Week 1 learnings

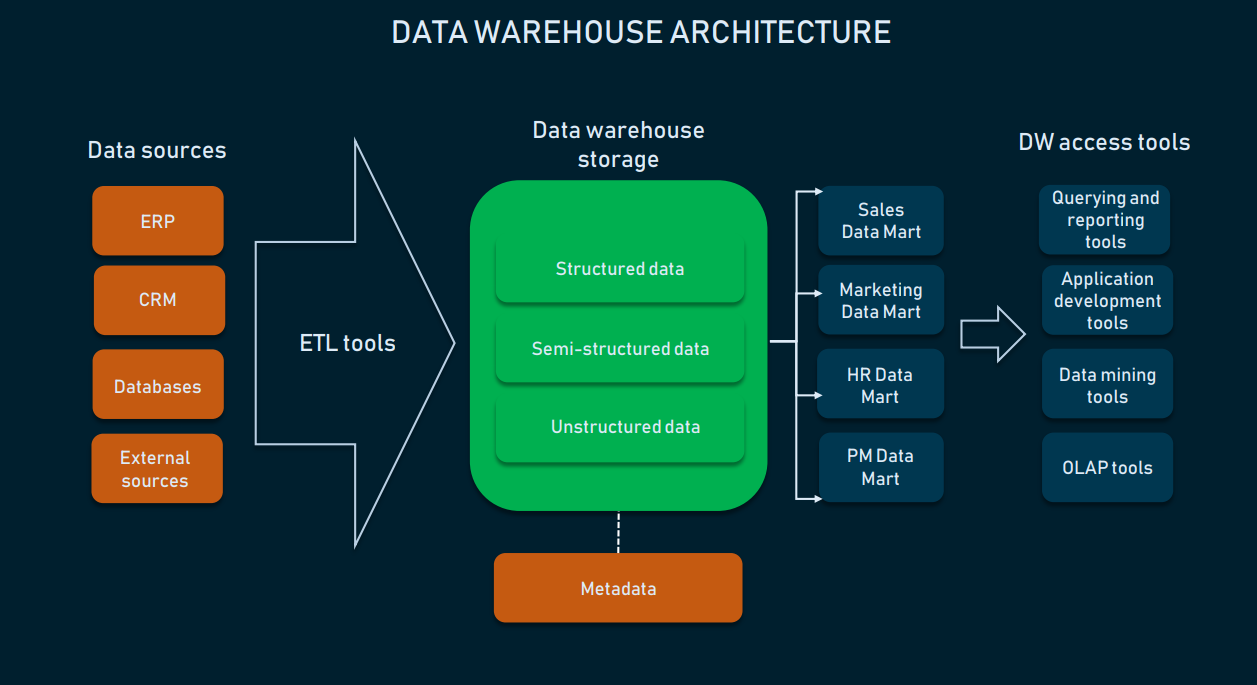
Online copy of this doc is available at : <https://docs.google.com/document/d/19O6HuKnlIxv8S8N-YgnBpSn9AVYSXDISJWSmClBIjZE/edit?usp=sharing>

# Q - What is Big Data?

* In simple words, huge amounts of data which is generated in very less time is considered as Big Data.
  + For e.g.: Petabytes of Videos getting uploaded on YouTube, Getting millions of tweets on Twitter in a minute, Millions of email sent in a minute etc.
* How do we classify any data as big data?
  + If the data has below characteristics, then we can consider it as big data.
    - **Volume**: This simply means, large amount of data. With big data, we’ll have to process high volumes of unstructured data. Initially the value of data is unknown, such as twitter data feeds, Clickstream on a webpage or a mobile app or sensor enabled equipment.
    - **Velocity**: The data which has been received at high speed and processed as close to real-time as possible. This determines the potential of data that how fast the data is generated and processed to meet the demands.
      * For e.g.: There are more than 3.5 billion searches per day made on Google. Also, Facebook users are increasing by 22%(Approx.) Year by year.
    - **Variety**: It refers to the nature of data that is structured, semi-structured and unstructured data.
      * Structured data: This data is basically an organized data. This type of data has defined length and format. For eg : xlsx, csv file or data in relational database.
      * Semi- Structured data: This data is basically a semi-organised data. This type of data does not have a concrete structure. Log and json 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 inconsistency in data, like biases, noise and abnormality. In general it checks how accurate and truthful the data set is and how much we can rely on it.
    - **Value**: It refers, how much knowledge we can discover from the data in hand, to achieve/solve the business problem or maybe to make a good profit.

# Q - What is the role of Data Engineer?

1. Majorly they **Extract, Transform and Load** the data.
   1. **Extracting data**: The information is located somewhere, so first we have to extract it. In terms of corporate data, the source can be some database, a website’s user interactions, an internal ERP/CRM system, etc. Or the source can be a sensor’s, for e.g. IoT devices. Or the data may come from public sources available online.
   2. **Transformation**: Raw data may not make much sense to the end users, because it’s hard to analyze in such form. Transformations aim at cleaning, structuring, and formatting the data sets to make data consumable for processing or analysis. In this form, it can finally be taken for further processing or queried from the [reporting layer](https://www.altexsoft.com/blog/data-analytics-tools/).
   3. **Data storing/transition**: The main architectural point in any data pipeline is storages. We need to store extracted data somewhere. In data engineering, the concept of a [data warehouse](https://www.altexsoft.com/blog/enterprise-data-warehouse-concepts/) embodies an ultimate storage for all data gathered for analytical purposes.



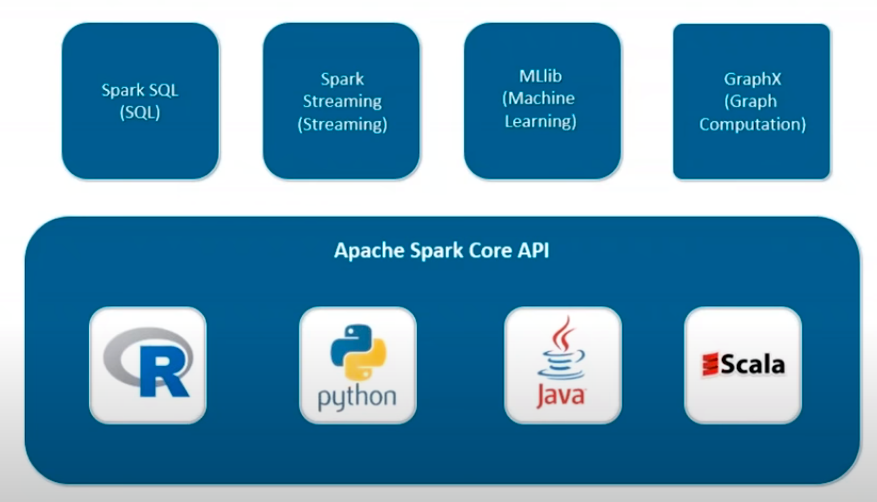
**ETL in Data warehousing**

1. DE are also responsible for **designing the architecture of data platforms**.
2. They automate tasks by **creating a Data Pipeline**.
3. They also maintain the **Data Pipeline** by testing the reliability and performance at each step.
4. They also **monitored the automated jobs** by checking the performance and stability of the data pipeline.

# Q - What is Apache Spark?

Refer this url for more info : <https://www.tutorialspoint.com/pyspark/pyspark_introduction.htm>

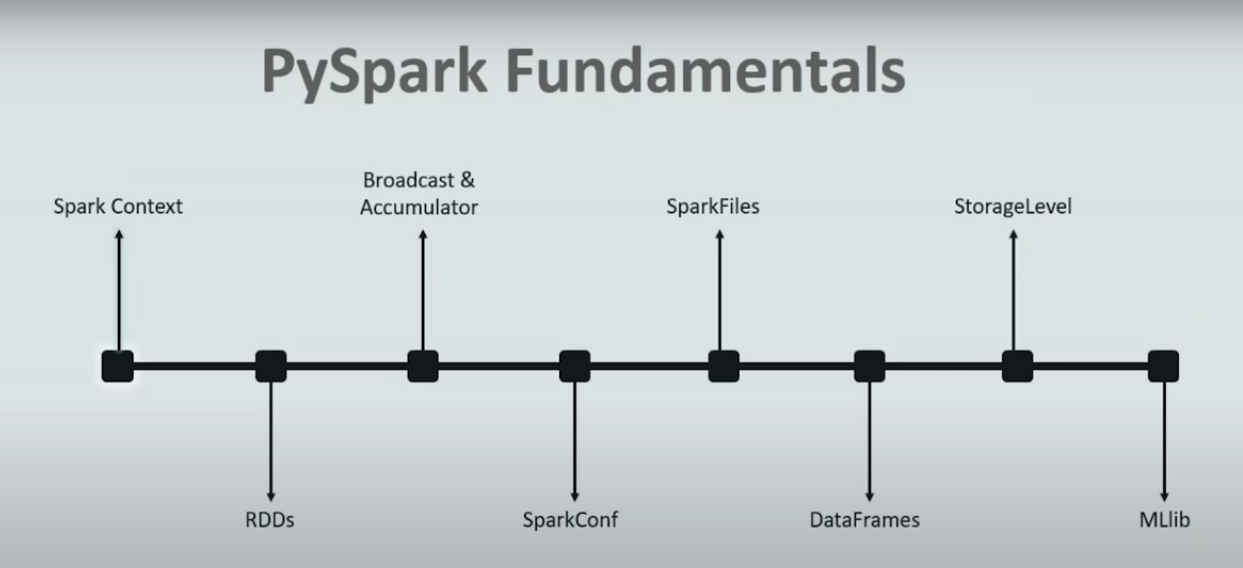
1. Apache Spark is an Open source framework used for processing data quickly. It came into picture as **Apache Hadoop MapReduce was performing batch processing only and lacked a real-time processing feature**. It offers 2 ways to process data.
   1. Real-time data processing: This is achieved via in memory computation.
   2. Batch data processing: This is achieved via scheduling a program.
2. **Spark EcoSystem**

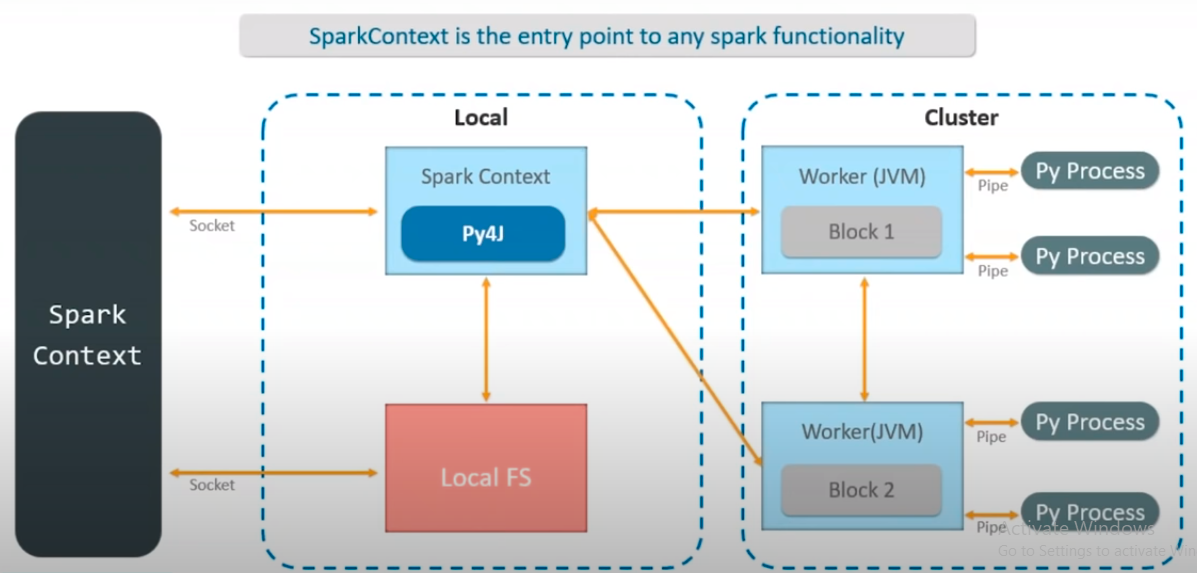
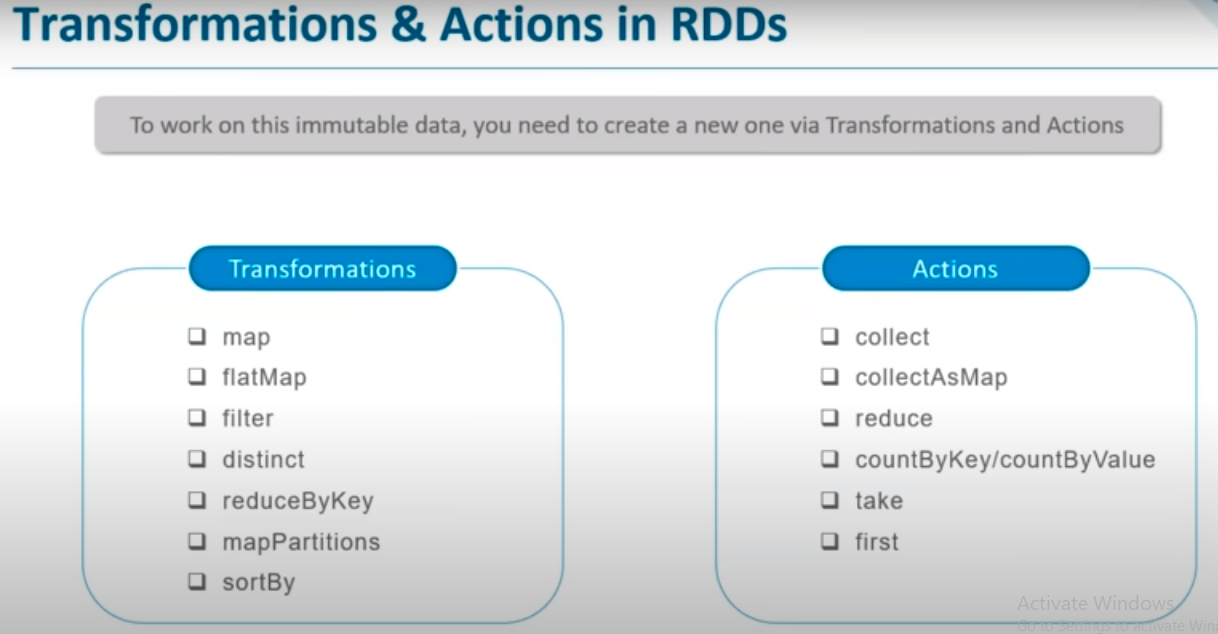


* 1. **Spark SQL** – It is used to leverage the power of SQL queries on the spark data which can be accessed in the form of RDDs and other external sources.
  2. **Spark Streaming** – This component provides real time processing of data and batch processing.
  3. **Mlib(Machine Learning)** – This component is used to create and deploy scalable machine learning pipelines.
  4. **GraphX(Graph Computation)** – Spark also comes with a library to manipulate graph databases and perform computations called GraphX.

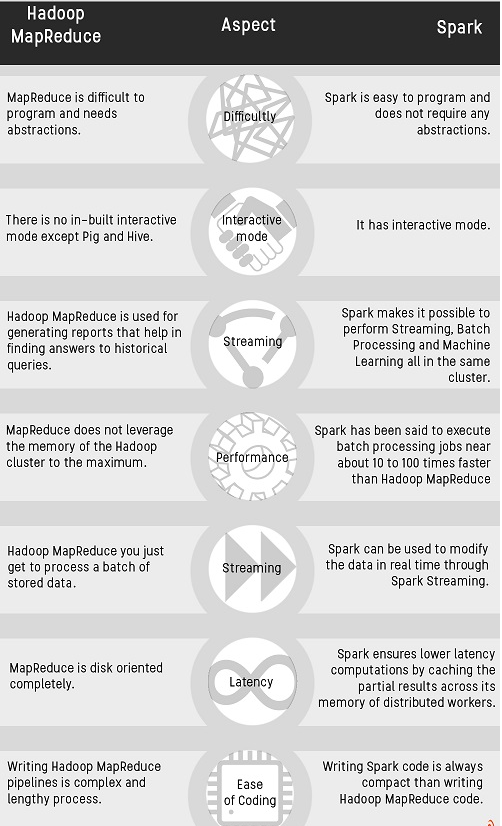
1. **Features of Spark**
   1. **Fast Processing** – This is achieved via RDDs
   2. **Flexibility** – As it supports multiple languages like Java, Scala, Python and R.
   3. **In-memory Computation** - Spark stores the data in the RAM of servers which allows quick access and in turn accelerates the speed of analytics
   4. **Real-Time processing** - Spark is able to process real-time streaming data. Unlike MapReduce which processes only stored data, Spark is able to process real-time data and is, therefore, able to produce instant outcomes.
   5. **Better analytics** – This is achieved using components like Spark SQL, Machine Learning libraries, GraphX analytics etc.

# Q - Spark Fundamentals?



1. **Spark Context**
   1. It is the heart or starting point of any Spark application. It establishes the connection to a Spark execution environment.
   2. Through the Spark context object we can create RDDs, accumulators and Broadcast variables, schedule and run jobs and much more.
   3. **If we use PySpark, it uses Py4J library internally to launch JVM and creates JavaSparkContext. By default, PySpark has SparkContext available as ‘sc’, so creating a new SparkContext won't work.**
   4. SparkContext has many parameters. Most commonly used is master and appName.
      1. **Master** - It is the url of the cluster it connects to.
      2. **appName** - it is the name of the job.
   5. 
2. **RDDs - Resilient Distributed dataset[**[**https://www.youtube.com/watch?v=e5ol7oyKV0A&ab\_channel=edureka%21**](https://www.youtube.com/watch?v=e5ol7oyKV0A&ab_channel=edureka%21)**]**
   1. It is the core data structure of spark. It can contains/holds any type of Java, Python, Scala and R objects(Classes, variables etc)
   2. It is immutable, which means, once initialised with some values then cannot be changed.
   3. RDDs are the elements that run and operate on multiple nodes to do parallel processing on a cluster.
   4. It can be created using 3 methods :
      1. Parallelized collections
      2. External Sources - AWS S3, HDFS, Hive, txt, csv etc
      3. From existing RDDs, by transforming the existing RDDs and it returns the data in RDD type.
   5. Below operations can be performed on RDDs.
3. **Broadcast and Accumulators**
   1. For parallel processing, Apache Spark uses shared variables. When the driver sends a task to the executor on the cluster a copy of the shared variable goes on each node of the cluster, so we can use it for performing tasks.
   2. There are 2 types of Shared variables.
      1. **Broadcast** - Basically, to save the copy of data across all nodes, Broadcast variables are used. However, on all the machines this variable is cached, not sent on machines. Also, we can use it to broadcast some information to all the executors. These variables are immutable.
      2. **Accumulators** - For aggregating the information through associative and commutative operations, Accumulator variables are used. As an example, for a sum operation or counters (in MapReduce), we can use an accumulator.
   3. **SparkConf**
      1. It helps in setting up the configuration of Spark application.
      2. SparkConf class is instantiated when the pyspark shell is launched, like SparkContext.
      3. For example, you can write SparkConf.setAppName(“PySpark App”).setMaster(“local”)
      4. Below are some commonly used methods using which we can or get configurations.
         1. **set(key, value)** − To set a configuration property.
         2. **setMaster(value)** − To set the master URL.
         3. **setAppName(value)** − To set an application name.
         4. **get(key, defaultValue=None)** − To get a configuration value of a key.
         5. **setSparkHome(value)** − To set Spark installation path on worker nodes.
   4. **SparkFiles** -
      1. In Apache Spark, you can upload your files using **sc.addFile** (sc is your default SparkContext) and get the path on a worker using SparkFiles.get. Thus, SparkFiles resolve the paths to files added through SparkContext.addFile().
      2. SparkFiles contain the following classmethods −
         1. **get(filename)** - It specifies the path of the file that is added through SparkContext.addFile().
         2. **getrootdirectory()** - It specifies the path to the root directory, which contains the file that is added through the SparkContext.addFile()
   5. **StorageLevels** -
      1. In Apache Spark, StorageLevel decides whether RDD should be stored in the **memory** or should it be stored over the **disk**, or both. It also decides whether to serialize RDD and whether to replicate RDD partitions.
      2. Class definition of storage level : **class pyspark.StorageLevel(useDisk, useMemory, useOffHeap, deserialized, replication = 1)**
         1. DISK\_ONLY = StorageLevel(True, False, False, False, 1)
         2. DISK\_ONLY\_2 = StorageLevel(True, False, False, False, 2)
         3. MEMORY\_AND\_DISK = StorageLevel(True, True, False, False, 1)
         4. MEMORY\_AND\_DISK\_2 = StorageLevel(True, True, False, False, 2)
         5. MEMORY\_AND\_DISK\_SER = StorageLevel(True, True, False, False, 1)
         6. MEMORY\_AND\_DISK\_SER\_2 = StorageLevel(True, True, False, False, 2)
         7. MEMORY\_ONLY = StorageLevel(False, True, False, False, 1)
         8. MEMORY\_ONLY\_2 = StorageLevel(False, True, False, False, 2)
         9. MEMORY\_ONLY\_SER = StorageLevel(False, True, False, False, 1)
         10. MEMORY\_ONLY\_SER\_2 = StorageLevel(False, True, False, False, 2)
         11. OFF\_HEAP = StorageLevel(True, True, True, False, 1)
   6. **MLib**
      1. Apache Spark offers a Machine Learning API called MLib.
      2. It offers many algorithms, like, Classification, Clustering, Regression, recommendation etc.
   7. **Serializer** -
      1. Serialization is used for performance tuning on Apache Spark. All data that is sent over the network or written to the disk or persisted in the memory should be serialized. There are 2 types of Serializers
         1. **pyspark.MarshalSerializer** - This Serializer is faster than the pickeleSerialiser but supports only fewer datatypes.
         2. **pyspark.PickleSerialiser** - This serializer supports nearly any Python object, but may not be as fast as more specialized serializers.

# Q - Hadoop vs Spark?



# Q - Why Spark?

* 1. Spark is mainly used to process the huge volume of data on the real time.
  2. Tasks most frequently associated with Spark include ETL and SQL batch jobs across large data sets, processing of streaming data from sensors, IoT, or financial systems, and machine learning tasks.

# Q - What is Spark Submit?

* spark-submit is the single script used to submit a spark program and launches the application on the cluster. For few cluster managers, spark-submit can run the driver within the cluster like in YARN on worker nodes whilst for others it runs only on local machines.
  + <https://www.youtube.com/watch?v=r-ig8zpP3EM&ab_channel=AnalyticsExcellence>

# Some useful links

<https://www.youtube.com/watch?v=QLGrLFOzMRw&ab_channel=AnalyticsExcellence>

<https://www.youtube.com/watch?v=zC9cnh8rJd0&ab_channel=GreatLearning>

<https://www.youtube.com/watch?v=e5ol7oyKV0A&ab_channel=edureka%21>

<https://www.youtube.com/watch?v=PRzSWWsyHZg&t=1533s&ab_channel=edureka%21>