1. **What is Data Lake?**

―A data lake is a storage repository that holds a vast amount of raw data in its native format, including structured, semi structured, and unstructured data. The data structure and requirements are not defined until the data is needed.

―If you think of a DataMart as a store of bottled water – cleansed and packaged and structured for easy consumption – the data lake is a large body of water in a more natural state. The contents of the data lake stream in from a source to fill the lake, and various users of the lake can come to examine, dive in, or take samples.

James Dixon CTO,

Founder & Chief Geek Pentaho

1. **What is Apache Tez? Is Tez replacement to MapReduce or it is running MapReduce under the hood?**

Apache Tez is a distribured task executor. Framework layer above Tez I.e. hive, pig etc submits a Directed Acyclic Graph of tasks to be executed in this case a map task or reduce task.

So Tez is not replacement of Map reduce it is executing MapR under the hood.

1. **How will you empty the trash in HDFS?**

Empty the HDFS Trash by Typing  
hdfs dfs –expunge  
You can force hdfs to immediately delete the file and skip the trash bin by giving the rm command a -skipTrash flag.  
hdfs dfs -rm -skipTrash /path/to/file/you/want/to/remove/permanently  
Trash file location by default is:  
/user/hdfs/.Trash

1. **Which tool you used to connect to Hadoop cluster?**

Putty, Winscp

1. **What is Seek time and Transfer rate in Hadoop?**
2. **How to change the replication of all blocks in HDFS?**

We can change the replication factor of a file using command:

hdfs dfs –setrep –w 3 /user/hdfs/file.txt

We can also change the replication factor of a directory using command:

hdfs dfs -setrep -R 2 /user/hdfs/test

But changing the replication factor for a directory will only affect the existing files and the new files under the directory will get created with the default replication factor (dfs.replication from hdfs-site.xml) of the cluster.

But you can temporarily override and turn off the HDFS default replication factor by passing:

-D dfs.replication=1

If we want to change the replication factor of the existing content in HDFS, which in our case is set to 4.

we can change the dfs.replication value to 4 in $HADOOP\_HOME/conf/hadoop-site.xml file. Which will start replicating to the factor of 4 for any new content that comes in. If we are looking to change for a specific file or a Directory, you can use the below commands to do that. To set replication of an individual file to 4:

$HADOOP\_HOME/bin/hadoop dfs -setrep -w 4 /path of the file

we can also do this on a Directory, which will change for all the files under it recursively.  
To change replication of entire directory under HDFS to 4:

./bin/hadoop dfs -setrep -R -w 4 /Directory path

– this is specific to a directory which we mention and if we give / (root)then it would do for all the files under it.

1. **How often data node send heartbeat to name node?**

In [HDFS](http://data-flair.training/blogs/comprehensive-hdfs-guide-introduction-architecture-data-read-write-tutorial/) DataNodes sends the heartbeat to NameNode in every **"3 sec"** (Default value of heartbeat.interval is 3 seconds.)

Heartbeat interval can be set in hdfs-site.xml as follows:

<property>

<name>heartbeat.recheck.interval</name>

<value>15</value>

<description>Determines data node heartbeat interval in seconds. </description>

</property>

1. **What happens when two users try to access the same file in the HDFS?**

HDFS NameNode supports exclusive write only. Hence, only the first user will receive the grant for file access and the second user will be rejected.

1. **What are three main components of MapR Program?**

* Drive
* Mapper
* Reducer

1. **Define Pig Latin Commands? Dump, Load, Store, Limit?**

* **Dump**: Output data to standard output device.
* **Load**: Looks for data as a delimited file in current hdfs directory.
* **Store**: Writes result to somewhere after processing is complete.
* **Limit**: View limited amount of output.

1. **How number of Mapper and reducer is decided?**

There are two conditions for no. of mappers.  
(1) No. of Mappers per slave  
(2) No. of Mappers per [MapReduce job](http://data-flair.training/blogs/hadoop-mapreduce-introduction-tutorial-comprehensive-guide/)

**(1) No. of Mappers per slave:** There is no exact formula. It depends on how many cores and how much memory you have on each slave. Generally, one mapper should get 1 to 1.5 cores of processors. So if you have 15 cores then one can run 10 Mappers per Node. So if you have 100 data nodes in [Hadoop Cluster](http://data-flair.training/blogs/install-configure-apache-hadoop-2-7-x-on-ubuntu/) then one can run 1000 Mappers in a Cluster.

**(2) No. of Mappers per MapReduce job:**The number of mappers depends on the amount of [InputSplit](http://data-flair.training/blogs/inputsplit-in-hadoop-mapreduce/) generated by trong>[InputFormat](http://data-flair.training/blogs/hadoop-inputformat-types/) (getInputSplits method). If you have 640MB file and [Data Block size](http://data-flair.training/blogs/data-blocks-hdfs-hadoop-distributed-file-system/) is 128 MB then we need to run 5 Mappers per MapReduce job.  
There are two conditions for no. of reducers.  
(1) No. of Reducers per slave  
(2) No. of Reducers per MapReduce job

**(1) No. of Reducers per slave:** It is same as No of Mappers per slave

**(2) No. of Reducers per MapReduce job:**  
The right no. reducer we can set with following formula:  
0.95 \* no. of nodes \* mapred.tasktracker.reduce.tasks.maximum  
or  
1.75 \* no. of nodes \* mapred.tasktracker.reduce.tasks.maximum

With 0.95 all of the reducers can launch immediately and start transferring map o/p when map finished.  
With 1.75 faster nodes will finish their first round of reduces and launch the second wave of reduces.

1. **Pig vs MapReduce?**

* Pig Scans whole dataset, so not suited if we are looking for small portion of data.
* Pig is highly supportive
* Pig is bit slower than Java written MR jobs.
* Pig runs series of MR programs under its hood.

1. **Managed vs External Table:**

* **Every** Object in Hive is a reference to the HDFS. Database points to a folder in HDFS, table points under folder, partition points a sub-folder under table folder and it goes on till the leaf node which is the file or data.
* Hive table is an **internal table** unless specified explicitly while creating. Internal table has strong reference to the inline file system, means if we drop the table in Hive, the data it is referencing will also be deleted. The **external table** does have reference to the data but has a loose coupling with the data. When you drop the table in Hive the data remains intact.