INTRODUCTION:

Data has been the buzzword for ages now. Either the data being generated from large-scale enterprises or the data generated from an individual, each and every aspect of data needs to be analyzed to benefit yourself from it.

Why is Data Analytics important?

Data Analytics has a key role in improving your business as it is used to gather hidden insights, generate reports, perform market analysis, and improve business requirements.

What is the role of Data Analytics?

- **Gather Hidden Insights** – Hidden insights from data are gathered and then analyzed with respect to business requirements.
- **Generate Reports** – Reports are generated from the data and are passed on to the respective teams and individuals to deal with further actions for a high rise in business.
- **Perform Market Analysis** – Market Analysis can be performed to understand the strengths and weaknesses of competitors.
- **Improve Business Requirement** – Analysis of Data allows improving Business to customer requirements and experience.

Ways to Use Data Analytics:

Now that you have looked at what data analytics is, let's understand how we can use data analytics.

Fig: Ways to use Data Analytics

1. Improved Decision Making: Data Analytics eliminates guesswork and manual tasks. Be it choosing the right content, planning marketing campaigns, or developing products. Organizations can use the insights they gain from data analytics to make informed decisions. Thus, leading to better outcomes and customer satisfaction.

2. Better Customer Service: Data analytics allows you to tailor customer service according to their needs. It also provides personalization and builds stronger relationships with customers. Analyzed data can reveal information about customers' interests, concerns, and more. It helps you give better recommendations for products and services.

3. Efficient Operations: With the help of data analytics, you can streamline your processes, save money, and boost production. With an improved understanding of what your audience wants, you spend lesser time creating ads and content that aren't in line with your audience's interests.

4. Effective Marketing: Data analytics gives you valuable insights into how your campaigns are performing. This helps in fine-tuning them for optimal outcomes. Additionally, you can also find potential customers who are most likely to interact with a campaign and convert into leads.

Steps Involved in Data Analytics:

Next step to understanding what data analytics is to learn how data is analyzed in organizations. There are a few steps that are involved in the data analytics lifecycle. Below are the steps that you can take to solve your problems.

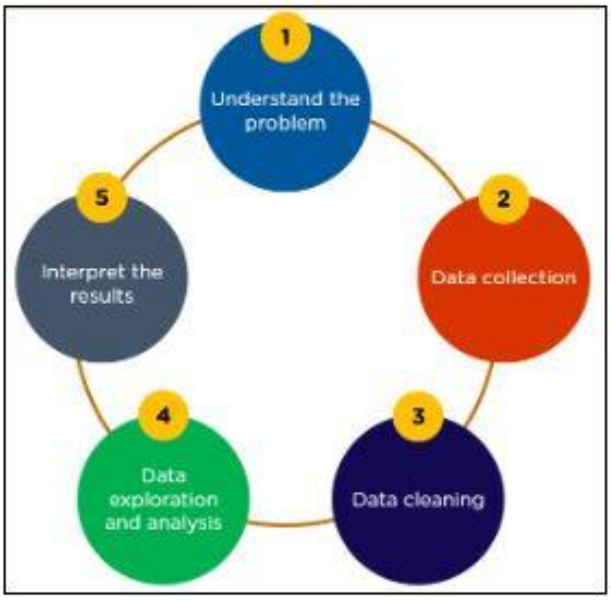


Fig: Data Analytics process steps

1. Understand the problem: Understanding the business problems, defining the organizational goals, and planning a lucrative solution is the first step in the analytics process. E-commerce companies often encounter issues such as predicting the return of items, giving relevant product recommendations, cancellation of orders, identifying frauds, optimizing vehicle routing, etc.

2. Data Collection: Next, you need to collect transactional business data and customer-related information from the past few years to address the problems your business is facing. The data can have information about the total units that were sold for a product, the sales, and profit that were made, and also when was the order placed. Past data plays a crucial role in shaping the future of a business.

3. Data Cleaning: Now, all the data you collect will often be disorderly, messy, and contain unwanted missing values. Such data is not suitable or relevant for performing data analysis. Hence, you need to clean the data to remove unwanted, redundant, and missing values to make it ready for analysis.

4. Data Exploration and Analysis: After you gather the right data, the next vital step is to execute exploratory data analysis. You can use data visualization and business intelligence tools, data mining techniques, and predictive modelling to analyze, visualize, and predict future outcomes from this data. Applying these methods can tell you the impact and relationship of a certain feature as compared to other variables.

Below are the results you can get from the analysis:

- You can identify when a customer purchases the next product.
- You can understand how long it took to deliver the product.
- You get a better insight into the kind of items a customer looks for, product returns, etc.
- You will be able to predict the sales and profit for the next quarter.
- You can minimize order cancellation by dispatching only relevant products.
- You'll be able to figure out the shortest route to deliver the product, etc.

5. Interpret the results: The final step is to interpret the results and validate if the outcomes meet your expectations. You can find out hidden patterns and future trends. This will help you gain insights that will support you with appropriate data-driven decision making.

What are the tools used in Data Analytics?

With the increasing demand for Data Analytics in the market, many tools have emerged with various functionalities for this purpose. Either open-source or user-friendly, the top tools in the data analytics market are as follows.

- **R programming** – This tool is the leading analytics tool used for statistics and data modeling. R compiles and runs on various platforms such as UNIX, Windows, and Mac OS. It also provides tools to automatically install all packages as per user-requirement.
- **Python** – Python is an open-source, object-oriented programming language that is easy to read, write, and maintain. It provides various machine learning and visualization libraries such as Scikit-learn, TensorFlow, Matplotlib, Pandas, Keras, etc. It also can be assembled on any platform like SQL server, a MongoDB database or JSON

- **Tableau Public** – This is a free software that connects to any data source such as Excel, corporate Data Warehouse, etc. It then creates visualizations, maps, dashboards etc with real-time updates on the web.
- **QlikView** – This tool offers in-memory data processing with the results delivered to the end-users quickly. It also offers data association and data visualization with data being compressed to almost 10% of its original size.
- **SAS** – A programming language and environment for data manipulation and analytics, this tool is easily accessible and can analyze data from different sources.
- **Microsoft Excel** – This tool is one of the most widely used tools for data analytics. Mostly used for clients' internal data, this tool analyzes the tasks that summarize the data with a preview of pivot tables.
- **RapidMiner** – A powerful, integrated platform that can integrate with any data source types such as Access, Excel, Microsoft SQL, Tera data, Oracle, Sybase etc. This tool is mostly used for predictive analytics, such as data mining, text analytics, machine learning.
- **KNIME** – Konstanz Information Miner (KNIME) is an open-source data analytics platform, which allows you to analyze and model data. With the benefit of visual programming, KNIME provides a platform for reporting and integration through its modular data pipeline concept.
- **OpenRefine** – Also known as GoogleRefine, this data cleaning software will help you clean up data for analysis. It is used for cleaning messy data, the transformation of data and parsing data from websites.
- **Apache Spark** – One of the largest large-scale data processing engine, this tool executes applications in Hadoop clusters 100 times faster in memory and 10 times faster on disk. This tool is also popular for data pipelines and machine learning model development.

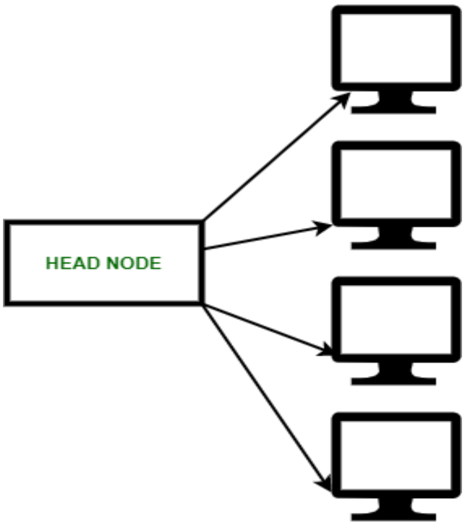
Data Analytics Applications:

Data analytics is used in almost every sector of business, let's discuss a few of them:

1. **Retail**: Data analytics helps retailers understand their customer needs and buying habits to predict trends, recommend new products, and boost their business. They optimize the supply chain, and retail operations at every step of the customer journey.
2. **Healthcare**: Healthcare industries analyse patient data to provide lifesaving diagnoses and treatment options. Data analytics help in discovering new drug development methods as well.
3. **Manufacturing**: Using data analytics, manufacturing sectors can discover new cost-saving opportunities. They can solve complex supply chain issues, labour constraints, and equipment breakdowns.
4. **Banking sector**: Banking and financial institutions use analytics to find out probable loan defaulters and customer churn out rate. It also helps in detecting fraudulent transactions immediately.

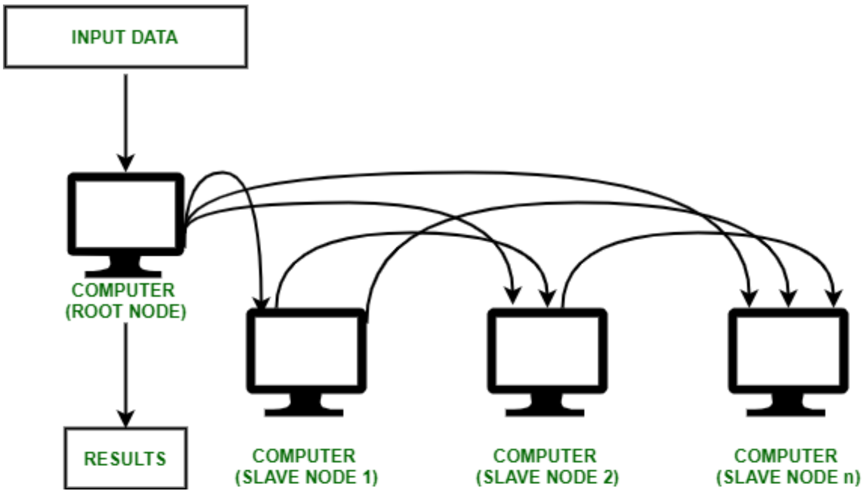
5. **Logistics**: Logistics companies use data analytics to develop new business models and optimize routes. This, in turn, ensures that the delivery reaches on time in a cost-efficient manner.

Cluster computing:



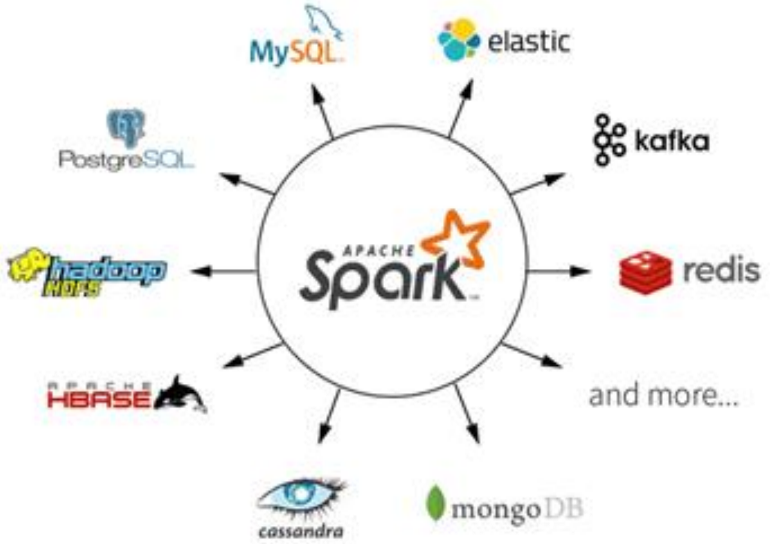
- Cluster computing is a collection of tightly or loosely connected computers that work together so that they act as a single entity.
- The connected computers execute operations all together thus creating the idea of a single system.
- The clusters are generally connected through fast local area networks (LANs)

Why is Cluster Computing important?



- Cluster computing gives a relatively inexpensive, unconventional to the large server or mainframe computer solutions.
- It resolves the demand for content criticality and process services in a faster way.
- Many organizations and IT companies are implementing cluster computing to augment their scalability, availability, processing speed and resource management at economic prices.
- It ensures that computational power is always available. It provides a single general strategy for the implementation and application of parallel high-performance systems independent of certain hardware vendors and their product decisions.

Apache Spark:



- Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing.
- The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application.
- Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming.
- Apart from supporting all these workloads in a respective system, it reduces the management burden of maintaining separate tools.

Evolution of Apache Spark

Spark is one of Hadoop's sub project developed in 2009 in UC Berkeley's AMPLab by Matei Zaharia. It was Open Sourced in 2010 under a BSD license. It was donated to Apache software foundation in 2013, and now Apache Spark has become a top level Apache project from Feb-2014.

Features of Apache Spark:

Apache Spark has following features.

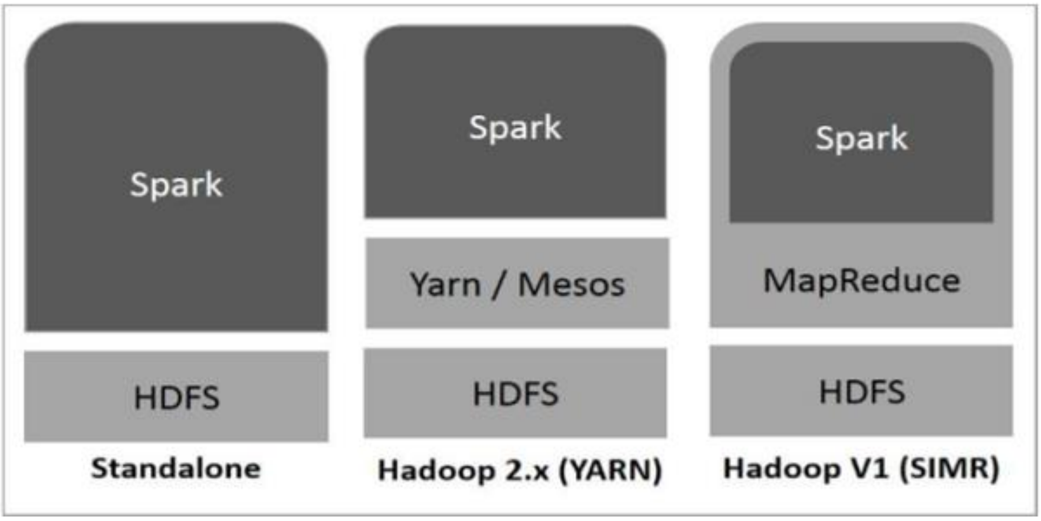
Speed – Spark helps to run an application in Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk. This is possible by reducing number of read/write operations to disk. It stores the intermediate processing data in memory.

Supports multiple languages – Spark provides built-in APIs in Java, Scala, or Python. Therefore, you can write applications in different languages. Spark comes up with 80 high-level operators for interactive querying.

Advanced Analytics – Spark not only supports 'Map' and 'reduce'. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.

Spark Built on Hadoop

The following diagram shows three ways of how Spark can be built with Hadoop components.



There are three ways of Spark deployment as explained below.

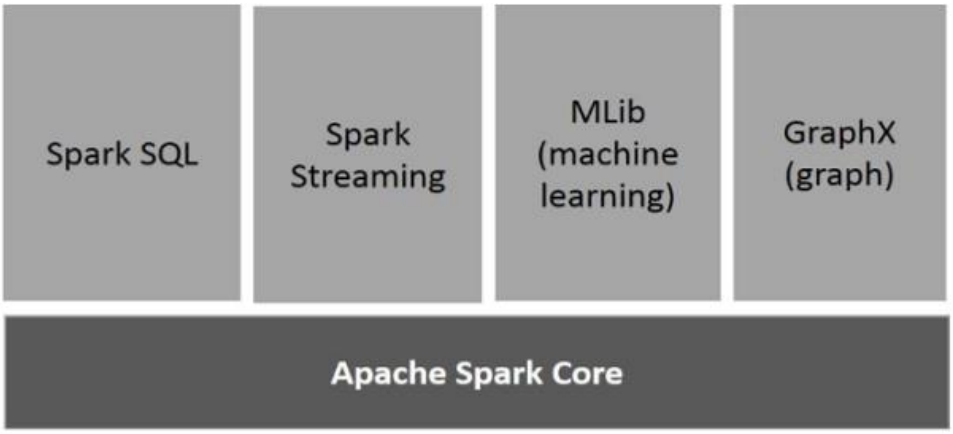
Standalone – Spark Standalone deployment means Spark occupies the place on top of HDFS(Hadoop Distributed File System) and space is allocated for HDFS, explicitly. Here, Spark and MapReduce will run side by side to cover all spark jobs on cluster.

Hadoop Yarn – Hadoop Yarn deployment means, simply, spark runs on Yarn (Yet Another Resource Negotiator) without any pre-installation or root access required. It helps to integrate Spark into Hadoop ecosystem or Hadoop stack. It allows other components to run on top of stack.

Spark in MapReduce (SIMR) – Spark in MapReduce is used to launch spark job in addition to standalone deployment. With SIMR, user can start Spark and uses its shell without any administrative access.

Components of Spark

The following illustration depicts the different components of Spark.



Apache Spark Core

Spark Core is the underlying general execution engine for spark platform that all other functionality is built upon. It provides In-Memory computing and referencing datasets in external storage systems.

Spark SQL

Spark SQL is a component on top of Spark Core that introduces a new data abstraction called SchemaRDD, which provides support for structured and semi-structured data.

Spark Streaming

Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD (Resilient Distributed Datasets) transformations on those mini-batches of data.

MLlib (Machine Learning Library)

MLlib is a distributed machine learning framework above Spark because of the distributed memory-based Spark architecture. It is, according to benchmarks, done by the MLlib developers against the Alternating Least Squares (ALS) implementations. Spark MLlib is nine times as fast as the Hadoop disk-based version of Apache Mahout (before Mahout gained a Spark interface).

GraphX

GraphX is a distributed graph-processing framework on top of Spark. It provides an API for expressing graph computation that can model the user-defined graphs by using Pregel abstraction API. It also provides an optimized runtime for this abstraction.

Scala

What is Scala?

- **Scala** is a statically typed programming language that incorporates both functional and object oriented, also suitable for imperative programming approaches.to increase scalability of applications. It is a general-purpose programming language. It is a strong static type language. In scala, everything is an object whether it is a function or a number. It does not have concept of primitive data.
- Scala primarily runs on JVM platform and it can also be used to write software for native platforms using Scala-Native and JavaScript runtimes through ScalaJs.
- This language was originally built for the Java Virtual Machine (JVM) and one of Scala's strengths is that it makes it very easy to interact with Java code.
- Scala is a Scalable Language used to write Software for multiple platforms. Hence, it got the name "Scala". This language is intended to solve the problems of Java

while simultaneously being more concise. Initially designed by Martin Odersky, it was released in 2003.

Why Scala?

- Scala is the core language to be used in writing the most popular distributed big data processing framework Apache Spark. Big Data processing is becoming inevitable from small to large enterprises.
- Extracting the valuable insights from data requires state of the art processing tools and frameworks.
- Scala is easy to learn for object-oriented programmers, Java developers. It is becoming one of the popular languages in recent years.
- Scala offers first-class functions for users
- Scala can be executed on JVM, thus paving the way for the interoperability with other languages.
- It is designed for applications that are concurrent (parallel), distributed, and resilient (robust) message-driven. It is one of the most demanding languages of this decade.
- It is concise, powerful language and can quickly grow according to the demand of its users.
- It is object-oriented and has a lot of functional programming features providing a lot of flexibility to the developers to code in a way they want.
- Scala offers many Duck Types(Structural Types)
- Unlike Java, Scala has many features of functional programming languages like Scheme, Standard ML and Haskell, including currying, type inference, immutability, lazy evaluation, and pattern matching.
- The name Scala is a portmanteau of "scalable" and "language", signifying that it is designed to grow with the demands of its users.

Where Scala can be used?

- Web Applications
- Utilities and Libraries
- Data Streaming
- Parallel batch processing
- Concurrency and distributed application
- Data analytics with Spark
- AWS lambda Expression

Cloudera Impala:

- Cloudera Impala is Cloudera's open source massively parallel processing (MPP) SQL query engine for data stored in a computer cluster running Apache Hadoop.
- Impala is the open source, massively parallel processing (MPP) SQL query engine for native analytic database in a computer cluster running Apache Hadoop.
- It is shipped by vendors such as Cloudera, MapR, Oracle, and Amazon.
- Cloudera Impala is a query engine that runs on Apache Hadoop.
- The project was announced in October 2012 with a public beta test distribution and became generally available in May 2013.
- Impala brings enabling users to issue low latency SQL queries to data stored in HDFS and Apache HBase without requiring data movement or transformation.
- Impala is integrated with Hadoop to use the same file and data formats, metadata, security and resource management frameworks used by MapReduce, Apache Hive, Apache Pig and other Hadoop software.
- Impala is promoted for analysts and data scientists to perform analytics on data stored in Hadoop via SQL or business intelligence tools.
- The result is that large-scale data processing (via MapReduce) and interactive queries can be done on the same system using the same data and metadata – removing the need to migrate data sets into specialized systems and/or proprietary formats simply to perform analysis.

Features include:

- Supports HDFS and Apache HBase storage,
- Reads Hadoop file formats, including text, LZO, SequenceFile, Avro, RCFile, and Parquet,
- Supports Hadoop security (Kerberos authentication),
- Fine-grained, role-based authorization with Apache Sentry,
- Uses metadata, ODBC driver, and SQL syntax from Apache Hive.

Databases & Types of Data and variables

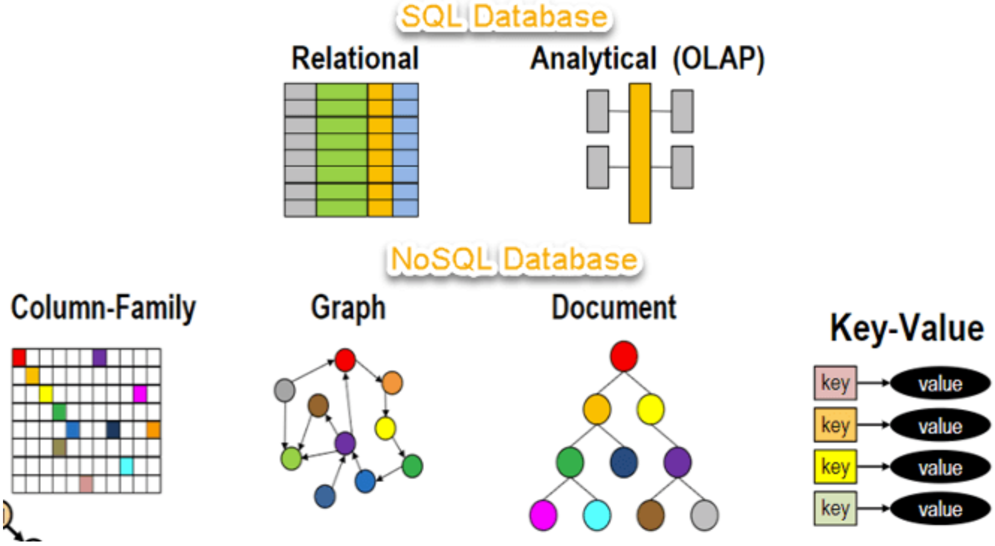
Data Base: A Database is a collection of related data.

Database Management System: DBMS is a software or set of Programs used to define, construct and manipulate the data.

Relational Database Management System: RDBMS is a software system used to maintain relational databases. Many relational database systems have an option of using the SQL.

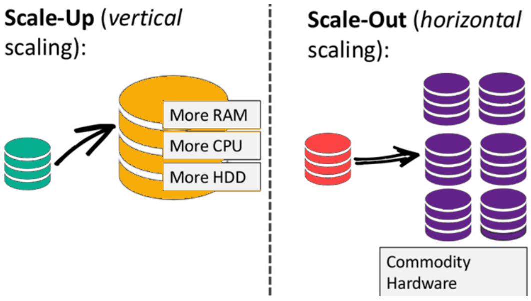
NoSQL:

- 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.
- Traditional RDBMS uses SQL syntax to store and retrieve data for further insights. Instead, a NoSQL database system encompasses a wide range of database technologies that can store structured, semi-structured, unstructured and polymorphic data.

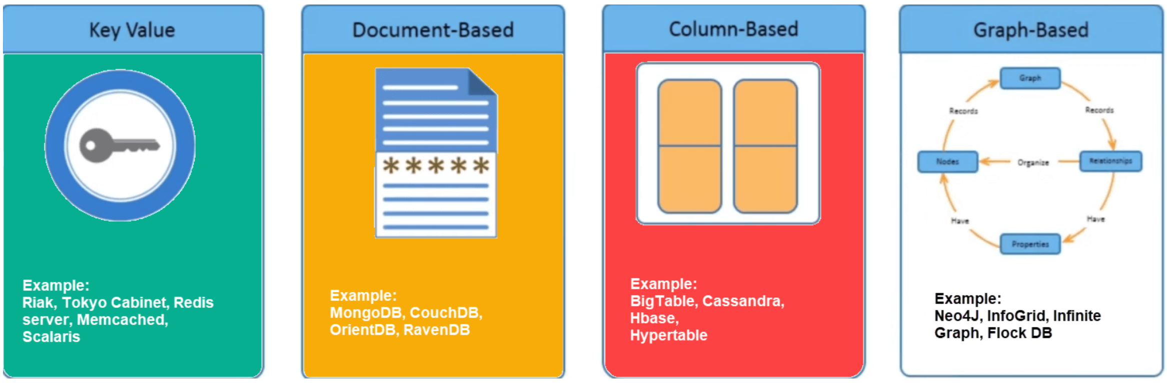


Why NoSQL?

- The concept of NoSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data. The system response time becomes slow when you use RDBMS for massive volumes of data.
- To resolve this problem, we could "scale up" our systems by upgrading our existing hardware. This process is expensive. The alternative for this issue is to distribute database load on multiple hosts whenever the load increases. This method is known as "scaling out."

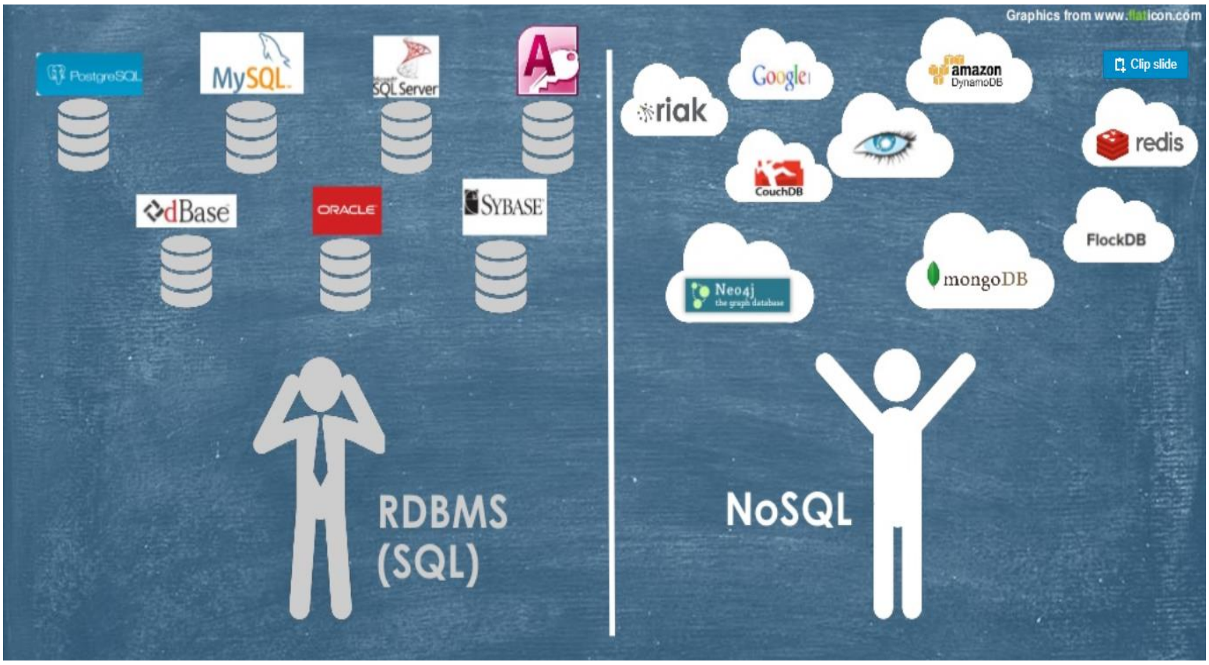

Scale-Up (*vertical* scaling):
More RAM
More CPU
More HDD
Scale-Out (*horizontal* scaling):
Commodity Hardware

Types of NoSQL Databases:


Key Value
Example:
Riak, Tokyo Cabinet, Redis server, Memcached, Scalaris
Document-Based
*****
Example:
MongoDB, CouchDB, OrientDB, RavenDB
Column-Based
Example:
BigTable, Cassandra, Hbase, Hypertable
Graph-Based
Graph
Records
Records
Nodes
Organize
Relationships
Have
Have
Properties
Example:
Neo4J, InfoGrid, Infinite Graph, Flock DB

- Document-oriented: JSON documents MongoDB and CouchDB
- Key-value: Redis and DynamoDB
- Wide-column: Cassandra and HBase
- Graph: Neo4j and Amazon Neptune

Relational Databases (SQL)	**Non-relational Databases (NoSQL)**
Oracle	**MongoDB**
MySQL	**couchDB**
SQL Server	**BigTable**



SQL vs NOSQL DB:

SQL	NoSQL
RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)	Non-relational or distributed database system.
These databases have fixed or static or predefined schema	They have dynamic schema
These databases are not suited for hierarchical data storage.	These databases are best suited for hierarchical data storage.
These databases are best suited for complex queries	These databases are not so good for complex queries
Vertically Scalable	Horizontally scalable
Follows ACID property	Follows CAP (consistency, availability, partition tolerance)

Differences between SQL and NoSQL

The table below summarizes the main differences between SQL and NoSQL databases.

	SQL Databases	NoSQL Databases
Data Storage Model	Tables with fixed rows and columns	Document: JSON documents, Key-value: key-value pairs, Wide-column: tables with rows and dynamic columns, Graph: nodes and edges
Development History	Developed in the 1970s with a focus on reducing data duplication	Developed in the late 2000s with a focus on scaling and allowing for rapid application change driven by agile and DevOps practices.
Examples	Oracle, MySQL, Microsoft SQL Server, and PostgreSQL	Document: MongoDB and CouchDB, Key-value: Redis and DynamoDB, Wide-column: Cassandra and HBase, Graph: Neo4j and Amazon Neptune
Primary Purpose	General purpose	Document: general purpose, Key-value: large amounts of data with simple lookup queries, Wide-column: large amounts of data with predictable query patterns, Graph: analyzing and traversing relationships between connected data
Schemas	Rigid	Flexible
Scaling	Vertical (scale-up with a larger server)	Horizontal (scale-out across commodity servers)
Multi-Record ACID Transactions	Supported	Most do not support multi-record ACID transactions. However, some—like MongoDB—do.
Joins	Typically required	Typically not required

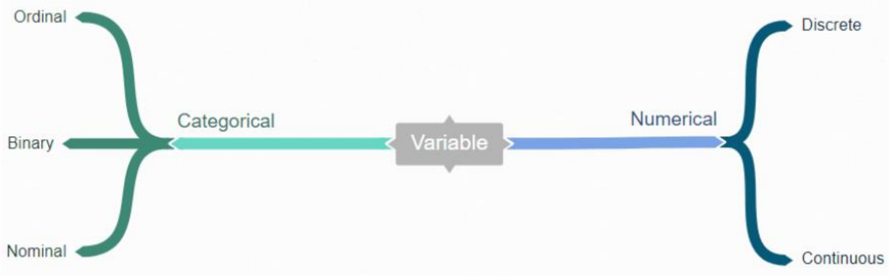
	SQL Databases	NoSQL Databases
Data to Object Mapping	Requires ORM (object-relational mapping)	Many do not require ORMs. MongoDB documents map directly to data structures in most popular programming languages.

Benefits of NoSQL

- The NoSQL data model addresses several issues that the relational model is not designed to address:
- Large volumes of structured, semi-structured, and unstructured data.
- Object-oriented programming that is easy to use and flexible.
- Efficient, scale-out architecture instead of expensive, monolithic architecture.

Variables:

- Data consist of individuals and variables that give us information about those individuals. An individual can be an object or a person.
- A variable is an attribute, such as a measurement or a label.
- Two types of Data
  - Quantitative data(Numerical)
  - Categorical data

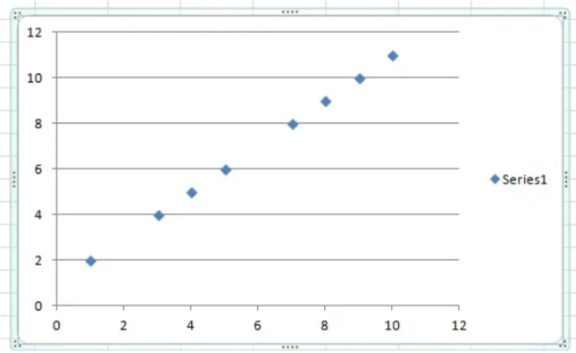


- **Quantitative Variables:** Quantitative data, contains numerical that can be added, subtracted, divided, etc.

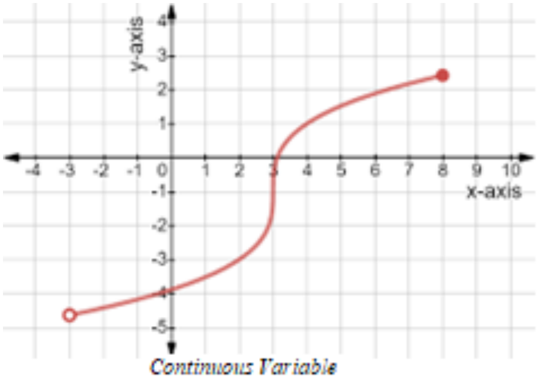
There are two types of quantitative variables: **discrete and continuous.**

Discrete vs continuous variables

Type of variable	**What does the data represent?**	**Examples**
Discrete variables	Counts of individual items or values.	• Number of students in a class • Number of different tree species in a forest
Continuous variables	Measurements of continuous or non-finite values.	• Distance • Volume • Age



Discrete variables on a scatter plot.



Categorical variables: Categorical variables represent groupings of some kind. They are sometimes recorded as numbers, but the numbers represent categories rather than actual amounts of things.

There are three types of categorical variables: **binary, nominal, and ordinal variables.**

Type of variable	**What does the data represent?**	**Examples**
Binary variables	Yes/no outcomes.	• Heads/tails in a coin flip • Win/lose in a football game
Nominal variables	Groups with no rank or order between them.	• Colors • Brands • ZIP CODE
Ordinal variables	Groups that are ranked in a specific order.	• Finishing place in a race • Rating scale responses in a survey*

Missing Imputations:

Imputation is the process of replacing missing data with substituted values.

Types of missing data

Missing data can be classified into one of three categories

1. **MCAR**

Data which is **M**issing **C**ompletely **A**t **R**andom has *nothing* systematic about which observations are missing values. There is no relationship between missingness and either observed or unobserved covariates.

2. **MAR**

Missing **A**t **R**andom is weaker than MCAR. The missingness is still random, but due entirely to observed variables. For example, those from a lower socioeconomic status may be less willing to provide salary information (but we know their SES status). The key is that the missingness is not due to the values which are not observed. MCAR implies MAR but not vice-versa.

3. **MNAR**

If the data are **M**issing **N**ot **A**t **R**andom, then the missingness depends on the values of the missing data. Censored data falls into this category. For example, individuals who are heavier are less likely to report their weight. Another example, the device measuring some response can only measure values above .5. Anything below that is missing.

There can be two types of gaps in Data:

1. Missing Data Imputation
2. Model based Technique

Imputations: (Treatment of Missing Values)

1. **Ignore the tuple**: This is usually done when the class label is missing (assuming the mining task involves classification). This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.
2. **Fill in the missing value manually**: In general, this approach is time-consuming and may not be feasible given a large data set with many missing values.
3. **Use a global constant to fill in the missing value**: Replace all missing attribute values by the same constant, such as a label like *"Unknown"* or -∞. If missing values

are replaced by, say, *"Unknown,"* then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common-that of *"Unknown."* Hence, although this method is simple, it is not foolproof.

4. **Use the attribute mean to fill in the missing value**: Considering the average value of that particular attribute and use this value to replace the missing value in that attribute column.
5. **Use the attribute mean for all samples belonging to the same class as the given tuple**:

   For example, if classifying customers according to *credit risk*, replace the missing value with the average *income* value for customers in the same credit risk category as that of the given tuple.
6. **Use the most probable value to fill in the missing value**: This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction. For example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for *income*.

Need for Business Modelling:

The main need of Business Modelling for the Companies that embrace big data analytics and transform their business models in parallel will create new opportunities for revenue streams, customers, products and services. ... Having a big data strategy and vision that identifies and capitalizes on new opportunities.

Analytics applications to various Business Domains

Application of Modelling in Business:

- Applications of Data Modelling can be termed as **Business analytics.**
- **Business analytics** involves the collating, sorting, processing, and studying of business-related data using statistical models and iterative methodologies. The goal of BA is to narrow down which datasets are useful and which can increase revenue, productivity, and efficiency.
- Business analytics (BA) is the combination of skills, technologies, and practices used to examine an organization's data and performance as a way to gain insights and make data-driven decisions in the future using statistical analysis.

Although business analytics is being leveraged in most commercial sectors and industries, the following applications are the most common.

1. Credit Card Companies

Credit and debit cards are an everyday part of consumer spending, and they are an ideal way of gathering information about a purchaser's spending habits, financial situation, behavior trends, demographics, and lifestyle preferences.

2. Customer Relationship Management (CRM)

Excellent customer relations is critical for any company that wants to retain customer loyalty to stay in business for the long haul. CRM systems analyze important performance indicators such as demographics, buying patterns, socio-economic information, and lifestyle.

3. Finance

The financial world is a volatile place, and business analytics helps to extract insights that help organizations maneuver their way through tricky terrain. Corporations turn to business analysts to optimize budgeting, banking, financial planning, forecasting, and portfolio management.

4. Human Resources

Business analysts help the process by pouring through data that characterizes high performing candidates, such as educational background, attrition rate, the average length of employment, etc. By working with this information, business analysts help HR by forecasting the best fits between the company and candidates.

5. Manufacturing

Business analysts work with data to help stakeholders understand the things that affect operations and the bottom line. Identifying things like equipment downtime, inventory levels, and maintenance costs help companies streamline inventory management, risks, and supply-chain management to create maximum efficiency.

6. Marketing

Business analysts help answer these questions and so many more, by measuring marketing and advertising metrics, identifying consumer behavior and the target audience, and analyzing market trends.

Data Modelling Techniques in Data Analytics:

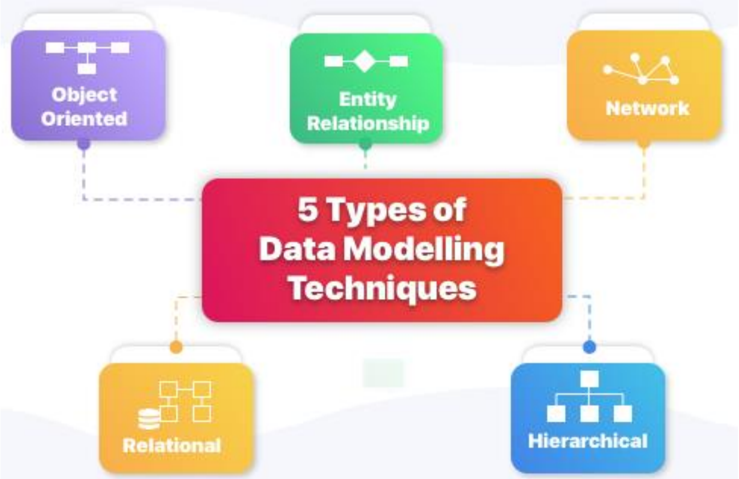
What is Data Modelling?

- Data Modelling is the process of analyzing the data objects and their relationship to the other objects. It is used to analyze the data requirements that are required for the business processes. The data models are created for the data to be stored in a database.
- The Data Model's main focus is on what data is needed and how we have to organize data rather than what operations we have to perform.
- Data Model is basically an **architect's building plan**. It is a process of documenting complex software system design as in a diagram that can be easily understood.

Uses of Data Modelling:

- Data Modelling helps create a robust design with a data model that can show an organization's entire data on the same platform.
- The database at the logical, physical, and conceptual levels can be designed with the help data model.
- Data Modelling Tools help in the improvement of data quality.
- Redundant data and missing data can be identified with the help of data models.
- The data model is quite a time consuming, but it makes the maintenance cheaper and faster.

Data Modelling Techniques:



Below given are 5 different types of techniques used to organize the data:

1. Hierarchical Technique

The hierarchical model is a tree-like structure. There is one root node, or we can say one parent node and the other child nodes are sorted in a particular order. But, the hierarchical model is very rarely used now. This model can be used for real-world model relationships.

2. Object-oriented Model

The object-oriented approach is the creation of objects that contains stored values. The object-oriented model communicates while supporting data abstraction, inheritance, and encapsulation.

3. Network Technique

The network model provides us with a flexible way of representing objects and relationships between these entities. It has a feature known as a schema representing the data in the form of a graph. An object is represented inside a node and the relation between them as an edge, enabling them to maintain multiple parent and child records in a generalized manner.

4. Entity-relationship Model

ER model (Entity-relationship model) is a high-level relational model which is used to define data elements and relationship for the entities in a system. This conceptual design provides a better view of the data that helps us easy to understand. In this model, the entire database is represented in a diagram called an entity-relationship diagram, consisting of Entities, Attributes, and Relationships.

5. Relational Technique

Relational is used to describe the different relationships between the entities. And there are different sets of relations between the entities such as one to one, one to many, many to one, and many to many.

***** End of Unit-2 *****