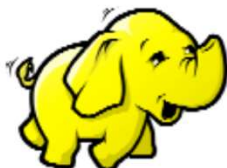


## Learning Objective:











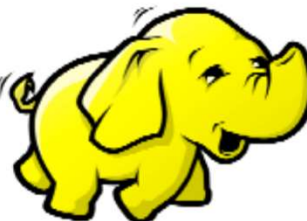
Hadoop Ecosystem Components.  
Hadoop Limitations

## University Questions on Hadoop Ecosystem:

1. Describe the structure of HDFS in a Hadoop Ecosystem using a diagram [ 10 Marks]
2. Explain Ecosystem with brief explanation of its components. [Dec16] [10Marks]
3. Explain Hadoop Ecosystem with core components. Explain it's physical architecture.  
State its limitations [May17, Dec18 ] [10 Marks]
4. What do you mean by Hadoop Ecosystem? Describe any three components of  
EcoSystem. [May18] [10Marks]



# Apache Hadoop EcoSystem

 Ambari		Provisioning, Managing and Monitoring Hadoop Clusters						Ambari
 <b>Flume</b> Log Collector	 <b>Zookeeper</b> Coordination	 <b>Sqoop</b> Data Exchange	 <b>Oozie</b> Workflow	 <b>Pig</b> Scripting	 <b>Mahout</b> Machine Learning	<b>R Connector</b> Statistics	 <b>Hive</b> SQL Query	 <b>Hbase</b> Columnar Store
 <b>Yarn Map Reduce v2</b> Distributed Processing Framework								
<b>HDFS</b> Hadoop Distributed File System 								

**Hadoop uses DFS to store big data.**

## **Limitations of Hadoop**

Hadoop can perform only **batch processing**, and data will be accessed only in a **sequential manner**. That means one has to search the entire dataset even for the simplest of jobs.

A huge dataset when processed **results** in another huge data set, which should also be **processed sequentially**. At this point, a new solution is needed to access any point of data in a single unit of time (random access).

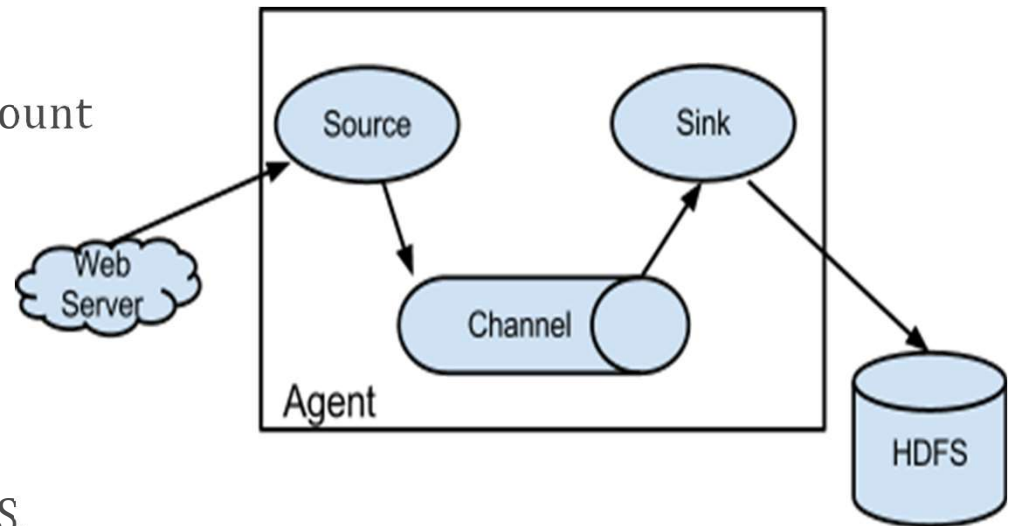
# Flume: Data Loading Tool



## Apache Flume is a tool for data ingestion in HDFS.

It collects, aggregates and transports large amount of streaming data such as

Web Servers- Click streams  
log files,  
events from various sources like  
network traffic,  
social media,  
email messages etc. to HDFS.



### Event:

A single **packet of data** passed through a system (Source → Channel → Sink) is called as an event.

In log files terminology, an event is a **line of text** followed by a **new line** character.

**1.Source:** It accepts the data from the incoming streamline and stores the data in the channel.

**2.Channel:** In general, the reading speed is faster than the writing speed. Thus, we need some buffer to match the read & write speed difference. Basically, the **buffer acts as a intermediary storage** that stores the data being transferred temporarily and therefore prevents data loss. Similarly, channel acts as the local storage or a temporary storage between the source of data and persistent data in the HDFS.

**3.Sink:** Then, our last component i.e. Sink, collects the data from the channel and commits or writes the data in the HDFS permanently.

# Sqoop

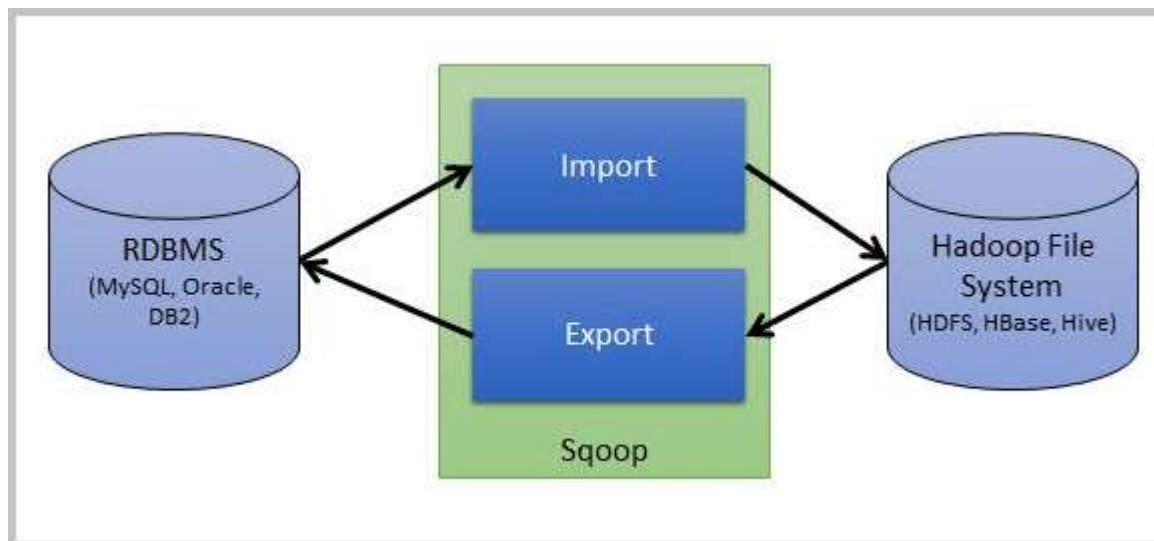


## ➤ SQL-to-Hadoop and Hadoop-to-SQL

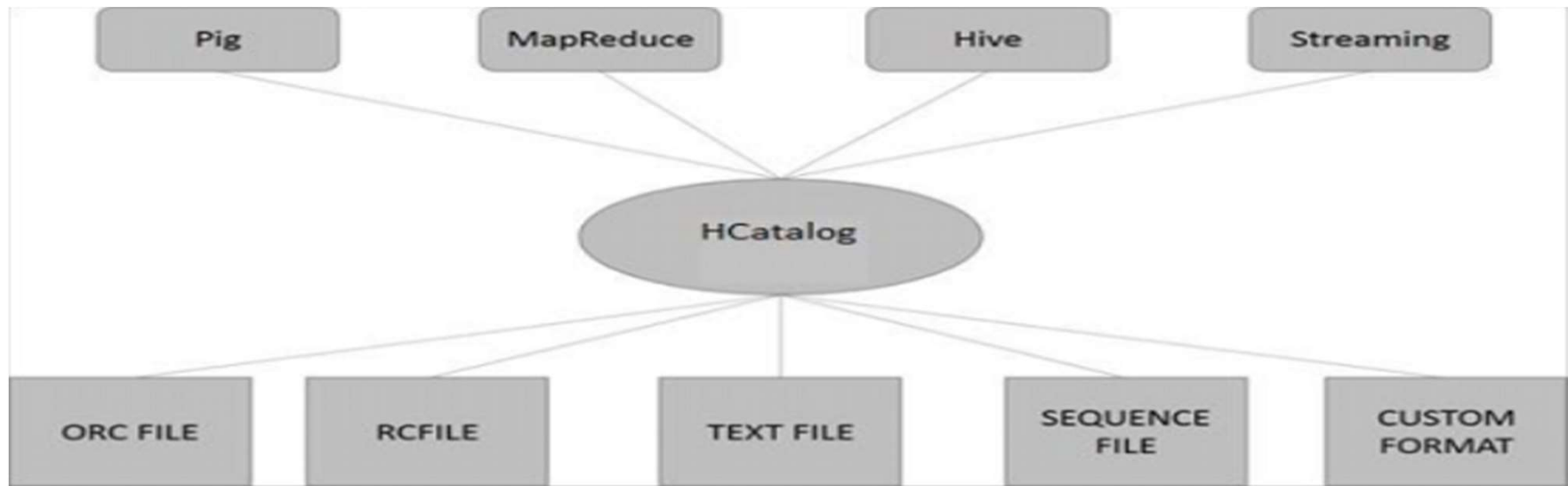
➤ Tool to transfer (import/export) data between RDMS/enterprise data warehouse to HDFS, Hive or Hbase

- The **import tool** imports individual tables from RDBMS to HDFS.
- Each **row** in a table is treated as a **record** in HDFS.
- All records are stored as **text data in text files** or as binary data in Avro and Sequence files.

- The **export tool** exports a set of files from HDFS back to an RDBMS.
- The files given as input to Sqoop contain **records**, which are called as **rows** in table.



# HCatalog



- is a **table storage management tool** for Hadoop
- It [exposes the tabular data of Hive metastore to other Hadoop applications.](#)
- It enables users with different data processing tools (Pig, MapReduce) to easily write data onto a grid.
- **HCatalog** ensures that users don't have to worry about where or in what format their data is stored.
- **HCatalog** works like a key component of Hive and it enables the users to store their data in any format and any structure.
- By default it supports RCFile, CSV, JSON, ORC file format.

# Apache Pig



- It is a run time environment tool/platform which is used to **analyze larger sets of data** representing them as data flows.
- To write **data analysis** programs, Pig provides a high-level language known as **Pig Latin**. (SQL-like)
- To analyze data using **Apache Pig**, programmers need to write scripts using **Pig Latin** language. All these scripts are internally **converted** to Map and Reduce tasks.
- Apache Pig has a component known as **Pig Engine** that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs.
- Using **Pig Latin**, programmers can perform MapReduce tasks easily without having to type complex codes in Java.
- Apache Pig uses **multi-query approach**, thereby reducing the length of codes.

**10 LoC in Apache Pig = 200 lines of Mapreduce code.**

Ultimately Apache Pig reduces the development time by almost **16 times**.

- Apache Pig provides many built-in operators to support data operations like **joins, filters, ordering** etc.
- In addition, it also provides nested data types like **tuples, bags** (tables) , and **maps** (used to represent semi-structured data types like **JSON** or **XML**) that are missing from MapReduce.
- This make Pig more flexible compare to Hive with respect to data model.



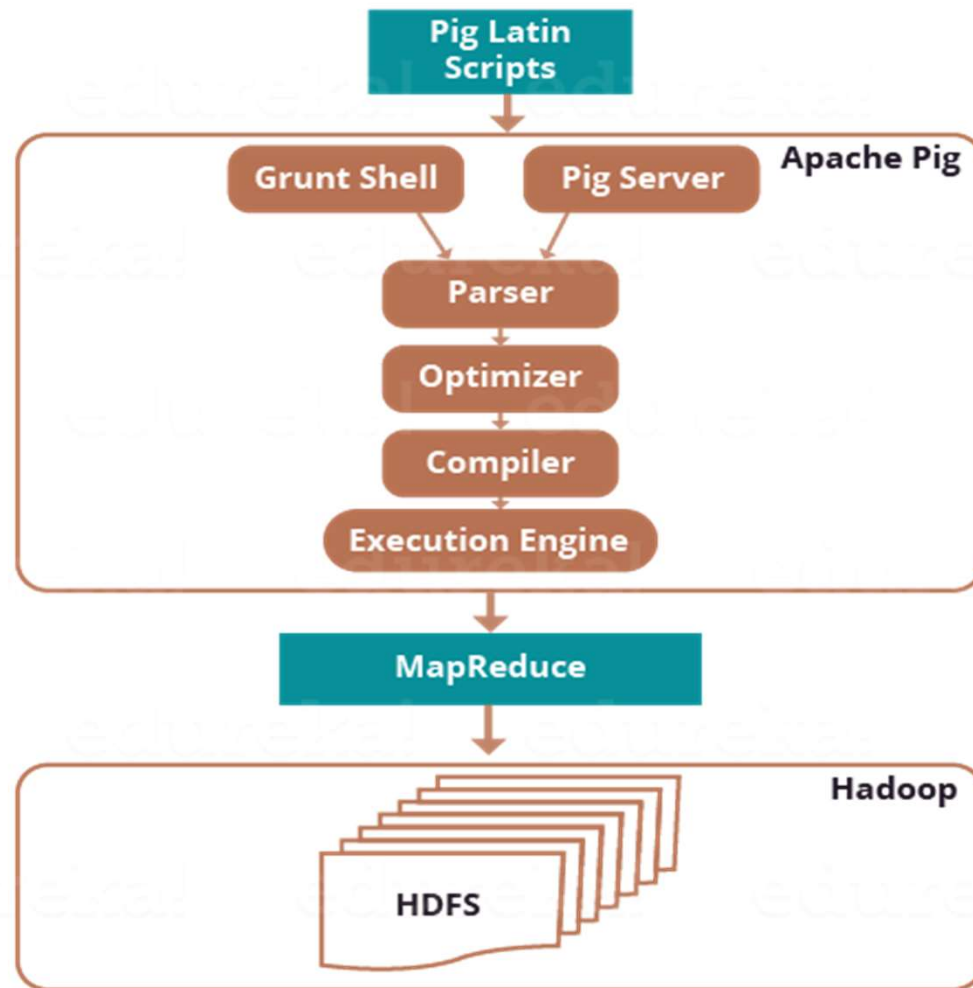
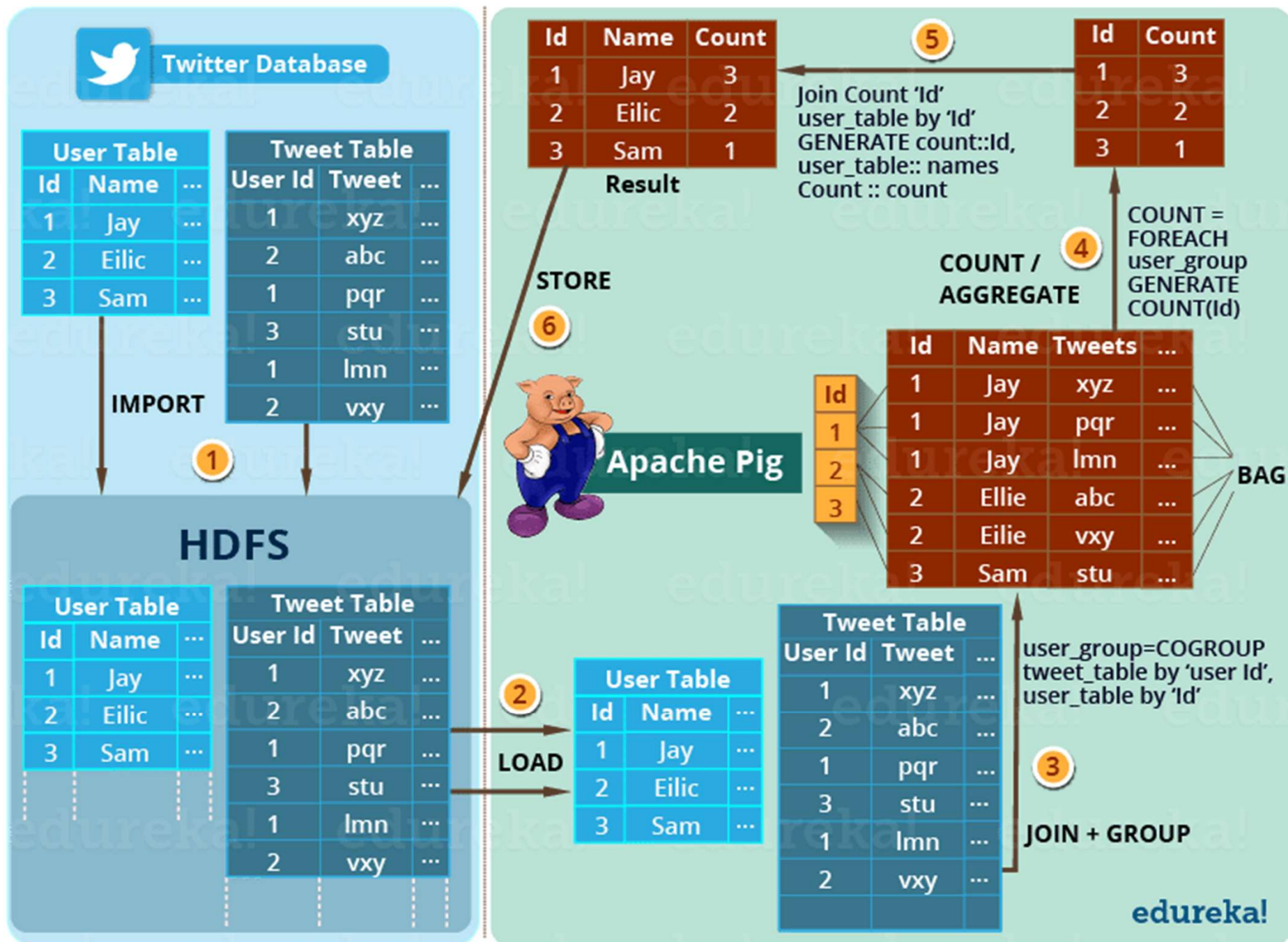


Figure: Apache Pig Architecture





# Hive



Apache Hive is a **data warehouse system** built on top of Hadoop and It is used for analyzing **structured and semi-structured** data. Hive abstracts the complexity of Hadoop MapReduce.

Basically, it provides a mechanism to **project structure** onto the data and **perform queries** written in HQL (Hive Query Language) that are similar to SQL statements.

Internally, these queries or HQL gets converted to map reduce jobs by the Hive compiler. Therefore, you don't need to worry about writing complex MapReduce programs to process your data using Hadoop.

It is targeted towards users who are comfortable with SQL. Apache Hive supports Data Definition Language (**DDL**), Data Manipulation Language (**DML**) and User Defined Functions (**UDF**).

# Hive



- Hive is a **data warehouse infrastructure tool** to large data in Hadoop.
- It resides on top of Hadoop to aggregate, **summarize** Big Data, and makes **querying** and **analyzing** easy.
- It is a platform used to develop SQL type scripts to do MapReduce operations. to allow **user-defined functions** (UDF)
- Hive's data model is based on tables, partitions, buckets.
- **Tables** corresponds to HDFS **directories** that are divided into **partitions**, which in turn can be divided into **buckets**.

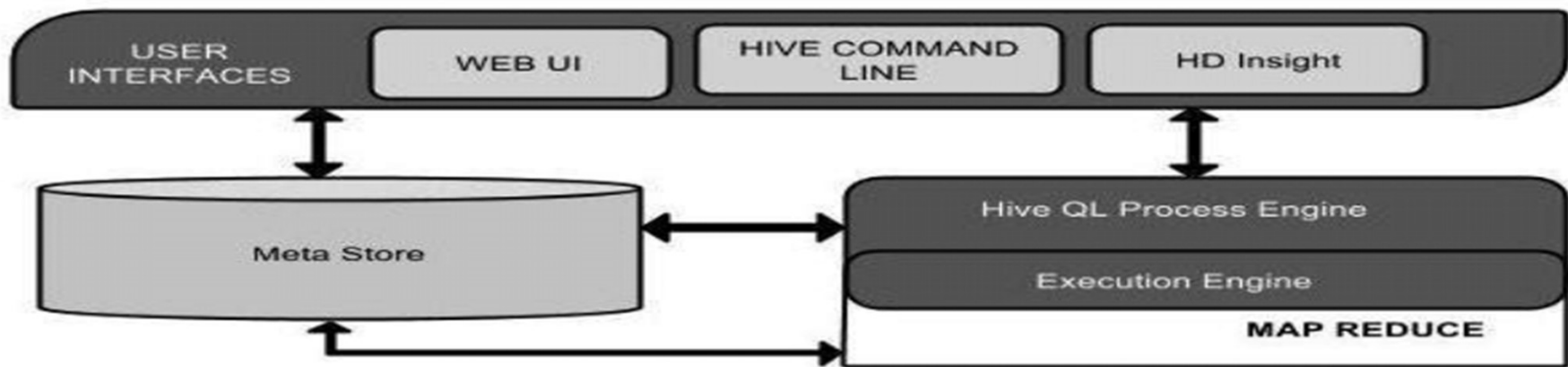
## Features of Hive

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP kind of operations.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.
- It supports MapReduce custom scripts to be plugged into queries.

## Application of Hive:

Managing Weblogs

It does not process real time queries nor update row based tables.



**Metastore = central repository** for storing all the Hive metadata information.

like **structure** of **tables** and the partitions along with the column, column type, serializer and deserializer which is required for Read/Write operation on the data present in HDFS.

The metastore comprises of two fundamental units:

A **service** that provides metastore access to other Hive services.

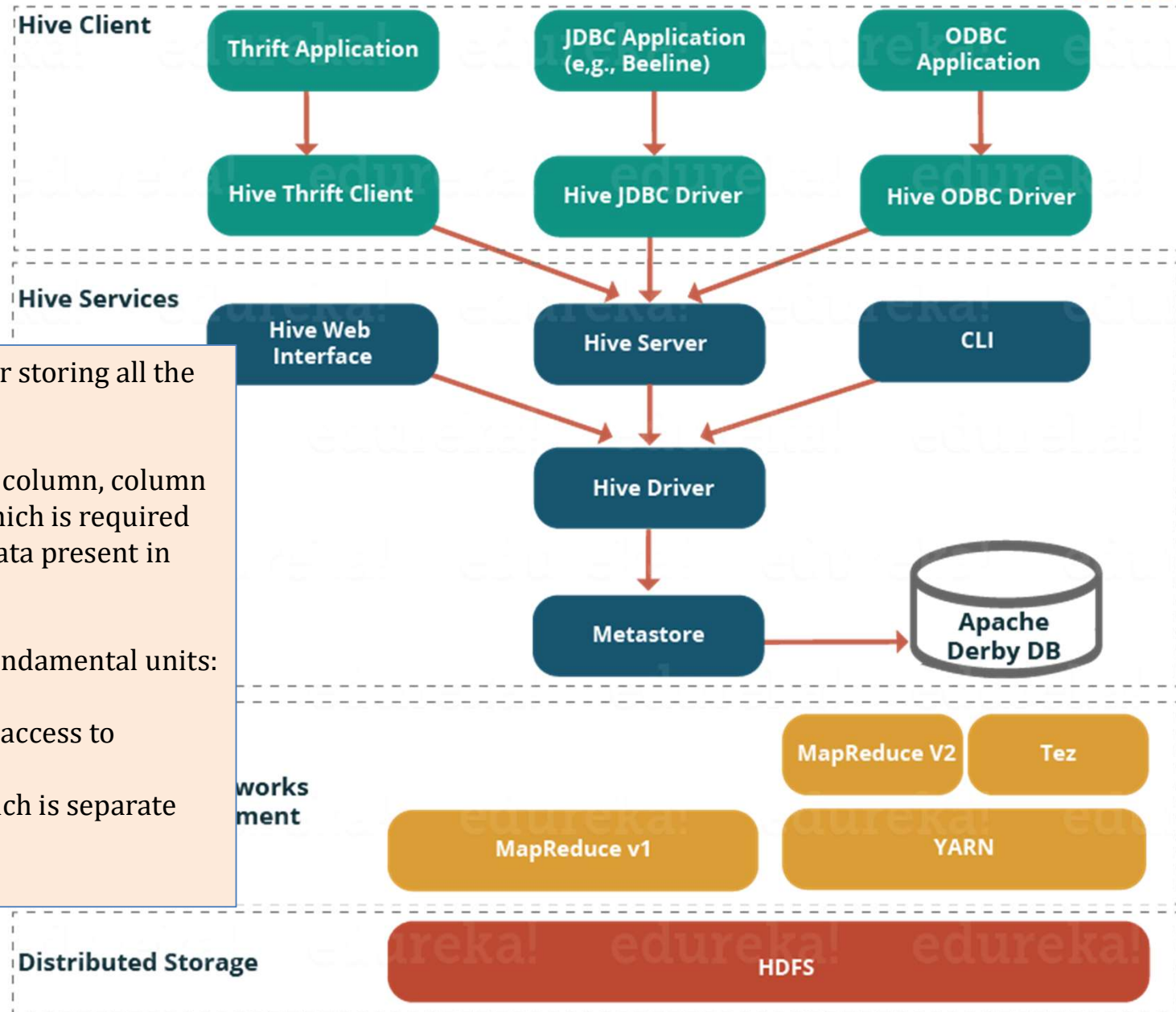
**Disk storage** for the metadata which is separate from HDFS storage.

Unit Name		
User Interface		(Windows)
Meta Store		es,
HiveQL Proces Engine		the process it.
Execution Engine		Reduce is query and the flavor
HDFS or HBASE	Hadoop distributed file system or text file or sequence files HBASE are the data storage techniques to store data into file system	



Java, C++, Python

edureka!



**Metastore** = **central repository** for storing all the Hive metadata information.  
like **structure** of **tables** and the partitions along with the column, column type, serializer and deserializer which is required for Read/Write operation on the data present in HDFS.

The metastore comprises of two fundamental units:

A **service** that provides metastore access to other Hive services.

**Disk storage** for the metadata which is separate from HDFS storage.

Hive uses a declarative language called HiveQL which is like SQL

Language Used

Pig uses a procedural data flow language called Pig Latin



Data Analyst

Mainly Used For



Researchers & Programmers



Used for Structured Data

Data



Structured & Semi-Structured Data are allowed



Schema Optional?



**Hive** Component Operates on Server Side of Cluster

Operates On



**Pig** Component Operates on Client Side of Cluster

Hive is Helpful for ETL



ETL

Pig itself is an ETL tool for Big Data



Partition



Usually No! But now with the help of serde "`org.apache.hadoop.hive.serde2.avro`", can be done

Avro file Format Support

Supports Avro File

facebook.

Developed By

YAHOO!

# Zookeeper

- Apache Zookeeper is an **open source distributed coordination service** that helps you manage a large set of hosts. and maintain shared data with robust synchronization techniques.
- It provides a single coherent view of multiple machines.
- ZooKeeper is itself a distributed application providing services for writing a distributed application.
- Distributed Service with master and slave that provides services like **storing and maintaining configuration information, naming, providing distributed synchronization**, etc
- Zookeeper allows distributed processes to coordinate with each other through a shared hierarchical name space of data registers called **znodes**.
- Zookeeper has ephemeral nodes representing different region servers. Master servers use these nodes to discover available servers.
- In addition to availability, the nodes are also used to track server failures or network partitions.
- Clients communicate with region servers via zookeeper.
- In pseudo and standalone modes, HBase itself will take care of zookeeper.





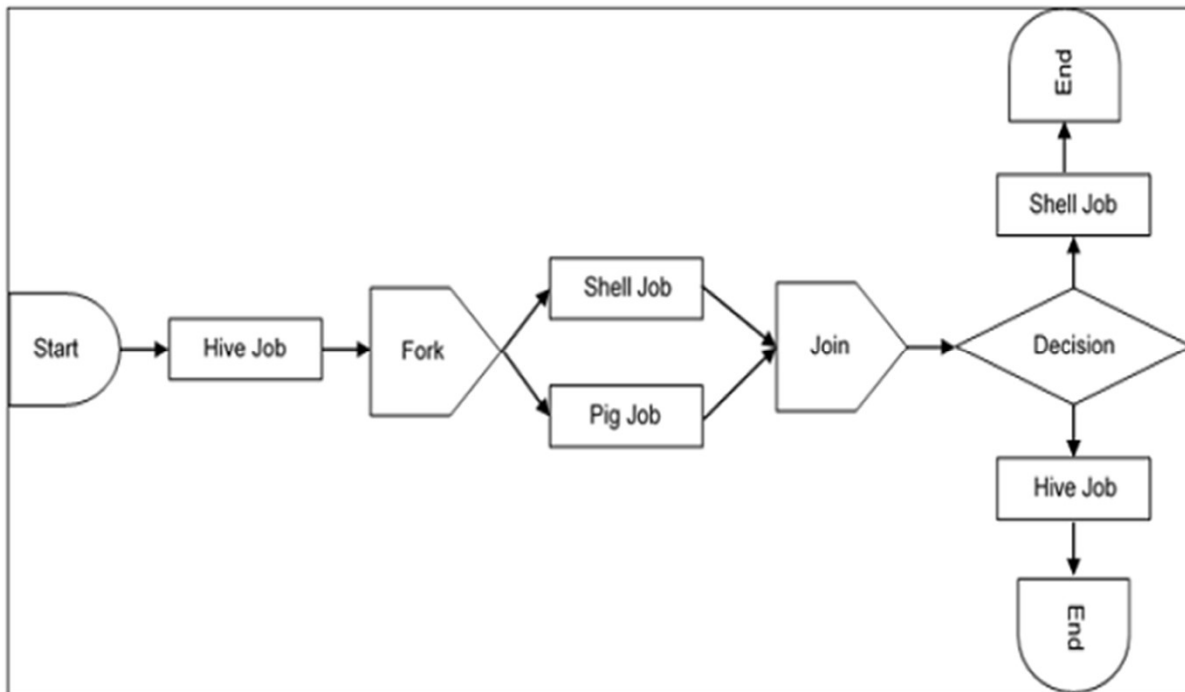
# Oozie



**Oozie** is an **Open Source Java Web-Application** available under Apache license 2.0.

It is a **Job Co-ordinator and Workflow manager for jobs** executed in Hadoop.

- An Oozie workflow is a collection of **actions** and **Hadoop jobs** arranged in a Directed Acyclic Graph (DAG), since tasks are executed in a sequence and also are subject to certain constraints.
- One of the main advantages of Oozie is that it is tightly integrated with Hadoop stack supporting various Hadoop jobs like **Hive, Pig, Sqoop, Mapreduce** as well as system-specific jobs like **Java and Shell**.
- **Oozie** detects completion of tasks through callback and polling.



Workflow=sequence of actions

# Mahout



A ***mahout*** is one who drives an elephant as its master. The name comes from its close association with Apache Hadoop which uses an elephant as its logo.

**Mahout is a scalable, machine learning and data mining library.**

It is not simply a collection pre-existing data mining algorithm  
but the algorithms of this library can be executed in a distributed fashion and  
have been written for MapReduce.

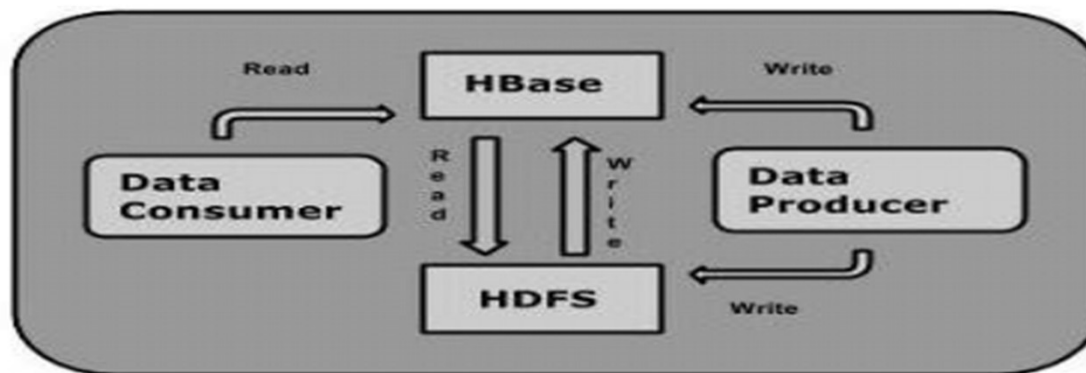
For Main groups of algorithms in Mahout.

- Recommendation/Collective Filtering.
- Classification/Categorization
- Clustering
- Frequent item-set mining/Parallel frequent pattern mining.

# HBase



- Hadoop Quick Random Access Databases.
- It is based on Google's BigTable.
- HBase is a distributed, versioned, column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.
- It is a part of the Hadoop ecosystem that provides **random real-time read/write access**
- It leverages the fault tolerance provided by the Hadoop File System (HDFS- replication support).
- Applications such as **HBase, Cassandra, couchDB, Dynamo, and MongoDB** are some of such database
- One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.



- HBase is a **column-oriented database** and the tables in it are sorted by row.
- The table schema defines only column families, which are the key value pairs.
- A table have multiple column families and each column family can have any number of columns.
- Subsequent column values are stored contiguously on the disk.
- Each cell value of the table has a timestamp.

In short, in an HBase:

- Table is a collection of rows.
- Row is a collection of column families.
- Column family is a collection of columns.
- **Column is a collection of key value pairs.**

Row Key		Column Family			Column Qualifier
Row Key		Customers		Products	
Customer ID	Customer Name	City & Country	Product Name	Price	Cell
1	Sam Smith	California, US	Mike	\$500	
2	Arijit Singh	Goa, India	Speakers	\$1000	
3	Ellie Goulding	London, UK	Headphones	\$800	
4	Wiz Khalifa	North Dakota, US	Guitar	\$2500	

Figure: HBase Table

Reference- <https://www.edureka.co/blog/hbase-architecture/>

## **Column Oriented and Row Oriented**

- Column-oriented databases are those that store data tables as sections of columns of data, rather than as rows of data. Shortly, they will have column families.

<b>Row-Oriented Database</b>	<b>Column-Oriented Database</b>
It is suitable for Online Transaction Process (OLTP).	It is suitable for Online Analytical Processing (OLAP).
Such databases are designed for small number of rows and columns.	Column-oriented databases are designed for huge tables.
Transactional	No Transactions

**Companies such as Facebook, Twitter, Yahoo, and Adobe use HBase internally.**

HDFS	HBase
HDFS is a distributed file system suitable for storing large files.	HBase is a <b>database</b> built on top of the HDFS.
HDFS does not support fast individual record lookups.	HBase provides <u>fast lookups for larger tables</u> .
It provides only sequential access of data.	HBase internally uses <b>Hash tables</b> and provides random access, and it stores the data in <b>indexed HDFS files</b> for faster lookups.
	Read/write operations may involve all rows but only small subset of all columns.
	It does not have its own query language. But it is accessible through <b>Java, thrift, REST APIs</b>

## Advantages of Hadoop

Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatically distributes the data and work across the machines and in turn, utilizes the underlying **parallelism** of the CPU cores.

Hadoop does not rely on hardware to provide **fault-tolerance** and **high availability** (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.

**Scalable:** Servers can be **added** or **removed** from the cluster dynamically and Hadoop continues to operate without interruption.

**Economical:** Another big advantage of Hadoop is that apart from being **open source**, it is compatible on all the platforms since it is Java based.

- **Huge data-** Large Data Set, high volume, variety and velocity
- **Portability Across Heterogeneous Hardware and Software Platforms**
- **Streaming Data Access-(Batch Processing)** so high throughput and low latency,
- **Moving Computation is Cheaper than Moving Data**
- **Supports tens of millions of files in a single instance.**
- **Reliable – checksum for a block, authentication, authorization**
- **Tools available**



## Limitations

1. HDFS can not be directly mounted by an existing OS. Getting data in and out from existing OS is inconvenient.
2. **File Access** can be achieved through
  1. Native Java API through C++/Ruby/Python etc
  2. Command line interface
  3. HDFS-UI web app over HTTP
3. **Security Concerns:** Hadoop security model is disabled by default. It does not provide encryption at storage and network level. So Govt. agencies and others do not prefer to keep their data in hadoop framework.
4. **Vulnerable by nature:** Framework is entirely written in java. A language most widely used by cyber criminals.
5. **Not fit for small data:** Due to high capacity design, HDFS lacks the ability to efficiently support the random reading of small files. It is not recommended for organizations with small data.
6. **Potential stability issues:** Open-source platform that is created by many developers. While improvements are constantly made, it has stability issues. Need to make sure that organizations are running the stable version.
7. **General limitations:** Google mentioned that Hadoop may not be the single solution for big data.
8. Cloud Dataflow .. Google's solution, Apache Flume has the ability to improve the efficiency and reliability of data, Collection, Integration and aggregation of data