1. **Discuss the types of 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 as well as a large amount of data we can make use of 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 which can help us derive valuable insights to make business predictions

**Types of Data Analytics**

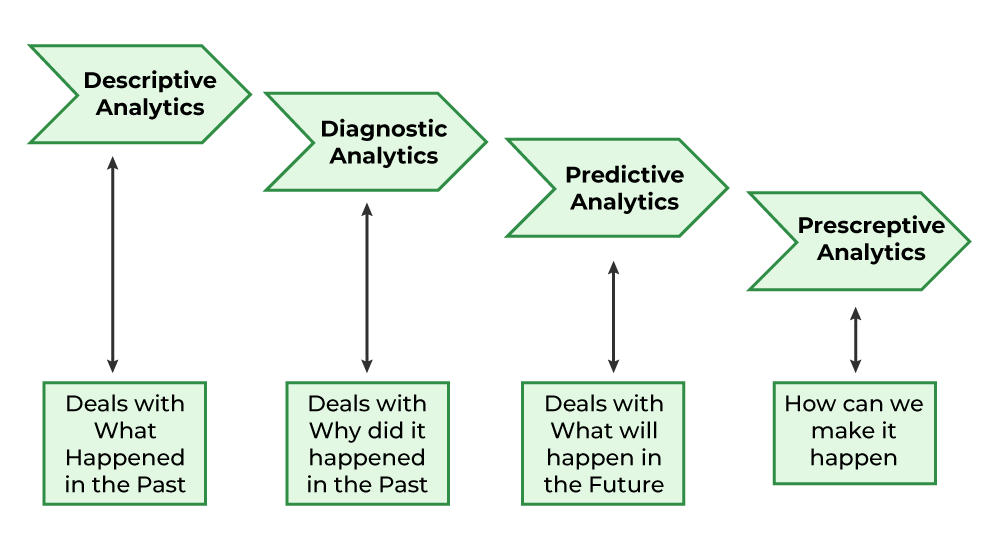
There are four major types of data analytics:

1. Predictive (forecasting)

2. Descriptive (business intelligence and data mining)

3. Prescriptive (optimization and simulation)

4. Diagnostic analytics



1. **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, data mining, and 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 Corner Stones of Predictive Analytics

• Predictive modeling

• Decision Analysis and optimization

• Transaction profiling

1. **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 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

1. **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 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.

1. **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

1. **Describe in detail about 4V’s?**

**Volume**

Big data is a form of data whose volume is so large that it would not fit on a single machine therefore specialized tools and frameworks are required to store process and analyze such data. For example, social media applications process billions of messages everyday, industrial and energy systems can generate terabytes of sensor data everyday, cab aggregation applications can process millions of transactions in a day, etc. The volumes of data generated by modern IT, industrial, healthcare, Internet of Things, and other systems is growing exponentially

driven by the lowering costs of data storage and processing architectures and the need to

extract valuable insights from the data to improve business processes, efficiency and service

to consumers. Though there is no fixed threshold for the volume of data to be considered as

big data, however, typically, the term big data is used for massive scale data that is difficult to store, manage and process using traditional databases and data processing architectures.

**Velocity**

Velocity of data refers to how fast the data is generated. Data generated by certain sources can arrive at very high velocities, for example, social media data or sensor data. Velocity

is another important characteristic of big data and the primary reason for the exponential

growth of data. High velocity of data results in the volume of data accumulated to become

very large, in short span of time. Some applications can have strict deadlines for data analysis

(such as trading or online fraud detection) and the data needs to be analyzed in real-time.

Specialized tools are required to ingest such high velocity data into the big data infrastructure and analyse the data in real-time.

**Variety**

Variety refers to the forms of the data. Big data comes in different forms such as structured, unstructured or semi-structured, including text data, image, audio, video and sensor data. Big data systems need to be flexible enough to handle such variety of data.

**Veracity**

Veracity refers to how accurate is the data. To extract value from the data, the data needs to

be cleaned to remove noise. Data-driven applications can reap the benefits of big data only

when the data is meaningful and accurate. Therefore, cleansing of data is important so that

incorrect and faulty data can be filtered out.

**Value**

Value of data refers to the usefulness of data for the intended purpose. The end goal of any

big data analytics system is to extract value from the data. The value of the data is also

related to the veracity or accuracy of the data. For some applications value also depends on

how fast we are able to process the data.

1. **Define the following terms for Big Data**
2. **Structured data**
3. **Semi structured data**
4. **Unstructured data**

Big Data includes huge volume, high velocity, and extensible variety of data. There are 3 types: Structured data, Semi-structured data, and Unstructured data.

1. **Structured data –**

Structured data is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database. It concerns all data which can be stored in database SQL in a table with rows and columns. They have relational keys and can easily be mapped into pre-designed fields. Today, those data are most processed in the development and simplest way to manage information. Example: Relational data.

1. **Semi-Structured data –**

Semi-structured data is information that does not reside in a relational database but that has some organizational properties that make it easier to analyze. With some processes, you can store them in the relation database (it could be very hard for some kind of semi-structured data), but Semi-structured exist to ease space. Example: XML data.

1. **Unstructured data –**

Unstructured data is a data which is not organized in a predefined manner or does not have a predefined data model, thus it is not a good fit for a mainstream relational database. So for Unstructured data, there are alternative platforms for storing and managing, it is increasingly prevalent in IT systems and is used by organizations in a variety of business intelligence and analytics applications. Example: Word, PDF, Text, Media logs.

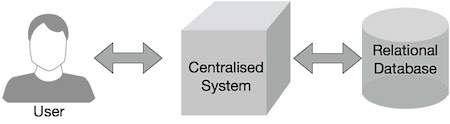
**Differences between Structured, Semi-structured and Unstructured data:**

| **Properties** | **Structured data** | **Semi-structured data** | **Unstructured data** |
| --- | --- | --- | --- |
| Technology | It is based on Relational database table | It is based on XML/RDF(Resource Description Framework). | It is based on character and binary data |
| Transaction management | Matured transaction and various concurrency techniques | Transaction is adapted from DBMS not matured | No transaction management and no concurrency |
| Version management | Versioning over tuples,row,tables | Versioning over tuples or graph is possible | Versioned as a whole |
| Flexibility | It is schema dependent and less flexible | It is more flexible than structured data but less flexible than unstructured data | It is more flexible and there is absence of schema |
| Scalability | It is very difficult to scale DB schema | It’s scaling is simpler than structured data | It is more scalable. |
| Robustness | Very robust | New technology, not very spread | — |
| Query performance | Structured query allow complex joining | Queries over anonymous nodes are possible | Only textual queries are possible |

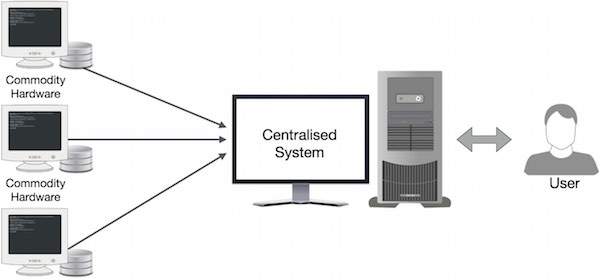
1. **Illustrate Map Reduce with example?**

**MapReduce:-**

Traditional Enterprise Systems normally have a centralized server to store and process data. The following illustration depicts a schematic view of a traditional enterprise system. Traditional model is certainly not suitable to process huge volumes of scalable data and cannot be accommodated by standard database servers. Moreover, the centralized system creates too much of a bottleneck while processing multiple files simultaneously.



Google solved this bottleneck issue using an algorithm called MapReduce. MapReduce divides a task into small parts and assigns them to many computers. Later, the results are collected at one place and integrated to form the result dataset.



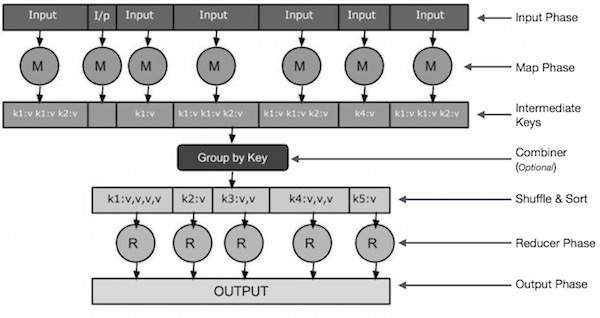
**Working Process:-**

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

* The Map task takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key-value pairs).
* The Reduce task takes the output from the Map as an input and combines those data tuples (key-value pairs) into a smaller set of tuples.

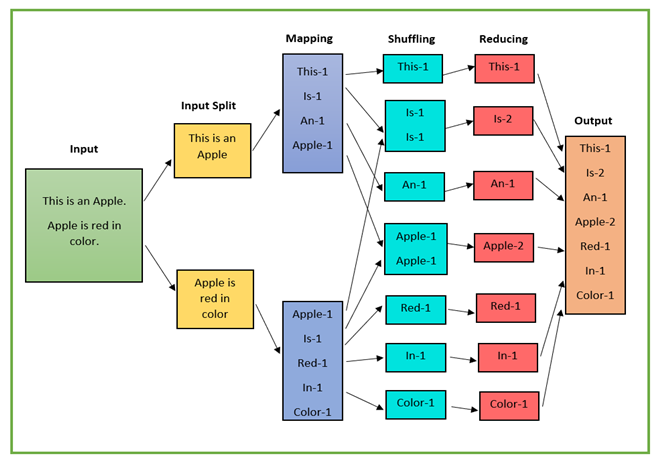
The reduce task is always performed after the map job.

Let us now take a close look at each of the phases and try to understand their significance.



* **Input Phase** − Here we have a Record Reader that translates each record in an input file and sends the parsed data to the mapper in the form of key-value pairs.
* **Map** − Map is a user-defined function, which takes a series of key-value pairs and processes each one of them to generate zero or more key-value pairs.
* **Intermediate Keys** − They key-value pairs generated by the mapper are known as intermediate keys.
* **Combiner** − A combiner is a type of local Reducer that groups similar data from the map phase into identifiable sets. It takes the intermediate keys from the mapper as input and applies a user-defined code to aggregate the values in a small scope of one mapper. It is not a part of the main MapReduce algorithm; it is optional.
* **Shuffle and Sort** − The Reducer task starts with the Shuffle and Sort step. It downloads the grouped key-value pairs onto the local machine, where the Reducer is running. The individual key-value pairs are sorted by key into a larger data list. The data list groups the equivalent keys together so that their values can be iterated easily in the Reducer task.
* **Reducer** − The Reducer takes the grouped key-value paired data as input and runs a Reducer function on each one of them. Here, the data can be aggregated, filtered, and combined in a number of ways, and it requires a wide range of processing. Once the execution is over, it gives zero or more key-value pairs to the final step.
* **Output Phase** − In the output phase, we have an output formatter that translates the final key-value pairs from the Reducer function and writes them onto a file using a record writer.

**Example:-**



* The input data is divided into multiple segments, then processed in parallel to reduce processing time. In this case, the input data will be divided into two input splits so that work can be distributed over all the map nodes.
* The Mapper counts the number of times each word occurs from input splits in the form of key-value pairs where the key is the word, and the value is the frequency.
* For the first input split, it generates 4 key-value pairs: This, 1; is, 1; an, 1; apple, 1; and for the second, it generates 5 key-value pairs: Apple, 1; is, 1; red, 1; in, 1; color.
* It is followed by the shuffle phase, in which the values are grouped by keys in the form of key-value pairs. Here we get a total of 6 groups of key-value pairs.
* The same reducer is used for all key-value pairs with the same key.
* All the words present in the data are combined into a single output in the reducer phase. The output shows the frequency of each word.
* Here in the example, we get the final output of key-value pairs as This, 1;  is, 2; an, 1; apple, 2; red, 1; in, 1;  color, 1.
* The record writer writes the output key-value pairs from the reducer into the output files, and the final output data is by default stored on HDFS.

1. **Explain the techniques used in Big Data Analytics?**

## TECHNIQUES FOR ANALYZING BIG DATA

There are several methods that can be used to analyze datasets based on disciplines such as computer science (particularly machine learning) and statistics.

### ****A/B testing****

A method in which a control group is compared with a variety of test groups in order to determine what changes will improve a given objective variable, e.g., marketing response rate. A/B testing is sometimes called split testing or bucket testing. You can use this to determine what type of copy, layout, image, or color will increase conversion rates to a Website for e-commerce. If more than one variable is simultaneously manipulated in the treatment, the multivariate generalization of this method, which applies statistical modeling, is called "A/B/N" testing.

### ****Association rule learning****

A set of methods for finding interesting relationships, i.e., "association rules," among variables in massive databases. To generate and test these rules, different algorithms are used. An example is market basket analysis, in which a retailer determines which products are frequently purchased together and uses that information for marketing Used for data mining.

### ****Classification****

These techniques are used to identify categories in which new data points belong based on data points that have already been categorized in a training set. It can be used, for instance, to predict segment-specific customer behavior (e.g., buying decisions, churn rate, consumption rate) for which there is a clear hypothesis or objective outcome. Used for data mining.

### ****Cluster analysis****

Statistical method for classifying objects based on similarities among diverse groups of objects, but without knowing in advance what characteristics make them similar. Segmenting consumers into similar groups for targeted marketing is an example of cluster analysis. Used for data mining

### ****Crowdsourcing****

Crowdsourcing is a method for collecting data submitted by a large group of people or crowd through an open call, usually through a networked medium such as the Internet. In this case, you're utilizing Web 2.0 technology and mass collaboration.

### ****Data fusion and data integration****

The process of integrating and analyzing data from multiple sources in order to uncover insights in a more efficient and accurate manner than if they were developed from just one source.

### ****Data mining****

Statistical and machine learning approaches are combined with database management to extract patterns from large datasets. These techniques include cluster analysis, association rule learning, regression, and classification. Customer data can be analyzed to identify segments that respond most quickly to an offer, employee data can be analyzed to identify attributes of the most successful employees, or market basket analysis can be used to predict what customers will purchase.

### ****Ensemble learning****

Multiple predictive models (each constructed using statistics and/or machine learning) are used to achieve better performance than any of the constituent models.

### ****Genetic algorithms****

This optimization technique involves encoding potential solutions as "chromosomes" that can combine and mutate in the same way as a natural evolution. Every chromosome is selected for survival within the context of a modeled "environment," determining its fitness or performance. These algorithms are often described as types of "evolutionary algorithms," which are ideal for solving nonlinear problems. A few examples include optimizing the performance of an investment portfolio and improving job scheduling in manufacturing.

### ****Machine learning****

Data-driven machine learning focuses on automatically recognizing complex patterns and making intelligent decisions from them.

### ****Neural networks****

Finding patterns in data using computational models inspired by the structure and workings of biological neural networks (such as the cells and connections found in the brain). Finding nonlinear patterns is a good application for neural networks.

### ****Network analysis****

An analysis technique for describing relationships among discrete nodes in a graph or a network. The social network analysis aims to investigate the connections among individuals in a group or organization, for example, how information moves or who has the most influence. The technology can be used, for instance, to identify key opinion leaders to target in marketing and to identify bottlenecks in information flows within enterprises.

### ****Optimization****

Numerical methods for redesigning complex systems and processes to improve their performance based on one or more objective metrics (e.g., cost, speed, or reliability). Among the applications for optimization are improving operational processes such as scheduling, routing, and floor planning, as well as formulating strategies such as product range strategy, linked investment analysis, and R&D portfolio strategy.

### ****Sentiment analysis****

The process of extracting and identifying subjective information from a text source using natural language processing and analytic techniques. This type of analysis includes identifying the product or feature about which a sentiment is expressed, determining the type of sentiment (e.g., positive, negative, neutral), and determining the degrees and strength of sentiment. The application of sentiment analysis in social media (e.g., blogs, microblogs, and social networks) may allow companies to measure how different customer segments and stakeholders are responding to their products and actions.

### ****Spatial analysis****

An analysis of topological, geometric, or geographical properties encoded in a data set using various techniques drawn from statistics. Spatial analysis often uses geographical information systems (GIS) that capture locational information, e.g., address and longitude/latitude coordinates. Spatial data can be incorporated into spatial regressions (for instance, how is consumer willingness correlated with location?) or simulations (for example, how would a manufacturing supply chain network work with sites located in different places?).

### ****Statistics****

This includes the study of survey methods and experiments, as well as how data is collected, organized, and interpreted. It is common for statistical techniques to be used to determine what relationships between variables could have been the result of chance (the "null hypothesis") and which relationships likely result from underlying causal relationships (i.e., those that are statistically significant).

### ****Time series analysis****

Combining statistical and signal processing techniques for analyzing sets of data points, each representing a value at a different time, in order to extract properties that are meaningful. For example, a time series analysis might look at the daily price of stock market or amount of patients diagnosed with a given condition on a daily basis. Forecasting time series involves using a mathematical model to predict future values of the series based on data from past values.

### ****Visualization****

Often used to synthesize data analysis results using images, diagrams, or animations in order to communicate a message.

## BIG DATA TECHNOLOGIES

Various technologies are available for aggregating, manipulating, managing, and analyzing big data. The list we have provided here contains some of the most prominent technologies, but not all, especially since more technologies continue to be developed for big data applications, some of which we have listed.

### ****Big Table****

Based on the Google File System, a proprietary distributed database system. Inspiration for HBase.

### ****Cassandra****

Database management system composed of open-source programs designed to handle large volumes of data over a distributed network. Initially developed at Facebook, this system is now managed by the Apache Software Foundation.

### ****Cloud computing****

In this paradigm, highly scalable computing resources, usually configured as distributed systems, are provided as services over the network.

### ****Data warehouse****

A database designed specifically for reporting is usually used to store large volumes of structured data. ETL (extract, transform, load) tools are used to upload data from operational data stores, and business intelligence tools are often used to generate reports.

### ****Distributed system****

Multiple computers, communicating through a network, are used to solve a common computational problem. Computers working in parallel solve each of the problems divided into multiple tasks. A distributed system provides better performance at a lower cost (i.e., because a cluster of lower-end computers can be cheaper than a single higher-end computer), as well as greater reliability (since there is no one point of failure) and scalability (since a distributed system can be made more powerful by simply adding more nodes rather than replacing a central computer).

### ****Extract, transform, and load (ETL)****

Extracting data from the outside sources, transforming it to fit operational needs, and storing it in databases or data warehouses.

### ****Hadoop****

The software framework used to process large datasets on a distributed system for particular types of problems. MapReduce framework and Google File System provide inspiration for the development of this system.

### ****MapReduce****

Google introduced a software framework for processing large datasets on certain types of problems on a distributed system.

### ****Stream processing****

Technology for processing large streams of event data in real-time. In addition to algorithmic trading, stream processing enables applications including RFID event processing applications, criminal analysis, process monitoring, and location-based services in telecommunications.

1. **Generalize how the data flow takes places in map-reduce framework?**

MapReduce is a popular programming model for data intensive applications. MapReduce

has been used for batch analysis of data in a wide range of applications (such as social networking, e-Commerce, finance, entertainment, government, healthcare, telecom, etc.). MapReduce allows the developers to focus on developing data-intensive applications without having to worry about issues such as input splitting, scheduling, load balancing and failover.

* MapReduce programming model for processing data on large clusters was originally

proposed by Dean and Ghemawat.

* Hadoop, which is an open source large-scale distributed batch processing framework, provides an implementation of the MapReduce model.
* Hadoop and MapReduce have made it easier for developing scalable and data intensive applications for cloud computing environments.

**MapReduce model has two phases:** Map and Reduce. MapReduce programs are written in a functional programming style to create Map and Reduce functions.

* The input data to the map and reduce phases is in the form of key-value pairs. Run-time systems for MapReduce are typically large clusters built of commodity hardware.
* The MapReduce run-time systems take care of tasks such partitioning the data, scheduling of jobs and communication between nodes in the cluster. This makes it easier for programmers to analyse massive scale data without worrying about tasks such as data partitioning and scheduling.

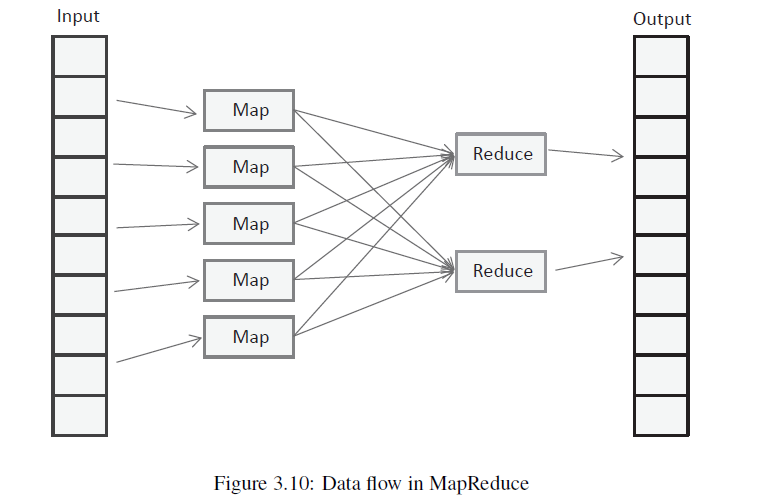


Figure 3.10 shows the flow of data for a MapReduce job. MapReduce programs take a set of input key-value pairs and produce a set of output key-value pairs.

* In the Map phase, data is read from a distributed file system, partitioned among a set of computing nodes in the cluster, and sent to the nodes as a set of key-value pairs. The Map tasks process the input records independently of each other and produce intermediate results as key-value pairs.
* The intermediate results are stored on the local disk of the node running the Map task. The choice of the key and value pairs at this step depends on the data analysis task to be accomplished.
* When all the Map tasks are completed, the Reduce phase begins with the shuffle and sort

step, in which the intermediate data is sorted by the key and the key-value pairs are grouped and shuffled to the reduce tasks.

* The reduce tasks then take the key-value pairs grouped by the key and run the reduce function for each group of key-value pairs. The data processing logic in reduce function depends on the analysis task to be accomplished.
* An optional Combine task can be used to perform data aggregation on the intermediate data of the same key for the output of the mapper before transferring the output to the Reduce task.

1. **Describe how map-reduce computation executes?**

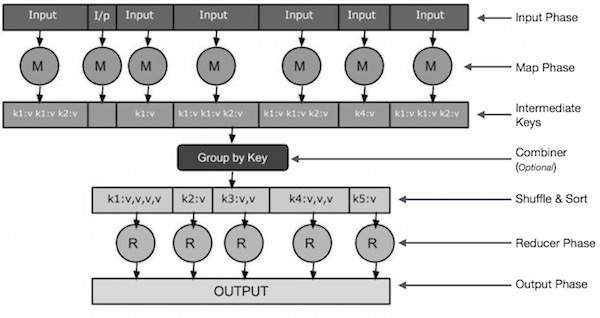
**Working Process:-**

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

* The Map task takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key-value pairs).
* The Reduce task takes the output from the Map as an input and combines those data tuples (key-value pairs) into a smaller set of tuples.

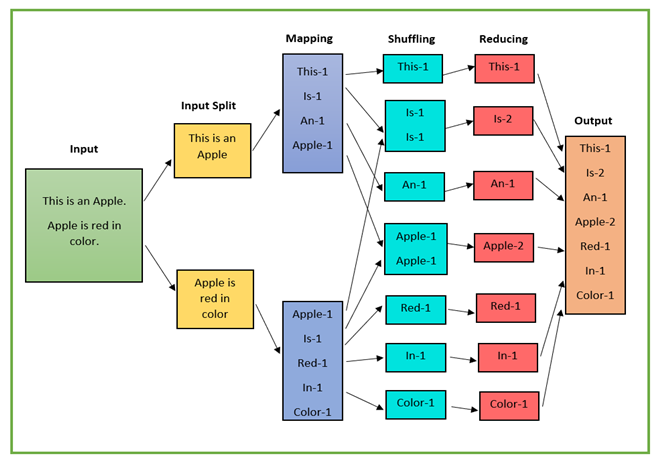
The reduce task is always performed after the map job.

Let us now take a close look at each of the phases and try to understand their significance.



* **Input Phase** − Here we have a Record Reader that translates each record in an input file and sends the parsed data to the mapper in the form of key-value pairs.
* **Map** − Map is a user-defined function, which takes a series of key-value pairs and processes each one of them to generate zero or more key-value pairs.
* **Intermediate Keys** − They key-value pairs generated by the mapper are known as intermediate keys.
* **Combiner** − A combiner is a type of local Reducer that groups similar data from the map phase into identifiable sets. It takes the intermediate keys from the mapper as input and applies a user-defined code to aggregate the values in a small scope of one mapper. It is not a part of the main MapReduce algorithm; it is optional.
* **Shuffle and Sort** − The Reducer task starts with the Shuffle and Sort step. It downloads the grouped key-value pairs onto the local machine, where the Reducer is running. The individual key-value pairs are sorted by key into a larger data list. The data list groups the equivalent keys together so that their values can be iterated easily in the Reducer task.
* **Reducer** − The Reducer takes the grouped key-value paired data as input and runs a Reducer function on each one of them. Here, the data can be aggregated, filtered, and combined in a number of ways, and it requires a wide range of processing. Once the execution is over, it gives zero or more key-value pairs to the final step.
* **Output Phase** − In the output phase, we have an output formatter that translates the final key-value pairs from the Reducer function and writes them onto a file using a record writer.

**Example:-**



* The input data is divided into multiple segments, then processed in parallel to reduce processing time. In this case, the input data will be divided into two input splits so that work can be distributed over all the map nodes.
* The Mapper counts the number of times each word occurs from input splits in the form of key-value pairs where the key is the word, and the value is the frequency.
* For the first input split, it generates 4 key-value pairs: This, 1; is, 1; an, 1; apple, 1; and for the second, it generates 5 key-value pairs: Apple, 1; is, 1; red, 1; in, 1; color.
* It is followed by the shuffle phase, in which the values are grouped by keys in the form of key-value pairs. Here we get a total of 6 groups of key-value pairs.
* The same reducer is used for all key-value pairs with the same key.
* All the words present in the data are combined into a single output in the reducer phase. The output shows the frequency of each word.
* Here in the example, we get the final output of key-value pairs as This, 1;  is, 2; an, 1; apple, 2; red, 1; in, 1;  color, 1.
* The record writer writes the output key-value pairs from the reducer into the output files, and the final output data is by default stored on HDFS.

1. **Define HDFS? Explain in detail?**

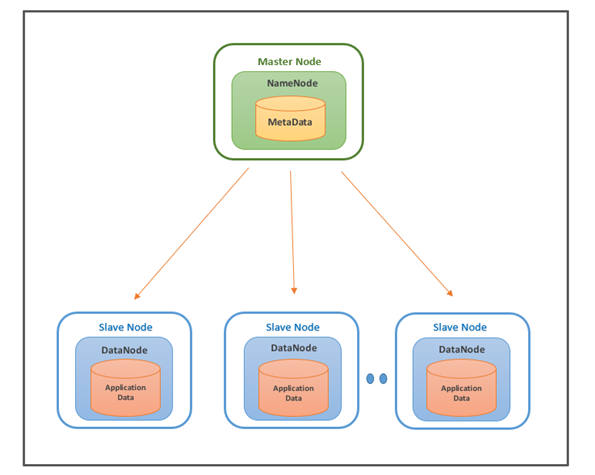
Hadoop Distributed File System (HDFS) is the primary data storage system used by Hadoop applications. To implement a distributed file system that provides high-performance access to data across highly scalable Hadoop clusters, HDFS uses the NameNode and DataNode architecture. Apache Hadoop is an open-source framework for managing data processing and storage for big data applications. HDFS is a crucial part of the Hadoop ecosystem. It can manage big data pools and support big data analytics applications.

**Components of HDFS**

HDFS has two components, which are as follows:

1) Namenode

2) Datanode



#### **Namenode**

The NameNode is the master node of HDFS and stores the metadata and slave configuration. In HDFS, there is one active NameNode and one or more standby NameNodes. The Active NameNode serves all client requests, and the standby NameNode handles high availability configuration.

**Functions of NameNode:**

* It manages the File system namespace and is the single Hadoop cluster failure point.
* It keeps track of all blocks in HDFS and where each block is located.
* It manages the client access requests for the actual data files.
* The metadata about the actual data is also stored here, like File information, Block information, permissions, etc.

#### **DataNode**

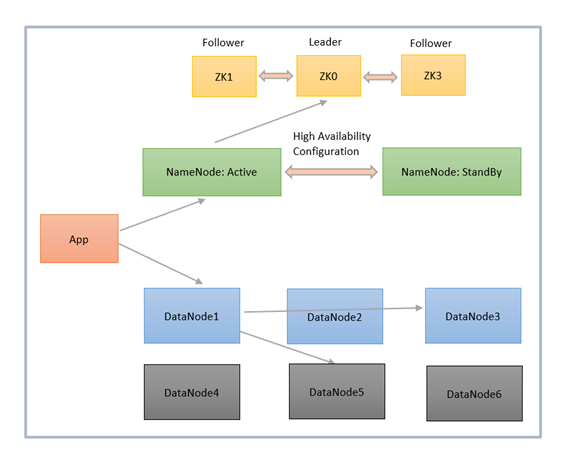
DataNode is a worker node of HDFS, which can be n in number. This node is responsible for serving read and write requests to clients. Datanodes store actual data in HDFS, so they typically have a lot of hard disk space.

**Functions of DataNode:**

* It stores actual data in HDFS.
* As instructed by the NameNode, the DataNodes are responsible for storing and deleting blocks and replicating these blocks.
* This node handles the client’s read and writes requests.
* DataNodes are synchronized to communicate and ensure that data is balanced across the cluster, moving data for high replication, and copying data as needed.

**Working of HDFS**

* HDFS is based on master-slave architecture. It can consist of hundreds or thousands of servers. Applications such as SQoop, NIFI, and Kafka Connect write data to HDFS. The client communicates with the NameNode for metadata, and the NameNode replies with the location of blocks, the number of replicas, and other attributes.
* NameNode sends the High Availability Configuration information to Zookeeper, and it replicates the information to multiple Zookeepers. Zookeepers serve as an election commissioner by selecting one StandBy NameNode when an Active NameNode is down, and there are multiple StandBy NameNodes.
* Using NameNode information, the client contacts the DataNode directly. Based on the information received from the NameNode, the client will start writing data directly to data nodes and reading data from the DataNodes in parallel.



**Benefits of HDFS**

The benefits of the Hadoop Distributed File System are as follows:

1) The Hadoop Distributed File System is designed for big data, not only for storing big data but also for facilitating the processing of big data.

2) HDFS is cost-effective because it can be run on cheap hardware and does not require a powerful machine.

3) HDFS has high fault tolerance since if a machine within a cluster fails, a replica of the data may be available from a different node through replication.

4) Hadoop is famous for its rack awareness to avoid data loss, which results in increased latency.

5) HDFS is scalable, and it includes vertical and horizontal scalability mechanisms so you can adjust the resources according to the size of your file system.

6) Streaming reads are made possible through HDFS.

1. **Explain the significances of Hadoop Distributed File System and its application?**

**Significance of HDFS:-**

* [**Data replication**](https://www.techtarget.com/searchdisasterrecovery/tutorial/Data-replication-technologies-and-disaster-recovery-planning-tutorial)**.** This is used to ensure that the data is always available and prevents data loss. For example, when a node crashes or there is a hardware failure, replicated data can be pulled from elsewhere within a cluster, so processing continues while data is recovered.
* **Fault tolerance and reliability.** HDFS' ability to replicate file blocks and store them across nodes in a large cluster ensures fault tolerance and reliability.
* **High availability.**As mentioned earlier, because of replication across notes, data is available even if the NameNode or a DataNode fails.
* **Scalability.** Because HDFS stores data on various nodes in the cluster, as requirements increase, a cluster can scale to hundreds of nodes.
* **High throughput.** Because HDFS stores data in a distributed manner, the data can be processed in parallel on a cluster of nodes. This, plus data locality (see next bullet), cut the processing time and enable high throughput.
* **Data locality.** With HDFS, computation happens on the DataNodes where the data resides, rather than having the data move to where the computational unit is. By minimizing the distance between the data and the computing process, this approach decreases network congestion and boosts a system's overall throughput.
* **Cost effectiveness.** The DataNodes that store the data rely on inexpensive off-the- shelf hardware, which cuts storage costs. Also, because HDFS is open source, there's no licensing fee.
* **Large data set storage.**HDFS stores a variety of data of any size -- from megabytes to [petabytes](https://www.techtarget.com/searchstorage/definition/petabyte) -- and in any format, including structured and unstructured data.
* **Fast recovery from hardware failure.** HDFS is designed to detect faults and automatically recover on its own.
* **Portability.**HDFS is portable across all hardware platforms, and it is compatible with several operating systems, including Windows, Linux and Mac OS/X.
* **Streaming data access.** HDFS is built for high data throughput, which is best for access to streaming data.

**Applications of HDFS:--**

1. **Electric companies.** The power industry deploys phasor measurement units (PMUs) throughout their transmission networks to [monitor the health of smart grids](https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/protection-relays-and-control/general-protection/phasor-measurement-unit-pmu.html). These high-speed sensors measure current and voltage by amplitude and phase at selected transmission stations. These companies analyze PMU data to detect system faults in network segments and adjust the grid accordingly. For instance, they might switch to a backup power source or perform a load adjustment. PMU networks clock thousands of records per second, and consequently, power companies can benefit from inexpensive, highly available file systems, such as HDFS.
2. **Marketing.** Targeted marketing campaigns depend on marketers knowing a lot about their target audiences. Marketers can get this information from several sources, including [CRM](https://www.techtarget.com/searchcustomerexperience/definition/CRM-customer-relationship-management) systems, direct mail responses, point-of-sale systems, Facebook and Twitter. Because much of this data is unstructured, an HDFS cluster is the most cost-effective place to put data before analyzing it.
3. **Oil and gas providers.**Oil and gas companies deal with a variety of data formats with very large data sets, including videos, 3D earth models and machine sensor data. An HDFS cluster can provide a suitable platform for the big data analytics that's needed.
4. **Research.**Analyzing data is a key part of research, so, here again, HDFS clusters provide a cost-effective way to store, process and analyze large amounts of data.
5. **What is NoSQL? What are the advantages of NoSQL? Explain the types of NoSQL databases?**

**NoSQL** Database is a non-relational Data Management System, that does not require a fixed schema. It avoids joins, and is easy to scale. The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs. NoSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook and Google collect terabytes of user data every single day.

**NoSQL database** stands for “Not Only SQL” or “Not SQL.” Though a better term would be “NoREL”, NoSQL caught on. Carl Strozz introduced the NoSQL concept in 1998.

**Advantages of NoSQL**

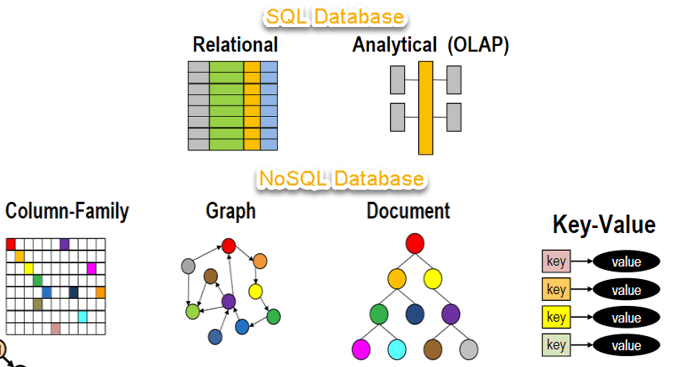
* Can be used as Primary or Analytic Data Source
* Big Data Capability
* No Single Point of Failure
* Easy Replication
* No Need for Separate Caching Layer
* It provides fast performance and horizontal scalability.
* Can handle structured, semi-structured, and unstructured data with equal effect
* Object-oriented programming which is easy to use and flexible
* NoSQL databases don’t need a dedicated high-performance server
* Support Key Developer Languages and Platforms
* Simple to implement than using RDBMS
* It can serve as the primary data source for online applications.
* Handles big data which manages data velocity, variety, volume, and complexity
* Excels at distributed database and multi-data center operations
* Eliminates the need for a specific caching layer to store data
* Offers a flexible schema design which can easily be altered without downtime or service disruption

**Types of NoSQL Databases**

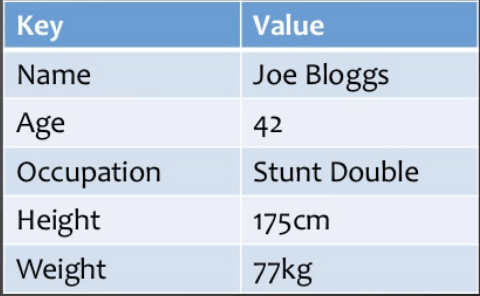
**NoSQL Databases** are mainly categorized into four types: Key-value pair, Column-oriented, Graph-based and Document-oriented. Every category has its unique attributes and limitations. None of the above-specified database is better to solve all the problems. Users should select the database based on their product needs.

Types of NoSQL Databases:

* Key-value Pair Based
* Column-oriented Graph
* Graphs based
* Document-oriented



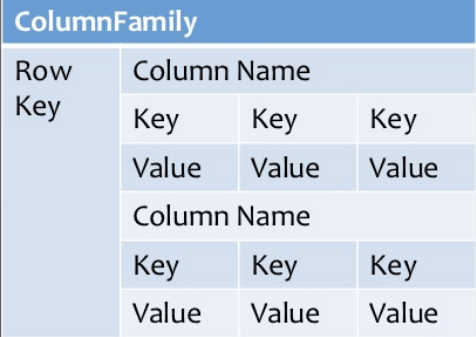
* 1. **Key Value Pair Based**
* Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load.
* Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.
* For example, a key-value pair may contain a key like “Website” associated with a value like “Guru99”.



It is one of the most basic NoSQL database example. This kind of NoSQL database is used as a collection, dictionaries, associative arrays, etc. Key value stores help the developer to store schema-less data. They work best for shopping cart contents.

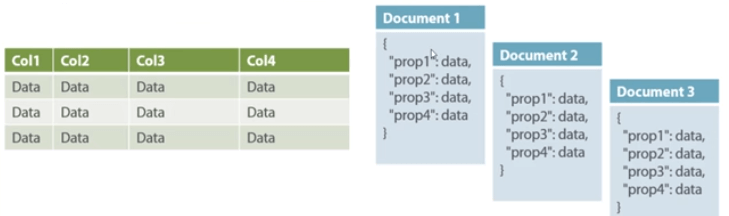
Redis, Dynamo, Riak are some NoSQL examples of key-value store DataBases. They are all based on Amazon’s Dynamo paper.

* 1. **Column-based**
* Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.



Column based NoSQL database

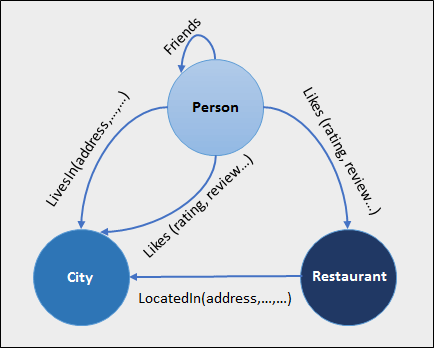
* They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.
* Column-based NoSQL databases are widely used to manage data warehouses, [business intelligence](https://www.guru99.com/business-intelligence-definition-example.html), CRM, Library card catalogs,
* HBase, Cassandra, HBase, Hypertable are NoSQL query examples of column based database.
  1. **Document-Oriented:**
* Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.



**Relational Vs. Document**

In this diagram on your left you can see we have rows and columns, and in the right, we have a document database which has a similar structure to JSON. Now for the relational database, you have to know what columns you have and so on. However, for a document database, you have data store like JSON object. You do not require to define which make it flexible.

* The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not use for complex transactions which require multiple operations or queries against varying aggregate structures.
* Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB, are popular Document originated DBMS systems.
  1. **Graph-Based**
* A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge has a unique identifier.



* Compared to a relational database where tables are loosely connected, a Graph database is a multi-relational in nature. Traversing relationship is fast as they are already captured into the DB, and there is no need to calculate them.
* Graph base database mostly used for social networks, logistics, spatial data.
* Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.

1. **Explain about graph databases in detail?**

Graph stores are NoSQL databases designed for storing data that has graph structure with nodes and edges. While relational databases model data in the form of rows and columns, the graph databases model data in the form of nodes and relationships. Nodes represent the entities in the data model. Nodes have a set of attributes. A node can represent different types of entities, for example, a person, place (such as a city, restaurant or a building) or an object (such as a car).

The relationships between the entities are represented in the form of links between the nodes. Links also have a set of attributes. Links can be directed or undirected. Directed links denote that the relationship is unidirectional. For example, for two entities author and book, a unidirectional relationship called ‘writes’ exists between them, such that an author writes a book Whereas for two friends, say A and B, the friendship relationship between A and B is bidirectional. In the graph theory terminology, the vertices in a graph are the nodes representing the entities and the edges between the vertices are the links between the nodes representing the relationships between the entities. A set of nodes along with the links between them form a path.

* Graph databases are useful for a wide range of applications, where you may need to

model entities and the relationships between them, such as social media, financial, networking

or various types of enterprise applications.

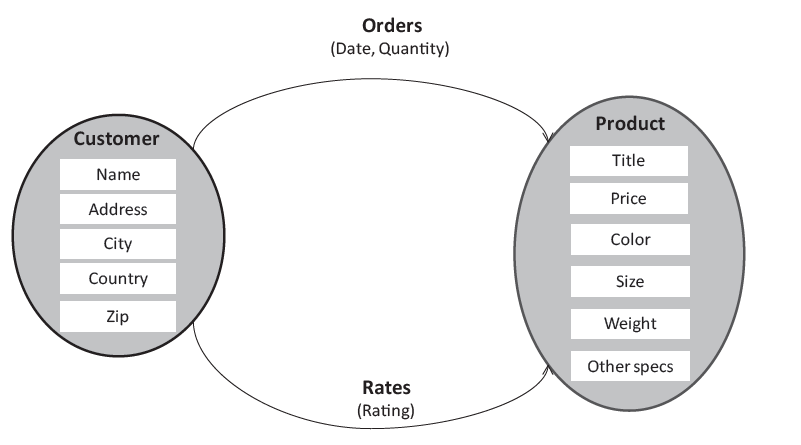
* Graph databases, in contrast to relational databases, model relationships in the form of links between the nodes. Since the relationships between the entities are explicitly stored in the form of links, querying for related entities in graph databases is much simpler and faster than relational databases as the complex join operations are avoided.
* Graph databases are suitable for applications in which the primary focus is on querying for relationships between entities and analyzing the relationships.

**Neo4j**

* Neo4j is one the popular graph databases which provides support for Atomicity, Consistency, Isolation, Durability (ACID). Neo4j adopts a graph model that consists of nodes and relationships.
* Both nodes and relationships have properties which are captured in the form of multiple attributes (key-value pairs). Nodes are tagged with labels which are used to represent different roles in the domain being modelled.

Let us look at an example of using a Graph database for an eCommerce application. Figure below shows a labeled property graph model for an eCommerce application.

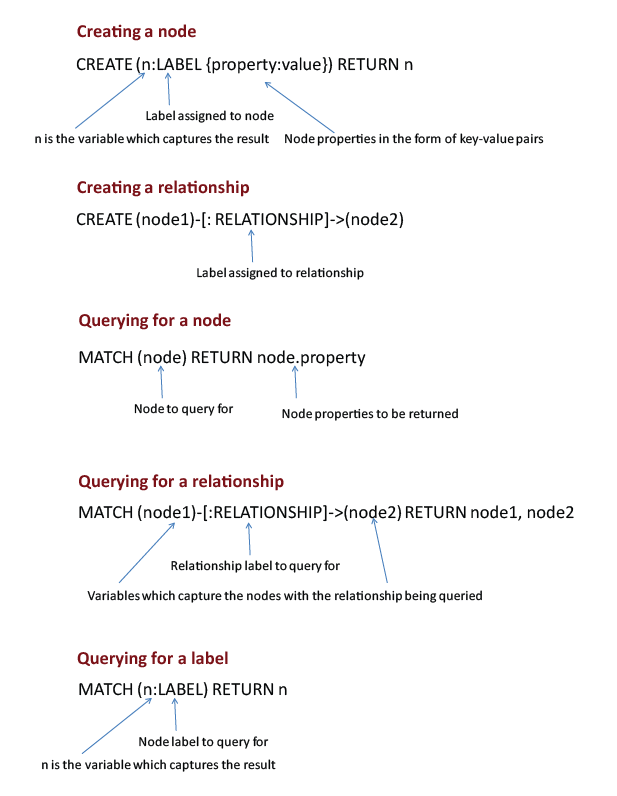
* In this graph, we have two types of nodes: Customer and Product. The Customer nodes have attributes such as customer name, address, city, country and zip code.
* The Product nodes have attributes such as product title, price and various other product-specific properties (such as color, size, weight, etc.). There are two types of relationships between the customer and product nodes: Orders or Rates. The Order relationship between a customer and product has properties such as the order date and quantity.
* The Rates relationship between a customer and product has a single property to capture the customer rating.



* For create, read, update and delete (CRUD) operations, Neo4j provides a query language

called Cypher. Cypher has some similarities with the SQL query language used for

relational databases.



1. **Differences between SQL vs NoSQL explain it with suitable examples?**

| **Parameter** | **SQL** | **NOSQL** |
| --- | --- | --- |
| Definition | SQL databases are primarily called RDBMS or Relational Databases | NoSQL databases are primarily called as Non-relational or distributed database |
| Design for | Traditional RDBMS uses SQL syntax and queries to analyze and get the data for further insights. They are used for OLAP systems. | NoSQL database system consists of various kind of database technologies. These databases were developed in response to the demands presented for the development of the modern application. |
| Query Language | Structured query language (SQL) | No declarative query language |
| Type | SQL databases are table based databases | NoSQL databases can be document based, key-value pairs, graph databases |
| Schema | SQL databases have a predefined schema | NoSQL databases use dynamic schema for unstructured data. |
| Ability to scale | SQL databases are vertically scalable | NoSQL databases are horizontally scalable |
| Examples | Oracle, Postgres, and MS-SQL. | [MongoDB](https://www.guru99.com/mongodb-tutorials.html), Redis, Neo4j, Cassandra, Hbase. |
| Best suited for | An ideal choice for the complex query intensive environment. | It is not good fit complex queries. |
| Hierarchical data storage | SQL databases are not suitable for hierarchical data storage. | More suitable for the hierarchical data store as it supports key-value pair method. |
| Variations | One type with minor variations. | Many different types which include key-value stores, document databases, and graph databases. |
| Development Year | It was developed in the 1970s to deal with issues with flat file storage | Developed in the late 2000s to overcome issues and limitations of SQL databases. |
| Open-source | A mix of open-source like Postgres & MySQL, and commercial like Oracle Database. | Open-source |
| Consistency | It should be configured for strong consistency. | It depends on DBMS as some offers strong consistency like MongoDB, whereas others offer only offers eventual consistency, like [Cassandra](https://www.guru99.com/cassandra-tutorial.html). |
| Best Used for | RDBMS database is the right option for solving ACID problems. | NoSQL is a best used for solving data availability problems |
| Importance | It should be used when data validity is super important | Use when it’s more important to have fast data than correct data |
| Best option | When you need to support dynamic queries | Use when you need to scale based on changing requirements |
| Hardware | Specialized DB hardware (Oracle Exadata, etc.) | Commodity hardware |
| Network | Highly available network (Infiniband, Fabric Path, etc.) | Commodity network (Ethernet, etc.) |
| Storage Type | Highly Available Storage (SAN, RAID, etc.) | Commodity drives storage (standard HDDs, JBOD) |
| Best features | Cross-platform support, Secure and free | Easy to use, High performance, and Flexible tool. |
| Top Companies Using | Hootsuite, CircleCI, Gauges | Airbnb, Uber, Kickstarter |
| Average salary | The average salary for any professional SQL Developer is $84,328 per year in the U.S.A. | The average salary for “NoSQL developer” ranges from approximately $72,174 per year |
| ACID vs. BASE Model | [ACID](https://www.guru99.com/dbms-transaction-management.html)( Atomicity, Consistency, Isolation, and Durability) is a standard for RDBMS | Base (Basically Available, Soft state, Eventually Consistent) is a model of many NoSQL systems |

1. **Discuss in detail about characteristics of NoSQL databases?**

**Characteristics of NoSQL Database**

  The most common features that define a basic NoSQL database are as follows:

1. **Complex-free working**

  Unlike SQL databases, NoSQL databases are not complicated. They store data in an unstructured or a semi-structured form that requires no relational or tabular arrangement. Perhaps they are easier to use and can be accomplished by all.

1. **Independent of Schema** Secondly, NoSQL databases are independent of schemas which implies that they can be run over without any predetermined schemas.

That said, they are far more efficient to work with and perhaps this particular feature works well for young programmers and organizations handling large amounts of heterogeneous data that requires no schemas to structure it.

1. **Better Scalability**

  One of the most prominent features of such a database is that it has high scalability that makes it suitable for large amounts of data. Needless to mention that the contemporary data scientists often prefer to work with NoSQL databases due to this feature since it allows them to accommodate humongous data without rupturing its efficacy.

1. **Flexible to accommodate**

 Since such databases can accommodate heterogeneous data that requires no structuring, they are claimed to be flexible in terms of their usage and reliability. For beginners intending to try their hands in the field, NoSQL databases are easy to handle yet very useful.

1. **Durable**

 If durability is not one of its most striking features, then what is? NoSQL databases are highly durable as they can accommodate data ranging from heterogeneous to homogeneous.

Not only can they accommodate structured data, but they can also incorporate unstructured data that requires no query language. Undoubtedly, these databases are durable and efficient.

## ****Applications of NoSQL Databases****

### ****Data Mining****

 When it comes to data mining, NoSQL databases are useful in retrieving information for data mining uses. Particularly when it’s about large amounts of data, NoSQL databases store data points in both structured and unstructured formats leading to efficient storage of big data.

Perhaps when a user wishes to mine a particular dataset from large amounts of data, one can make use of NoSQL databases, to begin with. Data is the building block of technology that has led mankind to such great heights.

 Therefore, one of the most essential fields where NoSQL databases can be put to use is data mining and data storage.

### ****Social Media Networking Sites****

 Social media is full of data, both structured and unstructured. A field that is loaded with tons of data to be discovered, social media is one of the most effective applications of NoSQL databases.

 From comments to posts, user-related information to advertising, [social media marketing](https://www.analyticssteps.com/blogs/social-media-marketing-working-and-advantages) requires NoSQL databases to be implemented in certain ways to retrieve useful information that can be helpful in certain ways.

 Social media sites like Facebook and Instagram often approach open-source NoSQL databases to extract data that helps them keep track of their users and the activities going on around their platforms.

### ****Software Development****

 The third application that we will be looking at is [software development](https://www.analyticssteps.com/blogs/7-top-trends-software-development). Software development requires extensive research on users and the needs of the masses that are met through software development.

However, a developer must be able to scan through data that is available.

 Perhaps NoSQL databases are always useful in helping software developers keep a tab on their users, their details, and other user-related data that is important to be noted. That said, NoSQL databases are surely helpful in software development.