with speed.

Text mining can be used in a variety of applications, including:

* **Customer insights**

Text mining can help businesses understand customer preferences, opinions, and sentiments. It can also help identify new features to improve products and customer experience.

* **Customer service**

Text mining can help streamline customer service processes by automatically routing tickets and improving responsiveness.

* **Market research**

Text mining can help businesses understand consumer behavior and conduct market research.

* **Brand reputation management**

Text mining can help businesses manage their brand reputation on social media platforms.

* **Targeted marketing**

Text mining can help businesses segment audiences based on their interests, behaviors, and preferences.

* **Influencer identification**

Text mining can help businesses identify influencers and thought leaders in specific industries.

* **Crisis management**

Text mining can help businesses identify potential crises and manage risks.

* **Digital advertising**

Text mining can help businesses show targeted advertisements to users based on what they type, view, or do online.

* **Scientific advancements**

Text mining can help scholars explore large amounts of literature and find information relevant to their research.

Text mining is a technique that uses advanced analytical techniques to analyze large collections of textual materials to capture key concepts, trends, and hidden relationships.