**ASSIGNMENT-31.2:**

**QUESTION 1 :- When should we use HBASE, list some of the scenarios for the same in real-time?**

Following are some of the key areas to be considered before finalizing HBase for your application.

**1. Data volume:**

The volume of data is the most common point to be considered. You should have peta bytes of data to be processed in a distributed environment. Otherwise, for a small amount of data, it will be stored and processed in a single node, keeping other nodes idle. So, it will be a misuse of technology framework.

**2. Application Types:**

HBase is not suitable for transactional applications, large volume MapReduce jobs, relational analytics, etc. It is also suitable when you are going for a key dependent access to your stored data.

**3. Hardware environment:**

HBase runs on top of HDFS. And HDFS works efficiently with a large number of nodes (minimum 5). So, if you have good hardware support, then HBase can be a good selection.

**4. No requirement of relational features**:

Your application should not have any requirement for RDBMS features like transaction, triggers, joins etc. If you can build your application without these features, then go for HBase.

**5. Quick access to data:**

If you need a random and real time access to your data, then HBase is a suitable candidate. It is also a perfect fit for storing large tables with multi structured data. It gives ‘flashback’ support to queries, which makes it more suitable for fetching data in a particular instance of time.

**QUESTION 2 :- What are the different modes in which Hbase can be run?**

HBase has two run modes:

**1. Standalone mode:**

* This is the default mode. Standalone mode is what is described in the Section 1.2, “Quick Start - Standalone HBase” section.
* In standalone mode, HBase does not use HDFS -- it uses the local filesystem instead -- and it runs all HBase daemons and a local ZooKeeper all up in the same JVM. Zookeeper binds to a well known port so clients may talk to HBase.

**2. Distributed mode:**

Distributed mode can be subdivided into distributed but all daemons run on a single node – i.e. a pseudo-distributed-- and fully-distributed where the daemons are spread across all nodes in the cluster

Distributed mode can be subdivided into :-

* **Pseudo-distributed mode:**

A pseudo-distributed mode is simply a fully-distributed mode run on a single host. Use this configuration testing and prototyping on HBase. Do not use this configuration for production nor for evaluating HBase performance.

* **Fully-distributed mode:**

 For a production environment, distributed mode is appropriate. In distributed mode, multiple instances of HBase daemons run on multiple servers in the cluster.

**QUESTION 3 : Need and working of zookeeper in Hbase?**

### \* A distributed HBase relies completely on Zookeeper (for cluster configuration and management).

### \* In Apache HBase, ZooKeeper coordinates, communicates, and shares state between the Masters and Region Servers.

### \* HBase has a design policy of using ZooKeeper only for transient data (that is, for coordination and state communication). Thus if the HBase’s ZooKeeper data is removed, only the transient operations are affected — data can continue to be written and read to/from HBase.

\* Apache HBase by default manages a ZooKeeper "cluster" for you. It will start and stop the ZooKeeper ensemble as part of the HBase start/stop process.

\* This variable, which defaults to true, tells HBase whether to start/stop the ZooKeeper ensemble servers as part of HBase start/stop.