# 3.1 MongoDB Queries, Filters and Criteria

### **What is MongoDB Query?**

MongoDB Query is a way to get the data from the MongoDB database. MongoDB queries provide the simplicity in process of fetching data from the database, it’s similar to SQL queries in SQL Database language. While performing a query operation, one can also use criteria or conditions which can be used to retrieve specific data from the database.

MongoDB provides the function names as ***db.collection\_name.find()***to operate query operation on database.

Here, we are working with:

**Database:** geeksforgeeks

**Collection:** Article

### **Field selection**

### The find() method displays the database collection in Non-Structured form({<Key> : <value>}) including auto-created <key> ” id  ” by MongoDB and collection data inserted by user or admin.

**Syntax:**

*db.collection\_name.find()*

**Example:**

db.article.find()

This method is used to display all the documents present in the article collection.

### **Finding a single document**

### In MongoDB, we can find a single document using *findOne()* method, This method returns the first document that matches the given filter query expression.

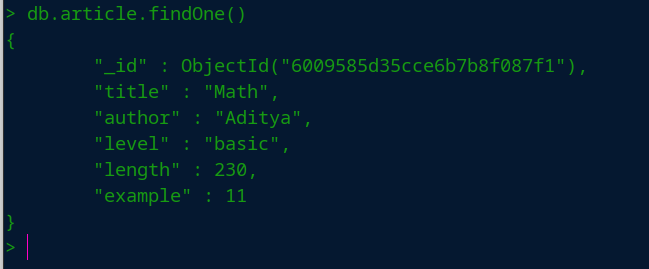
**Syntax:**

db.collection\_name.findOne ()

**Example:**

db.article.findOne()

Here, we are going to display the first document of the article collection.



### **Displaying documents in a formatted way**

In MongoDB, we can display documents of the specified collection in well-formatted way using **pretty()** method.

**Syntax:**

*db.collection\_name.find().pretty()*

**Example:**

*db.article.find().pretty()*

Here, we are going to display the documents of the article collection in a well-formatted way using pretty() method.



### **Equal filter query**

The equality operator([$eq](https://www.geeksforgeeks.org/mongodb-equality-operator-eq/)) is used to match the documents where the value of the field is equal to the specified value. In other words, the **$eq** operator is used to specify the equality condition.

**Syntax:**

*db.collection\_name.find({< key > : {$eq : < value >}})*

**Example:**

db.article.find({author:{$eq:"devil"}}).pretty()

Here, we are going to display the documents that matches the filter query(i.e., *{author  :  {$eq : “devil”}})*from the article collection.



### **Greater than filter query**

### 

To get the specific numeric data using conditions like greater than equal or less than equal use the [$gte](https://www.geeksforgeeks.org/mongodb-greater-than-equals-to-operator-gte/) or [$lte](https://www.geeksforgeeks.org/mongodb-less-than-equals-to-operator-lte/?ref=rp) operator in the find() method.

**Syntax:**

*db.collection\_name.find({< key > : {$gte : < value >}})*

*or*

*db.collection\_name.find({< key > : {$lte : < value >}})*

**Example:**

db.article.find({length:{$gte:510}}).pretty()

Here, we are querying to get documented data which has the length attribute value greater than 510. So, we pass a filter query that is *{length : {$gte : 510}}* in the find() method.



### **Check the existence filter query**

**$exists** operator shows all the collection documents if they exist on a given key.

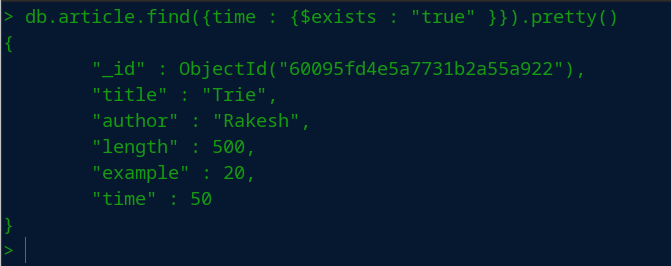
**Syntax:**

*db.collection\_name.find({< key > : {$exists : < boolean >}})*

**Example**:

db.article.find({time:{$exists:"true"}}).pretty()

Here, we are going to look all the documents which has the attribute named as time by passing a filter query that is *{time : {$exists : “true”}}*in the find() method.



### **Logical operator query**

[$and](https://www.geeksforgeeks.org/mongodb-and-operator-and/) operator comes under the type of MongoDB logical operator which perform logical AND operation on the array of one or more expressions and select or retrieve only those documents that match all the given expression in the array.

**Syntax:**

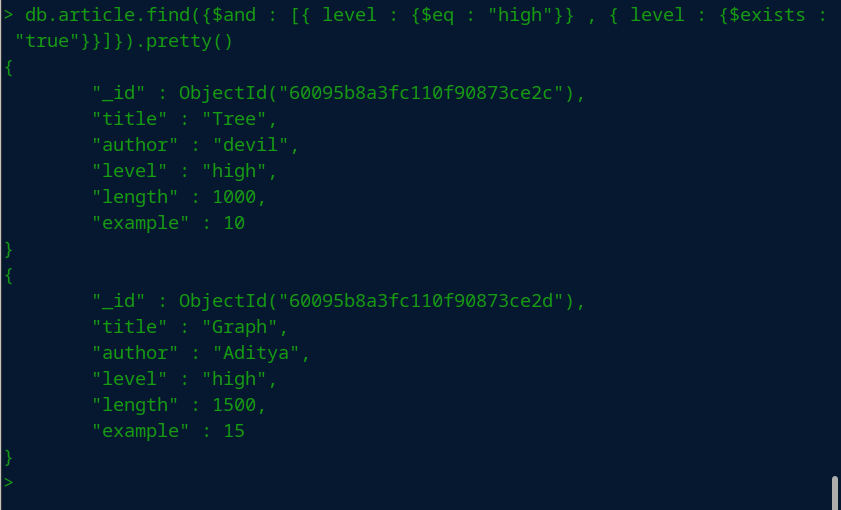
*db.collection\_name.find({$and : [{< key > : {$eq : < value1 >}}, {< key > : {$exists : < boolean >}}]})*

**Example:**

db.article.find({$and:[{level:{$eq:"high"}},{level:{$exists : "true"}}]}).pretty()

In this query example we are using and operator and given two condition which are highlighted following

* and operator: *{$and : [ first condition, second condition]}*
* first condition(level == “high”):*{ level : {$eq  :  “high”}}*
* second condition: *{level : {$exists : “true”}}*



### **Limit Query**

This query method specifies a maximum number of documents for a cursor to return.

**Syntax :**

*db.collection\_name.find({< key > : < value >}).limit(< Integer\_value >)*

**Example:**

db. article. find({author : “devil” }). limit(2) . pretty()

**Syntax :**

*db.collection\_name.find({< key > : < value >}).limit(< Integer\_value >)*

**Example:**

db. article. find({author : “devil” }). limit(2) . pretty()



# 3.2 Managing Multimedia Data with Database.

# 3.2.1. Multimedia Database

**Multimedia database** is the collection of interrelated multimedia data that includes text, graphics (sketches, drawings), images, animations, video, audio etc and have vast amounts of multisource multimedia data. The framework that manages different types of multimedia data which can be stored, delivered and utilized in different ways is known as multimedia database management system. There are three classes of the multimedia database which includes static media, dynamic media and dimensional media.

Content of Multimedia Database management system:

1. **Media data –** The actual data representing an object.
2. **Media format data –** Information such as sampling rate, resolution, encoding scheme etc. about the format of the media data after it goes through the acquisition, processing and encoding phase.
3. **Media keyword data –** Keywords description relating to the generation of data. It is also known as content descriptive data. Example: date, time and place of recording.
4. **Media feature data –** Content dependent data such as the distribution of colors, kinds of texture and different shapes present in data.

Types of multimedia applications based on data management characteristic are as below.

1. **Repository applications –** A Large amount of multimedia data as well as meta-data(Media format date, Media keyword data, Media feature data) that is stored for retrieval purpose, e.g., Repository of satellite images, engineering drawings, radiology scanned pictures.
2. **Presentation applications –** They involve delivery of multimedia data subject to temporal constraint. Optimal viewing or listening requires DBMS to deliver data at certain rate offering the quality of service above a certain threshold. Here data is processed as it is delivered. Example: Annotating of video and audio data, real-time editing analysis.
3. **Collaborative work using multimedia information –** It involves executing a complex task by merging drawings, changing notifications. Example: Intelligent healthcare network.

**Areas where multimedia database is applied are:**

* **Documents and record management:** Industries and businesses that keep detailed records and variety of documents. Example: Insurance claim record.
* **Knowledge dissemination:** Multimedia database is a very effective tool for knowledge dissemination in terms of providing several resources. Example: Electronic books.
* **Education and training:** Computer-aided learning materials can be designed using multimedia sources which are nowadays very popular sources of learning. Example: Digital libraries.
* Marketing, advertising, retailing, entertainment and travel. Example: a virtual tour of cities.
* **Real-time control and monitoring:** Coupled with active database technology, multimedia presentation of information can be very effective means for monitoring and controlling complex tasks Example: Manufacturing operation control.

## Example Media and Entertainment Solutions

**Content Management and Delivery.** Content lies at the heart of many digital products in media and entertainment. Text, images, audio, video, metadata, entitlements and other types of media produced by content workflows can be stored in MongoDB for rich, interactive user applications. Users can reliably engage with content across a wide variety of endpoints, including smartphones and tablets. MongoDB empowers publishers to engage their customers more quickly as new delivery models emerge and as new content and services move online.

**User Data Management.** As users interact with different properties, games and content assets, companies need to track and maintain a rich set of attributes in real-time. Regardless of the device – smartphone, tablet portable gaming or other device – users should experience a seamless, interactive experience that also incorporates their personal contexts, histories and relationships. Some of the world’s largest gaming and media companies rely on MongoDB for user profile management because of its flexible data model and powerful query capabilities, which enable a better user experience through rich access to user data and real-time analytics.

**Digital Asset Management.** The key assets for many media organizations are the products of their content workflows, like books, periodicals, journal articles, images and educational products. Various workflows are responsible for the production of these assets, and many different areas of the business – including online applications, syndication processes, licensing and billing – may need access to the associated metadata. MongoDB provides a flexible, searchable repository for managing all digital assets across the organization, facilitating cooperation and reducing time to market.

# 3.2.2 Managing Multimedia data with GridFS in MongoDB

[GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS) is a specification for storing and retrieving files that exceed the [BSON](https://docs.mongodb.com/manual/reference/glossary/#std-term-BSON)-document [size limit](https://docs.mongodb.com/manual/reference/limits/#std-label-limit-bson-document-size) of 16 MB.

Instead of storing a file in a single document, GridFS divides the file into parts, or chunks, and stores each chunk as a separate document. By default, GridFS uses a default chunk size of 255 kB; that is, GridFS divides a file into chunks of 255 kB with the exception of the last chunk. The last chunk is only as large as necessary. Similarly, files that are no larger than the chunk size only have a final chunk, using only as much space as needed plus some additional metadata.

GridFS uses two collections to store files. One collection stores the file chunks, and the other stores file metadata. The section [GridFS Collections](https://docs.mongodb.com/manual/core/gridfs/#std-label-gridfs-collections) describes each collection in detail.

When you query GridFS for a file, the driver will reassemble the chunks as needed. You can perform range queries on files stored through GridFS. You can also access information from arbitrary sections of files, such as to "skip" to the middle of a video or audio file.

GridFS is useful not only for storing files that exceed 16 MB but also for storing any files for which you want access without having to load the entire file into memory. See also [When to Use GridFS](https://docs.mongodb.com/manual/core/gridfs/#std-label-faq-developers-when-to-use-gridfs).

## When to Use GridFS

In MongoDB, use [GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS) for storing files larger than 16 MB.

In some situations, storing large files may be more efficient in a MongoDB database than on a system-level filesystem.

* If your filesystem limits the number of files in a directory, you can use GridFS to store as many files as needed.
* When you want to access information from portions of large files without having to load whole files into memory, you can use GridFS to recall sections of files without reading the entire file into memory.
* When you want to keep your files and metadata automatically synced and deployed across a number of systems and facilities, you can use GridFS. When using [geographically distributed replica sets](https://docs.mongodb.com/manual/core/replica-set-architecture-geographically-distributed/#std-label-replica-set-geographical-distribution), MongoDB can distribute files and their metadata automatically to a number of [mongod](https://docs.mongodb.com/manual/reference/program/mongod/" \l "mongodb-binary-bin.mongod) instances and facilities.

Do not use GridFS if you need to update the content of the entire file atomically. As an alternative you can store multiple versions of each file and specify the current version of the file in the metadata. You can update the metadata field that indicates "latest" status in an atomic update after uploading the new version of the file, and later remove previous versions if needed.

Furthermore, if your files are all smaller than the 16 MB [BSON Document Size](https://docs.mongodb.com/manual/reference/limits/#mongodb-limit-BSON-Document-Size) limit, consider storing each file in a single document instead of using GridFS. You may use the BinData data type to store the binary data. See your [drivers](https://docs.mongodb.com/drivers/) documentation for details on using BinData.

## Use GridFS

To store and retrieve files using [GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS), use either of the following:

* A MongoDB driver. See the [drivers](https://docs.mongodb.com/drivers/) documentation for information on using GridFS with your driver.
* The [mongofiles](https://docs.mongodb.com/database-tools/mongofiles/" \l "mongodb-binary-bin.mongofiles) command-line tool. See the [mongofiles](https://docs.mongodb.com/database-tools/mongofiles/" \l "mongodb-binary-bin.mongofiles) reference for documentation.

## GridFS Collections

[GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS) stores files in two collections:

* chunks stores the binary chunks.
* files stores the file's metadata.

GridFS places the collections in a common bucket by prefixing each with the bucket name. By default, GridFS uses two collections with a bucket named fs:

* fs.files
* fs.chunks

You can choose a different bucket name, as well as create multiple buckets in a single database. The full collection name, which includes the bucket name, is subject to the [namespace length limit](https://docs.mongodb.com/manual/reference/limits/#mongodb-limit-Namespace-Length).

### **The chunks Collection**

Each document in the chunks collection represents a distinct chunk of a file as represented in [GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS). Documents in this collection have the following form:

{

"\_id" : <ObjectId>,

"files\_id" : <ObjectId>,

"n" : <num>,

"data" : <binary>

}

A document from the chunks collection contains the following fields:

**chunks.id**

The unique [ObjectId](https://docs.mongodb.com/manual/reference/glossary/" \l "std-term-ObjectId) of the chunk.

**chunks.files\_id**

The \_id of the "parent" document, as specified in the files collection.

**chunks.n**

The sequence number of the chunk. GridFS numbers all chunks, starting with 0.

**chunks.data**

The chunk's payload as a [BSON](https://docs.mongodb.com/manual/reference/glossary/#std-term-BSON) Binary type.

### **The Files Collection**

Each document in the files collection represents a file in [GridFS](https://docs.mongodb.com/manual/reference/glossary/" \l "std-term-GridFS).

{

"\_id" : <ObjectId>,

"length" : <num>,

"chunkSize" : <num>,

"uploadDate" : <timestamp>,

"md5" : <hash>,

"filename" : <string>,

"contentType" : <string>,

"aliases" : <string array>,

"metadata" : <any>,

}

Documents in the **files** collection contain some or all of the following fields:

**files.\_id**

The unique identifier for this document. The \_id is of the data type you chose for the original document. The default type for MongoDB documents is [BSON](https://docs.mongodb.com/manual/reference/glossary/#std-term-BSON) [ObjectId](https://docs.mongodb.com/manual/reference/glossary/" \l "std-term-ObjectId).

**files.length**

The size of the document in bytes.

**files.chunkSize**

The size of each chunk in **bytes**. GridFS divides the document into chunks of size chunkSize, except for the last, which is only as large as needed. The default size is 255 kilobytes (kB).

**files.uploadDate**

The date the document was first stored by GridFS. This value has the Date type.

**files.md5**

**Deprecated**

The MD5 algorithm is prohibited by FIPS 140-2. MongoDB drivers deprecate MD5 support and will remove MD5 generation in future releases. Applications that require a file digest should implement it outside of GridFS and store in [files.metadata](https://docs.mongodb.com/manual/core/gridfs/" \l "mongodb-data-files.metadata).

An MD5 hash of the complete file returned by the [filemd5](https://docs.mongodb.com/manual/reference/command/filemd5/) command. This value has the String type.

**files.filename**

Optional. A human-readable name for the GridFS file.

**files.contentType**

**Deprecated**

Optional. A valid MIME type for the GridFS file. For application use only.

Use [files.metadata](https://docs.mongodb.com/manual/core/gridfs/" \l "mongodb-data-files.metadata) for storing information related to the MIME type of the GridFS file.

**files.aliases**

**Deprecated**

Optional. An array of alias strings. For application use only.

Use [files.metadata](https://docs.mongodb.com/manual/core/gridfs/" \l "mongodb-data-files.metadata) for storing information related to the MIME type of the GridFS file.

**files.metadata**

Optional. The metadata field may be of any data type and can hold any additional information you want to store. If you wish to add additional arbitrary fields to documents in the files collection, add them to an object in the metadata field.

## GridFS Indexes

GridFS uses indexes on each of the chunks and files collections for efficiency. [Drivers](https://docs.mongodb.com/drivers/) that conform to the [GridFS specification](https://github.com/mongodb/specifications/blob/master/source/gridfs/gridfs-spec.rst) automatically create these indexes for convenience. You can also create any additional indexes as desired to suit your application's needs.

### **The chunks Index**

[GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS) uses a [unique](https://docs.mongodb.com/manual/reference/glossary/#std-term-unique-index), [compound](https://docs.mongodb.com/manual/reference/glossary/#std-term-compound-index) index on the chunks collection using the files\_id and n fields. This allows for efficient retrieval of chunks, as demonstrated in the following example:

db.fs.chunks.find( { files\_id: myFileID } ).sort( { n: 1 } )

### **The files Index**

[GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS) uses an [index](https://docs.mongodb.com/manual/reference/glossary/#std-term-index) on the files collection using the filename and uploadDate fields. This index allows for efficient retrieval of files, as shown in this example:

db.fs.files.find( { filename: myFileName } ).sort( { uploadDate: 1 } )

## Sharding GridFS

There are two collections to consider with [GridFS](https://docs.mongodb.com/manual/reference/glossary/#std-term-GridFS) - files and chunks.

### **chunks Collection**

To shard the chunks collection, use either { files\_id : 1, n : 1 } or { files\_id : 1 } as the shard key index. Files\_id is an [ObjectId](https://docs.mongodb.com/manual/reference/glossary/" \l "std-term-ObjectId) and changes [monotonically](https://docs.mongodb.com/manual/core/sharding-choose-a-shard-key/#std-label-shard-key-monotonic).

For MongoDB drivers that do not run filemd5 to verify successful upload (for example, MongoDB drivers that support MongoDB 4.0 or greater), you can use [Hashed Sharding](https://docs.mongodb.com/manual/core/hashed-sharding/) for the chunks collection.

If the MongoDB driver runs [filemd5](https://docs.mongodb.com/manual/reference/command/filemd5/#mongodb-dbcommand-dbcmd.filemd5), you cannot use [Hashed Sharding](https://docs.mongodb.com/manual/core/hashed-sharding/).

**Files Collection**

The files collection is small and only contains metadata. None of the required keys for GridFS lend themselves to an even distribution in a sharded environment. Leaving files unsharded allows all the file metadata documents to live on the [primary shard](https://docs.mongodb.com/manual/reference/glossary/#std-term-primary-shard).

If you must shard the files collection, use the \_id field, possibly in combination with an application field.

3.3 Fundamentals of Big Data

A Collection of large and complex datasets which are difficult to store and process using the traditional database and data processing tools is considered as big data. Big data is collected from traditional and digital sources which, when refined properly can be used for research and analysis. With time, organizations are growing and with this data generated from these organizations are also increasing exponentially. The challenge is to have a platform which can provide a single, consistent view of the complete data. Another challenge is to organize this data so that it makes sense and can be utilized as useful information. Everything around us generates [big data](https://intellipaat.com/big-data-hadoop-training/) continuously. Social media websites and digital sources are responsible for producing such huge amount of data. How this huge amount of data is transmitted – sensors, mobile and systems are the answer.

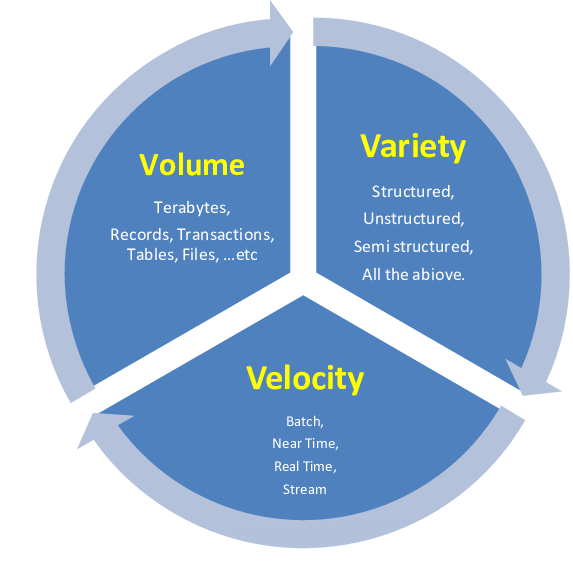
**3.3.1 Where is this Big Data coming from?**

* **Social media**: Big data companies like Facebook and google get the data from whatever activities we perform. Other examples are YouTube, Twitter, LinkedIn, blogs, SlideShare, Instagram, chatter, WordPress, Jive, etc.
* **Public Web:** This includes data coming from Wikipedia, health care services, the World Bank, government, weather, traffic, etc.
* **Archives:** This includes archives of any data like medical records, customer correspondence, insurance forms, scanned documents, etc.
* **Docs**: Documentation of any format including HTML, CSV, PDF, XLS, Word, XML, etc. are the sources of big data.
* **Media**: Images, video, audio, live stream, podcast etc.
* **Data storage**: The various database and file systems which are used to store the data serve as the source for big data.
* **Machine Log Data**: Data coming from server, application logs, audit logs, CDR- call detail records, various mobile apps, mobile location etc.
* **Sensor Data**: Data from sensors connected to medical devices, road cameras, satellites, traffic surveillance devices, video games, household appliances, air conditioning units, office buildings etc.

**3.3.2 Three Vs of Big data**

There are 3Vs that define Big data velocity, variety and volume

1. **Variety:** There are multiple formats to store data, e.g., database, MS-Access, MS-Excel, text and many more. It can also be in the form of pdf, video or SMS. So the challenge is to arrange this data to make it meaningful and it is easier when the data is in the same format.
2. **Volume:** The volume of data coming from multiple sources is huge. With this increase in the volume of data it is important for the organizations to reevaluate their architecture and applications.
3. **Velocity:**Velocity refers to the processing speed of the data. In earlier days, yesterday’s data were considered as recent data, but now this thing is valid only in the newspapers. Rest, everything gets updated even less than a fraction of seconds. News channel, radios, tweets, Facebook posts and comments everything updates so fast that data updates few minutes back is considered useless and old.

3v’s of Big Data

Reference: <https://www.researchgate.net/figure/Big-Data-Definition-3-Vs_fig3_264129835>

# 5 V’s of Big Data

Big Data was defined by the “3Vs” but now there is “5Vs” of Big Data which are also termed as the characteristics of Big Data as follows:

**Volume:**

* The name ‘Big Data’ itself is related to a size which is enormous.
* Volume is a huge amount of data.
* To determine the value of data, size of data plays a very crucial role. If the volume of data is very large then it is actually considered as a ‘Big Data’. This means whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data.
* Hence while dealing with Big Data it is necessary to consider a characteristic ‘Volume’.
* *Example:* In the year 2016, the estimated global mobile traffic was 6.2 Exabytes(6.2 billion GB) per month. Also, by the year 2020 we will have almost 40000 ExaBytes of data.

**Velocity:**

* Velocity refers to the high speed of accumulation of data.
* In Big Data velocity data flows in from sources like machines, networks, social media, mobile phones etc.
* There is a massive and continuous flow of data. This determines the potential of data that how fast the data is generated and processed to meet the demands.
* Sampling data can help in dealing with the issue like ‘velocity’.
* *Example:* There are more than 3.5 billion searches per day are made on Google. Also, FaceBook users are increasing by 22%(Approx.) year by year.

**Variety:**

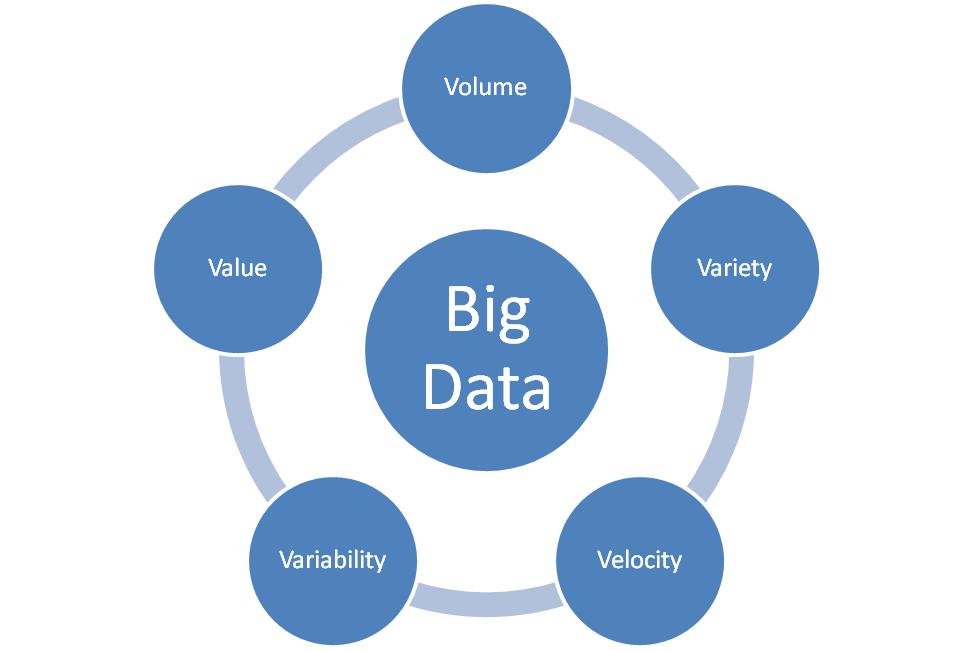
* It refers to nature of data that is structured, semi-structured and unstructured data.
* It also refers to heterogeneous sources.
* Variety is basically the arrival of data from new sources that are both inside and outside of an enterprise. It can be structured, semi-structured and unstructured.
  + **Structured data**: This data is basically an organized data. It generally refers to data that has defined the length and format of data.
  + **Semi- Structured data**: This data is basically a semi-organised data. It is generally a form of data that do not conform to the formal structure of data. Log files are the examples of this type of data.
  + **Unstructured data**: This data basically refers to unorganized data. It generally refers to data that doesn’t fit neatly into the traditional row and column structure of the relational database. Texts, pictures, videos etc. are the examples of unstructured data which can’t be stored in the form of rows and columns.

**Veracity:**

* It refers to inconsistencies and uncertainty in data, that is data which is available can sometimes get messy and quality and accuracy are difficult to control.
* Big Data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources.
* *Example:* Data in bulk could create confusion whereas less amount of data could convey half or Incomplete Information.

**Value:**

* After having the 4 V’s into account there comes one more V which stands for Value!. The bulk of Data having no Value is of no good to the company, unless you turn it into something useful.
* Data in itself is of no use or importance but it needs to be converted into something valuable to extract Information. Hence, you can state that Value! is the most important V of all the 5V’s.



5 V’s of Big Data

# What is Big Data?

Big Data is nothing but lots of data consisting of varieties of data. It is the concept of gathering useful insights from such voluminous amounts of structured, semi-structured and unstructured data that can be used for effective decision making in the business environment. This data is collected from various sources over a course of time and is cumbersome to be managed by traditional database tools.

Data science is the study of data analyzing by advance technology ([Machine Learning](https://www.geeksforgeeks.org/machine-learning/), Artificial Intelligence, Big data). It processes a huge amount of structured, semi-structured, unstructured data to extract insight meaning, from which one pattern can be designed that will be useful to take a decision for grabbing the new business opportunity, the betterment of product/service,

ultimately business growth.  
Data science process to make sense of Big data/huge amount of data that is used in business. The workflow of Data science is as below:

* Objective and the issue of business determining – What is organization objective, what level organization want to achieve at, what issue company is facing -these are the factors under consideration. Based on such factors which type of data are relevant is considered.
* Collection of relevant data- relevant data are collected from various source.
* Cleaning and filtering collected data – non-relevant data are removed.
* Explore the filtered, cleaned data – Finding any hidden pattern, synchronization in data, plotting them in the graph, chart, etc. form that is understandable to non-technical person.
* Creating a model by analyzing data – creating a model, validate it.
* Visualization of finding by interpreting data or created a model to a business person.
* Help businessperson in making the decision and taking the step for the sack of business growth.

**Data Mining:**It is a process of extracting insight meaning, hidden pattern from collected data that is useful to take a business decision in the purpose of decreasing expenditure and increasing revenue.

**Big Data:**This is a term related to extracting meaningful data by analyzing the huge amount of complex, variously formatted data generated at high speed, that cannot be handled, processed by the traditional system.

**Data Expansion Day by Day:**Day by day amount of data increasing exponentially because of today’s various data production sources like a smart electronic device. As per IDC (International Data Corporation) report, new data created per each person in the world per second by 2020 will be 1.7 MB. The amount of total data in the world by 2020 will reach around 44 ZettaBytes (44 trillion GigaByte) and 175 ZettaBytes by 2025. It is being seen that total volume of data being double every two years. Total size growth of data worldwide, year to year as per IDC report is shown below:

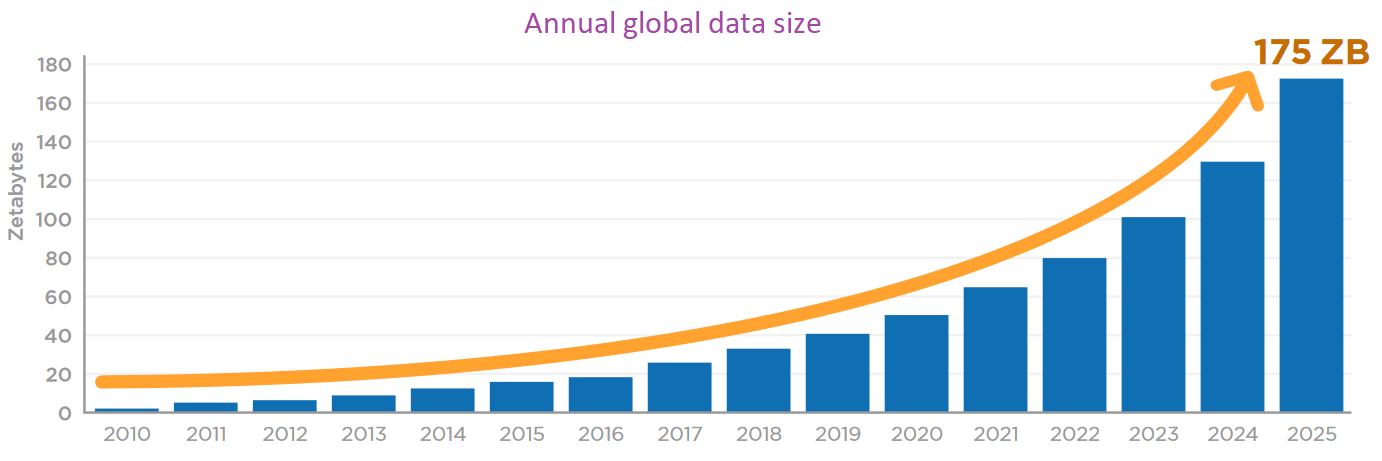
[](https://media.geeksforgeeks.org/wp-content/uploads/20190817192013/Data-expansion-over-years1.png)

Image Source: <https://www.geeksforgeeks.org/what-is-big-data/?ref=lbp>

**3.4 Managing Big Data**

**Steps for Big data management.**

## ****1. Determine your goals****.

For every study or event, you have to outline certain goals that you want to achieve. You have to ask yourself questions. You want to discuss with your team what they see as most important. The goals will determine what data you should collect and how to move forward.

Without setting clear goals and mapping out strategies towards achieving them, you’re either going to collect the wrong data, or too little of the right data. And even if you were to collect the right amount of the right data, you’d not know what exactly to do with it. It just makes zero sense to expect to get to a destination you didn’t know.

## ****2. Secure your data****.

You have to make sure that whatever container holds your data is accessible and secure. You don’t want to lose your data. You can’t analyze what you don’t have. Ensure you implement proper firewall security, spam filtering, malware scanning and permission control for team members.

Recently, I attended a webinar by Robert Carter, CEO of Your Company Formations and he shared his experience with the entrepreneurs they work with. He said many business owners collect data from users’ interactions with their sites and products but don’t take any or enough precautions and measures to secure the data. This has cost some businesses their clients’ trust, crashed the businesses of some others, and even sent some bankrupt with heavy fines in damages.

“Securing your data appears like an obvious point but too many businesses and organizations observe the advise in the breach,” he concluded. So don’t be one of them.

## ****3. Protect the data****

Aside human intruders and artificial threats to your data, some natural elements could also corrupt your data or make you lose them totally.

Often, people forget that heat, humidity and extreme cold can harm data. These problems can lead to system failure which causes downtime and frustration. You want to watch for these environmental situations, and take actions to stop your data loss before it happens. Don’t be sorry when you can avoid it.

## ****4. Follow audit regulations****

Even though many data managers are on the go, they still must maintain the right components in case of an audit. Whether you’re managing customer’s payment data, credit score (or cibil score) data or even seemingly mundane data like anonymous details of site users, you have to manage your assets correctly.

This ensures you stay safe from liability and continue to earn cutomers’ and users’ trust.

## ****5. Data need to talk to each other****

Make sure you use software that integrates many solutions. The last thing you need is for you to have problems caused by applications not being able to communicate with your data or vice versa.

You should make good use of cloud storage, remote database administrator and other data management tools to ensure seamless synchronization of your data sets, especially where more than one of your team members do access or work on them simultaneously.

## ****6. Know what data to capture****

When you are the manager of big data, you have to understand what data are the best for a particular situation. Therefore, you have to know which data to collect and when to do it.

This goes back to the basis: Knowing your objectives clearly and how to achieve them with the right data.

## ****7. Adapt to changes****

Software and data are changing almost daily. New tools and products hit the market daily making the previous gamechanging ones seem outdated. For instance, if you’re a niche site offering excellent television entertainment options, you’ll find the products you review and recommend change with time. Again, if you sell toothbrush and you already know a lot about your customers’ taste after having collected data about their demographics and interests over a period of six months, you’ll need to change your sales strategy if the need and taste of your customers start showing a strong preference for electric tootbrush over the manual one. You’ll also need to change how you collect data about their interests. This fact applies to all industries and refusing to adapt in that situation is a recipe for failure.

You have to be flexible to adapt to new ways of managing your data and to changes in your data. That’s how to stay relevant in your industry and truly reap the benefits of big data.

Keeping these tips in mind will help you handle big data in an easy manner.

**3.5 Big Data Tools and Technologies**

Different tools have been developed by different developers in order to manage huge data. Different tool have its own method of implementation and management. This functionality of tools depends on the aim for which that particular tool was developed.

Many architectures for data management has been developed till the date. The basic functionality of these tools can be divided into two types of management:

* Master-job management- which examines the processing nodes assigns tasks to the nodes and manages their activity.
* Storage management- it examines the data storage and divides the datasets to different database.

For this there are different tools developed which are discussed below

**HDFS**

Hadoop Distributed File System (HDFS) distributes the data over the data nodes. There are four types of nodes involved within HDFS. They are:

* Name Node: a facilitator that provides information on the location of data. It knows which nodes are available, where in the cluster certain data resides, and which nodes have failed.
* Secondary Node: a backup to the Name Node
* Job Tracker: coordinates the processing of the data using Map Reduce.
* Slave Nodes: store data and take direction from the Job Tracker.

A Job Tracker is the entry point for a ―map job‖ or process to be applied to the data. A map job is typically a query written in java and is the first step in the Map Reduce process. The Job Tracker asks the name node to identify and locate the necessary data to complete the job.

Once it has this information it submits the query to the relevant named nodes. Any required processing of the data occurs within each named node, which provides the massively parallel characteristic of Map Reduce.

When the each node has finished processing, it stores the results. The client then initiates a "Reduce" job. The results are then aggregated to determine the ―answer‖ to the original query. The client then accesses these results on the file system and can use them for whatever purpose.

**MapReduce**

MapReduce is the combination of job management and the models used in programming for execution. It basically works on the principle of master-slave model in which one node acts as the master node called as the Job Tracker and the other node as slave node called as the Task Trackers.

* Job Tracker- it manages the resources acting as the slave nodes. It continuously monitors the availability as well as the accessibility of resources acting as the slave nodes so that the execution can be completed in time and efficiently. It also has the fault tolerance capability.
* Task Tracker- these are the nodes actually performing the task. The working of these nodes is simpler but these are the actual worker who performs the task. When the task or job is assigned to it, it accepts the task according to the availability and then performs the execution.

There are mainly three limitations in this model.

* Locality- the applications closer to the processing and data are easily provided with the outputs but those who are farther always suffer from the latency issues of the network.
* Mapping- applications are not easily mapped on these models. So this creates the problem of execution of the applications.
* Allocation- allocation of the processing nodes is fixed in some cases. Thus when we compare the allocation of tasks to the nodes several nodes come out to be unused. Thus the proper allocation of processing nodes is not done in this model.

These issues of MapReduce are being tackled in new model called YARN. YARN has the centralized resource management and each node have its local Node Manager which manages the task assigned to the node and the execution of the task. This allows the flexibility of assigning jobs or tasks to the nodes. Thus it reduces the latency issues of the applications and the performance of the model is improved.

**Zookeeper**

When there are multiple jobs being performed in single distributed system then the developer needs to manage the working of the system and have to maintain the synchronization between the various nodes of the same system. Zookeeper is a centralized service for maintaining configuration information, providing group services, naming and providing distributed synchronization.

Thus Zookeeper is used to manage the process of naming the objects and then further manages theses objects in the hierarchical manner. Due to synchronisation capability of Zookeeper it is also used to control the shared resources such that no deadlock state is met.

**HBase**

HBase is the non-relational data management. It does not support SQL queries. Id divides the huge datasets such that these datasets could be used to acquire the required data. The basic development of HBase was from Google ‘s Big Table. So HBase is column oriented. HBase supports memory execution because the column-oriented data can be compressed such that more amounts of data can be represented.

The main advantage HBase is that it acts as an alternative for data storage in MapReduce applications. Because of the columnar orientation and the data organizing capacity of HBase.

**Hive**

Although MapReduce have methodologies for executing different applications and develop them but companies or organizations relies on the tabular representation of data. Thus, this problem of MapReduce gave basis to the development of Hive which comes under the category of Data Warehousing.

Hive enables the data organization in data warehousing. Hive works on the query language called HiveQL which is same the SQL. This system provides the way of loading, extracting and transforming data. It also acts as the native access to MapReduce programmers because the users or developers or programmers can integrate the Map and Reduce Functions to HiveQL queries such that it can programs can be used in Hive also. Thus it reduces the effort of programmers to write program again and again.

**Pig**

It is based on the process of abstracting some details by using a higher level programming language called Pig Latin. It allows the user to specify the process of analysis is being performed. It works in similar way as the SQL works on structured dataset. The similarity exists when removing the duplicate data, splitting datasets, splitting datasets, joins etc. It also allows user to create new user defined functions which can be further used in

**3.6 Big Data in Cloud**

# Big Data and Cloud Computing – A Perfect Combination

two mainstream technologies are the center of concern in IT – Big Data and Cloud Computing. Fundamentally different, Big data is all about dealing with the massive scale of data whereas Cloud computing is about infrastructure. However, the simplification offered by Big data and Cloud technology is the main reason for their huge enterprise adoption. For example Amazon “Elastic Map Reduce” demonstrates how the power of Cloud Elastic Computes is leveraged for Big Data processing.

The combination of both yields beneficial outcome for the organizations. Not to mention, both the technologies are in the stage of evolution but their combination leverages scalable and cost-effective solution in big data analytics.

So, can we say Big data and Cloud computing a perfect combination? Well, there are data points in support of it. Besides that, there are also some real-time challenges to deal with. In this blog, we will discuss both the aspects. We assume you have some idea and knowledge on Big data and Cloud computing.

## Big Data and Cloud Computing Relationship

Big data and Cloud computing both the technologies are valuable on its own. Furthermore, many businesses are targeting to combine the two techniques to reap more business benefits. Both the technologies aim to enhance the revenue of the company while reducing the investment cost. While Cloud manages the local software, Big data helps in business decisions.

Let’s start with the basic outline of the two technologies!

### **Big Data and Cloud Computing**

Big data deals with massive structured, semi-structured or unstructured data to store and process it for data analysis purpose. There are five aspects of Big Data which are described through 5Vs

* Volume – the amount of data
* Variety – different types of data
* Velocity – data flow rate in the system
* Value– the value of data based on the information contained within
* Veracity– data confidentiality and availability

Cloud computing offers services to the users on a pay-as-you-go model. Cloud providers offer three primary services, these services are outlined below:

##### **Infrastructure as a Service (IAAS)**

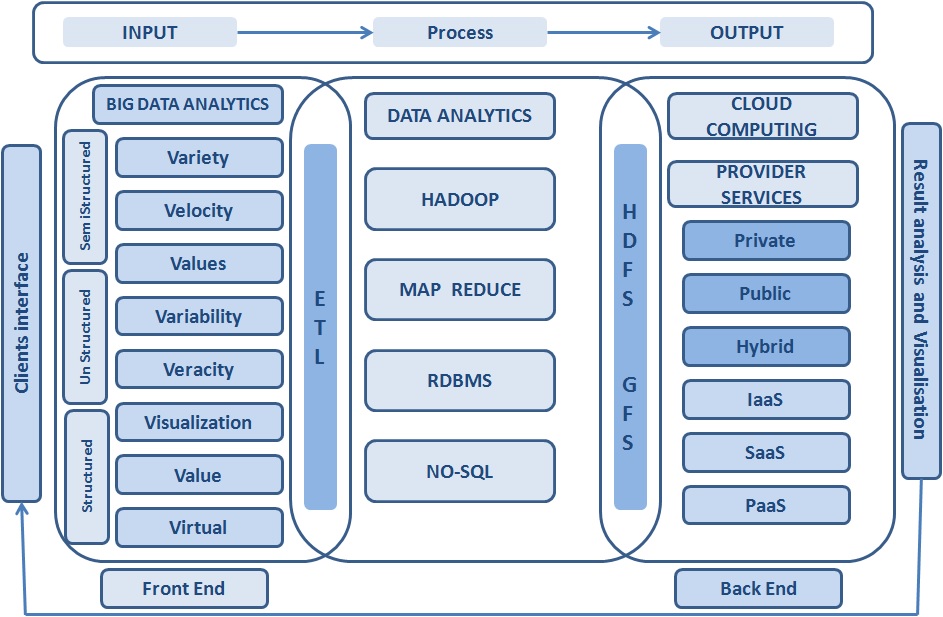
Here the service provider offers entire infrastructure along with the maintenance related tasks.

##### **Platform as a Service (PAAS)**

in this service, the Cloud provider offers resources like object storage, runtime, queuing, databases, etc. However, the responsibility of configuration and implementation related tasks depend on the consumer.

##### **Software as a Service (SAAS)**

This service is the most facilitated one which provides all the necessary settings and infrastructure provides IaaS for the platform and infrastructure are in place.



**Image :** *Big Data and Cloud Computing Relationship Model*

*Reference:* [*https://www.whizlabs.com/blog/big-data-and-cloud-computing/*](https://www.whizlabs.com/blog/big-data-and-cloud-computing/)

### **Cloud Computing Role for Big Data**

Big data and Cloud computing relationship can be categorized based on service types:

##### **IAAS in Public Cloud**

IaaS is a cost-effective solution and utilizing this Cloud service, Big Data services enable people to access unlimited storage and compute power. It is a very cost-effective solution for enterprises where the Cloud provider bears all the expenses of managing underlying hardware.

##### **PAAS in Private Cloud**

PaaS vendors incorporate Big Data technologies into their offered service. Hence, they eliminate the need for dealing with the complexities of managing single software and hardware elements which is a real concern while dealing with terabytes of data.

##### **SAAS in Hybrid Cloud**

Analyzing social media data is nowadays an essential parameter for companies for business analysis. In this context, SaaS vendors provide an excellent platform for conducting the analysis.

### **How is Big Data Related to Cloud Computing?**

Hence, from the above description, we can see that Cloud enables “As-a-Service” pattern by abstracting the challenges and complexity through a scalable and elastic self-service application. Big data requirement is same where distributed processing of massive data is abstracted from the end users.

There are multiple benefits of Big data analysis in Cloud.

##### **Improved analysis**

With the advancement of Cloud technology, big data analysis has become more improved causing better results. Hence, companies prefer to perform big data analysis in the Cloud. Moreover, Cloud helps to integrate data from numerous sources.

##### **Simplified Infrastructure**

Big Data analysis is a tremendous strenuous job on infrastructure as the data comes in large volumes with varying speeds, and types which traditional infrastructures usually cannot keep up with. As the Cloud computing provides flexible infrastructure, which we can scale according to the needs at the time, it is easy to manage workloads.

##### **Lowering the cost**

Both Big data and Cloud technology delivers value to organizations by reducing the ownership. The Pay-per-user model of Cloud turns CAPEX into OPEX. On the other hand, Apache cut down the licensing cost of Big data which is supposed to be cost millions to build and buy. Cloud enables customers for big data processing without large-scale big data resources. Hence, both Big Data and Cloud technology are driving the cost down for enterprise purposes and bringing value to the enterprise.

##### **Security and Privacy**

Data security and privacy are two major concerns when dealing with enterprise data. Moreover, when your application is hosted on a Cloud platform due to its open environment and limited user control security becomes a primary concern. On the other hand, being an open source application, Big data solution like Hadoop uses a lot of third-party services and infrastructure. Hence, nowadays system integrators bring in Private Cloud Solution that is Elastic and Scalable. Furthermore, it also leverages Scalable Distributed Processing.

Besides that Cloud data is stored and processed in a central location commonly known as Cloud storage server. Along with it the service provider and the customer signs a service level agreement (SLA) to gain the trust between them. If require the provider also leverages required advanced level of security control. This enables the security of big data in Cloud computing covering the following issues:

* 1. Protecting big data from advanced threats.
  2. How Cloud service providers maintain storage and data.

There are rules associated with service level agreements for protecting

* data
* capacity
* scalability
* security
* privacy
* availability of data storage and data growth

On the other hand in many organizations, big data analytics is utilized to detect and prevent advanced threats and malicious hackers.

##### **Virtualization**

Infrastructure plays a crucial role to support any application. Virtualization technology is the ideal platform for big data. Virtualized big data applications like Hadoop provide multiple benefits which are not accessible on physical infrastructure, but it simplifies big data Management. Big data and Cloud computing point to the convergence of various technologies and trends that makes IT infrastructure and related applications more dynamic, more expendable and more modular and. Hence, Big data and Cloud computing projects rely heavily on virtualization

**How Big Companies use Big data with Cloud.**

**Nokia**

Nokia was one of the first companies to understand the advantage of big data in cloud environments (Cloudera, 2012). Several years ago, the company used individual DBMSs to accommodate each application requirement. However, realizing the advantages of integrating data into one application, the company decided to migrate to Hadoop-based systems, integrating data within the same domain, leveraging the use of analytics algorithms to get proper insights over its clients. As Hadoop uses commodity hardware, the cost per terabyte of storage was cheaper than a traditional RDBMS (Cloudera, 2012). Since Cloudera Distributed Hadoop (CDH) bundles the most popular open source projects in the Apache Hadoop stack into a single, integrated package, with stable and reliable releases, it embodies a great opportunity for implementing Hadoop infrastructures and transferring IT and technical concerns onto the vendors’ specialized teams. Nokia regarded Big Data as a Service (BDaaS) as an advantage and trusted Cloudera to deploy a Hadoop environment that copes with its requirements in a short time frame. Hadoop, and in particular CDH, strongly helped Nokia to fulfil their needs (Cloudera, 2012).

**RedBus**

RedBus is the largest company in India specialized in online bus ticket and hotel booking. This company wanted to implement a powerful data analysis tool to gain insights over its bus booking service (Kumar, 2006). Its datasets could easily stretch up to 2 terabytes in size. The application would have to be able to analyse booking and inventory data across hundreds of bus operators serving more than 10.000 routes. Furthermore, the company needed to avoid setting up and maintaining a complex in-house infrastructure. At first, RedBus considered implementing inhouse clusters of Hadoop servers to process data. However they soon realized it would take too much time to set up such a solution and that it would require specialized IT teams to maintain such infrastructure. The company then regarded Google bigQuery as the perfect match for their needs, allowing them to:

* Know how many times consumers tried to find an available seat but were unable to do it due bus overload;
* Examine decreases in bookings;
* Quickly identify server problems by analysing data related to server activity;

Moving towards big data brought RedBus business advantages. Google bigQuery armed RedBus with real-time data analysis capabilities at 20% of the cost of maintaining a complex Hadoop infrastructure (Kumar, 2006). As supported by Nokia and RedBus examples, switching towards big data enables organizations to gain competitive advantage. Additionally, BDaaS provided by big data vendors allows companies to leave the technical details for big data vendors and focus on their core business needs.

**3.7 Big Data Analytics**

Big Data analytics is the process of collecting, organizing, and analyzing a large amount of data to uncover hidden patterns, correlations, and other meaningful insights. It helps an organization to understand the information contained in their data and use it to provide new opportunities to improve their business which in turn leads to more efficient operations, higher profits, and happier customers.

To analyze such a large volume of data, Big Data analytics applications enables [big data analysts](https://www.educba.com/what-is-data-analyst/), data scientists, predictive modelers, statisticians, and other analytical performers to analyze the growing volume of structured and unstructured data. It is performed using specialized software tools and applications. Using these tools, various data operations can be performed like data mining, text mining, predictive analysis, forecasting, etc.; all these processes are performed separately and are a part of high-performance analytics. Using Big Data analytic tools and software enables an organization to process a large amount of data and provide meaningful insights that provide better business decisions in the future.

### **Key Technologies behind Big Data Analytics**

Analytics comprises various technologies that help you get the most valued information from the data.

#### **Hadoop**

The open-source framework is widely used to store a large amount of data and run various applications on a cluster of commodity hardware. It has become a key technology to be used in big data because of the constant increase in the variety and volume of data, and its distributed computing model provides faster access to data.

#### **Data Mining**

Once the data is stored in the data management system, you can use [data mining techniques](https://www.educba.com/data-mining-techniques/) to discover the patterns which are used for further analysis and answer complex business questions. With data mining, all the repetitive and noisy data can be removed and point out only the relevant information that is used to accelerate the pace of making informed decisions.

#### **Text Mining**

With text mining, we can analyze the text data from the web like the comments, likes from social media, and other text-based sources like the email; we can identify if the mail is spam. Text Mining uses technologies like machine learning or[natural language processing](https://www.educba.com/what-is-natural-language-processing/) to analyze a large amount of data and discover the various patterns.

#### **Predictive Analytics**

Predictive analytics uses data, statistical algorithms, and machine learning techniques to identify future outcomes based on historical data. It’s all about providing the best future outcomes so that organizations can feel confident in their current business decisions.

### **Benefits of Big Data Analytics**

Big Data Analytics has been popular among various organizations. Organizations like the e-commerce industry, social media, healthcare, Banking, Entertainment industries, etc., are widely using analytics to understand various patterns, collecting and utilizing customer insights, fraud detection, monitor financial market activities, etc.

**Let’s take an example of the e-commerce industry:**

e-commerce industry like Amazon, Flipkart, Myntra, and many other online shopping sites make use of big data.

They collect customer data in several ways like

* Collect information about the items searched by the customer.
* Information regarding their preferences.
* Information about the popularity of the products and many other data.
* Using these kinds of data, organizations derive some patterns and provide the best customer service, like
* displaying the popular products that are being sold.
* show the products that are related to the products that a customer bought.
* Provide secure money transitions and identify if there are any fraudulent transactions being made.
* Forecast the demand for the products and many more.

References

1. <https://www.geeksforgeeks.org/multimedia-database/>
2. <https://docs.mongodb.com/manual/core/gridfs/>
3. <https://www.geeksforgeeks.org/what-is-big-data/?ref=lbp>
4. <https://www.researchgate.net/publication/297662056_Managing_Big_Data_A_Step_towards_Huge_Data_Security>
5. [What is Big Data Analytics? | Comprehensive Guide to Big Data Analytics (educba.com)](https://www.educba.com/what-is-big-data-analytics/)
6. <https://www.smartdatacollective.com/7-helpful-tips-managing-big-data/>
7. <https://acme.able.cs.cmu.edu/pubs/uploads/pdf/IoTBD_2016_10.pdf>
8. <https://www.whizlabs.com/blog/big-data-and-cloud-computing/>