# ASSIGNMENT 11.1

## TASK-1

Explain the below concepts with an example in brief.

## Nosql Databases

NoSQL is a schema-less alternative to SQL and RDBMSs designed to store, process, and analyze extremely large amounts of unstructured data.
In NoSQL databases, the principles of ACID (atomicity, consistency, isolation, and durability) are reduced. In addition, the process of normalization is not mandatory in NoSQL. Due to the size and speed of modern data, it is preferable for NoSQL databases to be de-normalized.

NoSQL databases have the following properties:

1. They have higher scale and performance
2. They use distributed computing or run over a cluster
3. They support flexible schema.
4. They are able to process both unstructured and semi-structured data.
5. There are no complex relationships, such as the ones between tables in an RDBMS.

## Types of Nosql Databases

There are following types of NoSQL databases

**Key-value data stores**: Key-value NoSQL databases emphasize simplicity and are very useful in accelerating an application to support high-speed read and write processing of non-transactional data. Stored values can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key.  Redis,Riak,DynamoDB are examples of such databases

**Document stores:** Document databases typically store self-describing JSON, XML, and BSON documents. They are similar to key-value stores, but in this case, a value is a single document that stores all data related to a specific key. Popular fields in the document can be indexed to provide fast retrieval without knowing the key. Each document can have the same or a different structure. Elastic, MongoDB,CouchDB are examples of such databases

**Wide-column stores:** Wide-column NoSQL databases store data in tables with rows and columns similar to RDBMS, but names and formats of columns can vary from row to row across the table. Wide-column databases group columns of related data together. A query can retrieve related data in a single operation because only the columns associated with the query are retrieved. In an RDBMS, the data would be in different rows stored in different places on disk, requiring multiple disk operations for retrieval. HBase and Cassandra are examples of such databases

**Graph stores:** A graph database uses graph structures to store, map, and query relationships. They provide index-free adjacency, so that adjacent elements are linked together without using an index. Neo4J and Giraph are example of such databases

## CAP Theorem

The CAP theorem applies to distributed systems that store state. Eric Brewer, at the 2000 Symposium on Principles of Distributed Computing (PODC), conjectured that in any networked shared-data system there is a fundamental trade-off between consistency, availability, and partition tolerance. In 2002, Seth Gilbert and Nancy Lynch of MIT published a formal proof of Brewer's conjecture. The theorem states that networked shared-data systems can only guarantee/strongly support two of the following three properties:

- **Consistency** (C)- A guarantee that every node in a distributed cluster returns the same, most recent, successful write. Consistency refers to every client having the same view of the data. There are various types of consistency models. Consistency in CAP (used to prove the theorem) refers to linearizability or sequential consistency, a very strong form of consistency.
- **Availability** (A)- Every non-failing node returns a response for all read and write requests in a reasonable amount of time. The key word here is every. To be available, every node on (either side of a network partition) must be able to respond in a reasonable amount of time.
- **Partition Tolerant** (P) - The system continues to function and upholds its consistency guarantees in spite of network partitions. Network partitions are a fact of life. Distributed systems guaranteeing partition tolerance can gracefully recover from partitions once the partition heals.

Choosing two out of the three properties is usually a business decision. There are many types of distributed/clustered databases differentiated based on what 2 properties are offered by them. Also CAP theorem is only applicable when the database is distributed over several nodes connected by a network. It is not relevant for single node databases. Following are some example of DBs

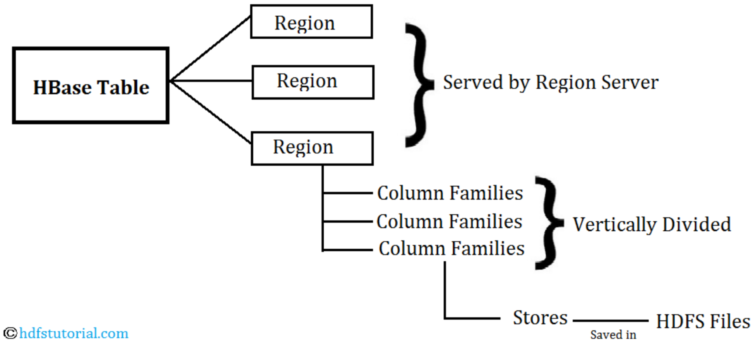
DynamoDB/Cassandra – AP (partition tolerant and available)
HBase/BigTable – CP (partition tolerant and consistent)

## HBase Architecture

HBase architecture mainly consists of three components-

- Client Library
- Master Server
- Region Server

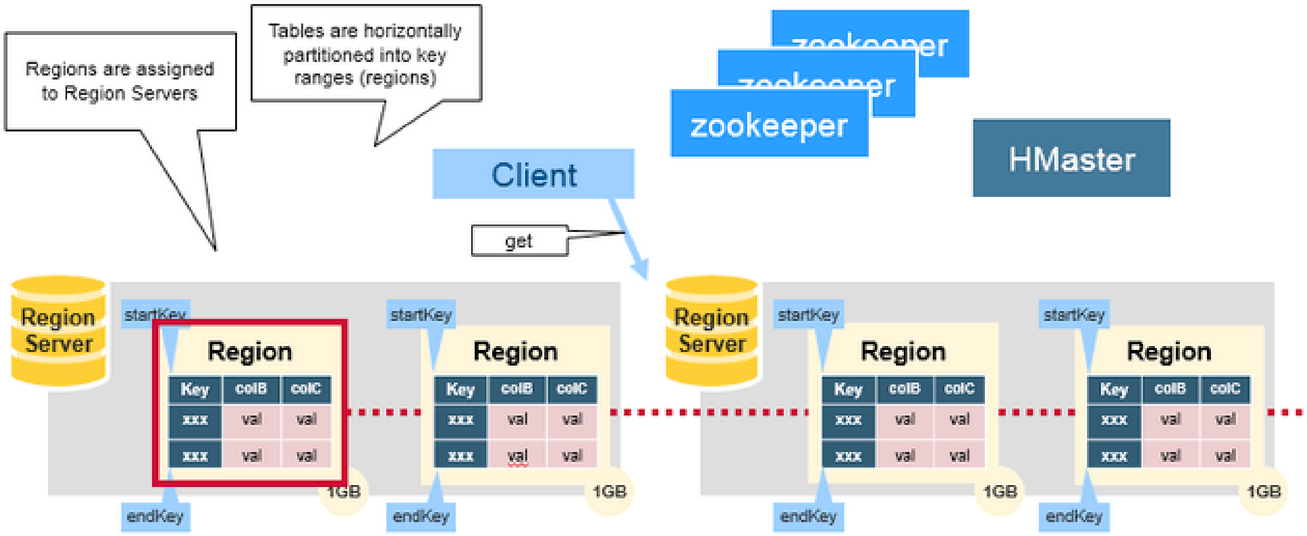
HBase tables are mainly divided into regions ( a set of sorted rows between a range of the row key) and are being served by Region servers. Further Regions are divided into Column Families vertically into Stores and then stores are saved in HDFS files in the form of hfiles.



**HBase Master Server**

The main role of Master server in HBase architecture is as follows-

- Master server assigns region to region server with the help of Apache Zookeeper
- It is also responsible for load balancing. With that mean, master server will unload the busy servers and assign that region to less occupied servers.
- Responsible for schema changes like hbase table creation, the creation of column families etc.
- Interface for creating, deleting, updating tables
- Monitor all the region servers in the cluster



**HBase regions**

HBase tables are divided horizontally into row-key range called "regions" and are managed by region server. So regions are nothing but the tables that are split horizontally into regions. Regions are assigned to a node in the cluster called Region server. A single region server can serve around 1000 regions.

**HBase Region Server**

Region server manages regions and runs on HDFS DataNodes. Many times in big data you will find the tables going beyond the configurable limit and in such cases, HBase system automatically splits the table and distributes the load to another Region Server.

The above process is called auto-sharding and is being done automatically in HBase till the time you have servers available in the rack.

Here are some of the important functions of Region server-

- It communicates with the client and handles data-related operation
- Decide the size of the region
- Handle the read and write request for all the regions under it.

**HBase Memstore**

HBase memstore is like the cache memory. When we want to write anything to HBase, first it is getting stores in memstore.

Later the data will be sent and saved in Hfiles as blocks .There will be one memstore per column family. When the memstore accumulates enough data then the entire data is transferred to Hfiles in HDFS.

**HBase Hfile**

As said, whenever any data is being written into HBase, first that gets written into memstore. And when memstore accumulates enough data, the entire sorted key-value set is written into a new Hfiles in HDFS.The write into HFile is sequential and is very fast.

**Zookeeper**

HBase uses zookeeper as a coordinator service to maintain the server state in the cluster. It tells which servers are alive and available and also provides server failure notification. It also takes care of the network partitions and client communicate with regions through Zookeeper. In Standalone Hadoop and Pseudo-Distributed Hadoop modes HBase alone will take care of Zookeeper.

## HBase vs RDBMS

Following table captures the differences between HBase and traditional RDBMS

| **Property** | **HBase** | **RDBMS** |
|---|---|---|
| | | |
| Database type | Column oriented | Row oriented |
| Schema type | Less restrictive, only column families are required | Very restrictive |
| Sparse Tables | Good with sparse tables | Not good with sparse tables |
| Scaling | Horizontal scaling by adding more nodes to the cluster | Vertical scaling by adding more memory/CPU and disk to the node running it |
| Amount of data | Unlimited as more nodes can be added to the server | Limited as per the node disk capacity |
| Transaction Integrity | No | Yes |
| Data type | Structured and Unstructured | Structured only |

### TASK-2

### Importing TSV data from HDFS to Hbase

To import TSV data which is stored in HDFS to Hbase, we will make use of **org.apache.hadoop.hbase.mapreduce.ImportTsv utility class**

First we create a sample TSV file , example.txt with the following format –

Employee ID Name Salary Location

Employee ID is the unique key which will serve as row-key in hbase table.

The contents of the file example.txt are below. It has 5 records

| | | | |
|---|---|---|---|
| 100 | Mohit | 20000 | Lucknow |
| 200 | Rajan | 30000 | Bombay |
| 301 | Priya | 24000 | Jaipur |
| 807 | Suman | 34000 | Agara |
| 899 | Sujit | 29000 | Mathura |

Step#1 we copy the example tsv file into HDFS under /files

```
[acadgild.mmisra ~]$ hadoop fs -put example.txt /files
18/08/06 15:14:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
[acadgild.mmisra ~]$ hadoop fs -ls /files
18/08/06 15:19:35 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
Found 7 items
drwxr-xr-x   - acadgild supergroup          0 2018-07-19 16:38 /files/assignment_1
drwxr-xr-x   - acadgild supergroup          0 2018-07-19 16:41 /files/assignment_2
-rw-r--r--   1 acadgild supergroup        317 2018-07-22 22:21
/files/custompiginput.txt
-rw-r--r--   1 acadgild supergroup        145 2018-08-06 15:14 /files/example.txt
-rw-r--r--   1 acadgild supergroup    2283410 2018-07-19 21:35 /files/movies.csv
-rw-r--r--   1 acadgild supergroup  709550327 2018-07-19 16:06 /files/ratings.csv
-rw-r--r--   1 acadgild supergroup        336 2018-07-19 21:50 /files/test.txt
You have new mail in /var/spool/mail/acadgild
[acadgild.mmisra ~]$
```

Step#2 –We launch HBase shell and we create a table in HBase called ‘workers’ and create two column families – ‘details’ and ‘location’. Name and salary columns would be part of ‘details’ and the location column would be part of ‘location’ column family. We scan the table to see that there are no records yet

```
[acadgild.mmisra ~]$ hbase shell
2018-08-06 15:17:11,457 WARN  [main] util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-
log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> create  'workers', 'details','location'
0 row(s) in 1.4690 seconds

=> Hbase::Table - workers
hbase(main):002:0> describe 'workers'
Table workers is ENABLED
workers
COLUMN FAMILIES DESCRIPTION
{NAME => 'details', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
KEEP_DELETED_CELLS => 'FALSE', DATA
_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS =>
'0', BLOCKCACHE => 'true', BLOCK
SIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'location', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
KEEP_DELETED_CELLS => 'FALSE', DAT
A_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS =>
'0', BLOCKCACHE => 'true', BLOC
KSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.1440 seconds



hbase(main):003:0> scan 'workers'
ROW                            COLUMN+CELL
```

```
0 row(s) in 0.0390 seconds
```

## Step#3 – We launch hbase with ImportTsv utility with the following command

hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.columns=HBASE_ROW_KEY,details:name,details:salary,location:location workers /files/example.txt

We are telling hbase that first column contains the row key (employee ID) , then next 2 columns are part of the ‘details’ column family. Last column is part of ‘location’ column family. We provide HDFS path of the tsv file as part of the command

```
[acadgild.mmisra ~]$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -
Dimporttsv.columns=HBASE_ROW_KEY,details:name,details:salary,location:location workers
/files/example.txt

2018-08-06 15:17:52,222 WARN  [main] util.NativeCodeLoader: Unable to load native-
hadoop library for your platform..
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-
log4j12-1.7.5.jar!/org/slf4j/im
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/slf4j-log4j12-1
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2018-08-06 15:17:52,694 INFO  [main] zookeeper.RecoverableZooKeeper: Process
identifier=hconnection-0x6025e1b6 conne
2018-08-06 15:17:52,704 INFO  [main] zookeeper.ZooKeeper: Client
environment:zookeeper.version=3.4.6-1569965, built
2018-08-06 15:17:52,704 INFO  [main] zookeeper.ZooKeeper: Client
environment:host.name=localhost
2018-08-06 15:17:52,704 INFO  [main] zookeeper.ZooKeeper: Client
environment:java.version=1.8.0_151
2018-08-06 15:17:52,704 INFO  [main] zookeeper.ZooKeeper: Client
environment:java.vendor=Oracle Corporation
2018-08-06 15:17:52,704 INFO  [main] zookeeper.ZooKeeper: Client
environment:java.home=/usr/java/jdk1.8.0_151/jre
2018-08-06 15:17:52,704 INFO  [main] zookeeper.ZooKeeper: Client
environment:java.class.path=/home/acadgild/install/se-
1.2.6:/home/acadgild/install/hbase/hbase-1.2.6/lib/activation-
1.1.jar:/home/acadgild/install/hbase/hbase-
1.2.6/liar:/home/acadgild/install/hbase/hbase-1.2.6/lib/apacheds-kerberos-codec-2.0.0-
M15.jar:/home/acadgild/install/hbase/htil-1.0.0-
M20.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/asm-
3.1.jar:/home/acadgild/install/hbase/hbase-1.2.6//home/acadgild/install/hbase/hbase-
1.2.6/lib/commons-beanutils-core-1.8.0.jar:/home/acadgild/install/hbase/hbase-
1.2r:/home/acadgild/install/hbase/hbase-1.2.6/lib/commons-collections-
3.2.2.jar:/home/acadgild/install/hbase/hbase-1.2.uration-
1.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/commons-daemon-
1.0.13.jar:/home/acadgild/install/hbase/s-el-
1.0.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/commons-httpclient-
3.1.jar:/home/acadgild/install/hbase/hb.6.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/commons-logging-1.2.jar:/home/acadgild/install/hbase/hbase-
1.2.6ar:/home/acadgild/install/hbase/hbase-1.2.6/lib/commons-net-
3.1.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/disar:/home/acadgild/install/hbase/hbase-1.2.6/lib/guava-
12.0.1.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/guice-stall/hbase/hbase-
```

```
1.2.6/lib/hadoop-annotations-2.5.1.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/hadoop-auth-2.install/hbase/hbase-1.2.6/lib/hadoop-common-
2.5.1.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hadoop-hdfs-
2.5.1e/acadgild/install/hbase/hbase-1.2.6/lib/hadoop-mapreduce-client-common-
2.5.1.jar:/home/acadgild/install/hbase/hbaseib/hadoop-mapreduce-client-jobclient-
2.5.1.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hadoop-mapreduce-client-
me/acadgild/install/hbase/hbase-1.2.6/lib/hadoop-yarn-client-
2.5.1.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/-common-
2.5.1.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-annotations-
1.2.6.jar:/home/acadgild/install/hb.6/lib/hbase-client-
1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-common-
1.2.6.jar:/home/acadgild/ins.2.6/lib/hbase-examples-
1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-external-blockcache-
1.2.6.jar:/stall/hbase/hbase-1.2.6/lib/hbase-hadoop-compat-
1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-it-
1.2.nstall/hbase/hbase-1.2.6/lib/hbase-prefix-tree-
1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-
procedurgild/install/hbase/hbase-1.2.6/lib/hbase-resource-bundle-
1.2.6.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/hbascadgild/install/hbase/hbase-1.2.6/lib/hbase-server-1.2.6-
tests.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/hbasacadgild/install/hbase/hbase-1.2.6/lib/htrace-core-3.1.0-
incubating.jar:/home/acadgild/install/hbase/hbase-
1.2.6/libacadgild/install/hbase/hbase-1.2.6/lib/jackson-core-asl-
1.9.13.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/jackjar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jackson-xc-
1.9.13.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jasper-runtime-
5.5.23.jar:/home/acadgild/install/hbase/hbase-
1.2.6//home/acadgild/install/hbase/hbase-1.2.6/lib/jaxb-api-
2.2.2.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jaxb-imd/install/hbase/hbase-
1.2.6/lib/jersey-client-1.9.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jersey-
core-1.9.jhbase/hbase-1.2.6/lib/jersey-json-
1.9.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jersey-server-
1.9.jar:/home/a2.6/lib/jettison-1.3.3.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/jetty-6.1.26.jar:/home/acadgild/install/hbastty-util-
6.1.26.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/joni-
2.1.2.jar:/home/acadgild/install/hbase/hbase-1ar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/jsp-2.1-6.1.14.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jsp-
d/install/hbase/hbase-1.2.6/lib/leveldbjni-all-
1.8.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/libthrift-0.9.3.se/hbase-
1.2.6/lib/metrics-core-2.2.0.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/netty-
all-4.0.23.Final.jar:/hase-1.2.6/lib/protobuf-java-
2.5.0.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/servlet-api-2.5-
6.1.14.jar:/home/e-1.2.6/lib/slf4j-api-1.7.7.jar:/home/acadgild/install/hbase/hbase-
1.2.6/lib/slf4j-log4j12-1.7.5.jar:/home/acadgild/.6/lib/spymemcached-
2.11.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/xmlenc-
0.52.jar:/home/acadgild/install/h.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/etc/hadoop/:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop//commons-
lang-2.6.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/guava-
11.0.2.jar:/home/acadtall/hadoop/hadoop-2.6.5/share/hadoop/common/lib/avro-
1.7.4.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadmmon/lib/paranamer-
2.3.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-
log4j12-1.7.5.jacadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/activation-1.1.jar:/home/acadgild/install/hadoop/hadoop-
.6.5/share/hadoop/common/lib/jaxb-impl-2.2.3-
1.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lipi-
1.7.5.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/commons-
compress-1.4.1.jar:/home/aca/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/htrace-core-
3.0.4.jar:/home/acadgild/install/hadoop/hadoop-2.6.-
2.6.5/share/hadoop/common/lib/jaxb-api-2.2.2.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/libcommons-httpclient-
```

```
3.1.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/jackson-
jaxrs-1.9.13.jild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/commons-io-
2.4.jar:/home/acadgild/install/hadoop/hadoop-2.6.hadoop/common/lib/commons-beanutils-
1.7.0.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/curcommons-net-
3.1.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/commons-
beanutils-core-1.8.0.0.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/jersey-core-1.9.jar:/home/acadgild/install/p/hadoop-
2.6.5/share/hadoop/common/lib/commons-digester-
1.8.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/had/common/lib/hadoop-auth-
2.6.5.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/httpclient-4.2.ome/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/jackson-core-asl-
1.9.13.jar:/home/acadgild/install/tall/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/hamcrest-core-
1.3.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/shp/common/lib/api-util-1.0.0-
M20.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/zookeeper-
3.41.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/jasper-
runtime-5.5.23.jar:/home/acadgild/in/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/jasper-compiler-
5.5.23.jar:/home/acadgild/install/hadoop/hadoophadoop/common/lib/jackson-xc-
1.9.13.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/netty-
3.6nt-2.6.0.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/stax-api-1.0-2.jar:/home/acadgild/inll/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/commons-codec-
1.4.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/shar/common/lib/commons-logging-
1.1.3.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/commons-
el-ar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/commons-
collections-3.2.2.jar:/home/acadgild/ioop/hadoop-2.6.5/share/hadoop/common/lib/jets3t-
0.9.0.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/coare/hadoop/common/lib/java-xmlbuilder-
0.4.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/curtty-
util-6.1.26.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/hadoop-annotations-
2.6.5.jar:adgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/hadoop-common-
2.6.5.jar:/home/acadgild/install/hadoop/hadoop--
2.6.5/share/hadoop/hdfs:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/xercesImpl-2.9.1.jar:/homeild/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/guava-11.0.2.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/sh/hdfs/lib/jsp-api-2.1.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/protobuf-java-2.5.0.jar:/adgild/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/htrace-core-
3.0.4.jar:/home/acadgild/install/hadoop/hadoop-are/hadoop/hdfs/lib/jsr305-
1.3.9.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/hdfs/lib/commons-
daemon.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/hdfs/lib/jersey-
server-1.9.jar:/home/acadgild/install/hop-2.6.5/share/hadoop/hdfs/lib/jackson-core-
asl-1.9.13.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/hp/hdfs/lib/xmlenc-0.52.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/jasper-runtime-5.5.23.jad/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/netty-
3.6.2.Final.jar:/home/acadgild/install/hadoop/hadoop-2.6.5e/hadoop/hdfs/lib/commons-
logging-1.1.3.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/commons.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/hdfs/lib/commons-cli-1.2.jar:/home/acadgild/install/hadadoop-
2.6.5/share/hadoop/hdfs/hadoop-hdfs-2.6.5-
tests.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/hd-hdfs-nfs-
2.6.5.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/commons-
lang-2.6.jar:/home/acadll/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/jetty-
6.1.26.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadooib/protobuf-java-
2.5.0.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/jaxb-impl-
2.2.3-1.jar:/hdgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/commons-
compress-1.4.1.jar:/home/acadgild/install/hadoop/had.5/share/hadoop/yarn/lib/jaxb-api-
2.2.2.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/commons-
```

```
jaxrs-1.9.13.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/xz-
1.0.jar:/home/acadgild/installhadoop-2.6.5/share/hadoop/yarn/lib/jsr305-
1.3.9.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/libsey-core-
1.9.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/jersey-json-
1.9.jar:/home/acadgildll/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/jersey-server-
1.9.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/n/lib/jackson-core-asl-
1.9.13.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/jackson-
mapper-as3.0.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/yarn/lib/zookeeper-
3.4.6.jar:/home/acadgild/install/.6.5/share/hadoop/yarn/lib/jackson-xc-
1.9.13.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/nei-1.0-
2.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/commons-codec-
1.4.jar:/home/acadgild/inop/hadoop-2.6.5/share/hadoop/yarn/lib/commons-logging-
1.1.3.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadn/lib/guice-
3.0.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/commons-
collections-3.2.2.jar:/ild/install/hadoop/hadoop-
2.6.5/share/hadoop/yarn/lib/javax.inject-1.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5//hadoop/yarn/lib/jersey-client-1.9.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/yarn/hadoop-yarn-servop/yarn/hadoop-yarn-api-
2.6.5.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/hadoop-yarn-
client-2.s-2.6.5.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/yarn/hadoop-yarn-server-nodemanager-2.6.5.jar:/hr-
2.6.5.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/hadoop-yarn-
registry-2.6.5.jar:/home/acadgi.6.5.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/yarn/hadoop-yarn-server-web-proxy-
2.6.5.jar:/home/aome/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/hadoop-
yarn-applications-unmanaged-am-launcher-2.6.5.jar:ome/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/mapreduce/lib/avro-
1.7.4.jar:/home/acadgild/install/hadoop/had2.6.5/share/hadoop/mapreduce/lib/protobuf-
java-2.5.0.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/mapduce/lib/commons-compress-
1.4.1.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/lib/xz-
1.0.jarome/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/lib/leveldbjni-
all-1.8.jar:/home/acadgild/install/haop/hadoop-
2.6.5/share/hadoop/mapreduce/lib/jersey-guice-
1.9.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/lib/log4j-
1.2.17.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/mapreduce/lib/jackson-coron-mapper-asl-
1.9.13.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/mapreduce/lib/hamcrest-core-
1.3.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/lib/snappy-
java-1.0.4.1.jar:/home/acadgild/instaladoop-2.6.5/share/hadoop/mapreduce/lib/netty-
3.6.2.Final.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoopce/lib/junit-
4.11.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/mapreduce/lib/javax.inject-
1.jar:/homer:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-
mapreduce-client-core-
2.6.5.jar:/home/aca.6.5.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-client-hs-2.6.5.jar:/homent-2.6.5-
tests.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-
mapreduce-examples-2.6.ent-shuffle-2.6.5.jar:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-client-app--client-hs-plugins-
2.6.5.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-
mapreduce-cliejar
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:java.library.path=/usr/java/packages/li
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:java.io.tmpdir=/tmp
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:java.compiler=<NA>
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:os.name=Linux
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
```

```
environment:os.arch=amd64
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:os.version=2.6.32-696.18.7.el6.x86_64
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:user.name=acadgild
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:user.home=/home/acadgild
2018-08-06 15:17:52,708 INFO  [main] zookeeper.ZooKeeper: Client
environment:user.dir=/home/acadgild
2018-08-06 15:17:52,710 INFO  [main] zookeeper.ZooKeeper: Initiating client
connection, connectString=localhost:2181base
2018-08-06 15:17:52,751 INFO  [main-SendThread(localhost:2181)] zookeeper.ClientCnxn:
Opening socket connection to s
2018-08-06 15:17:52,777 INFO  [main-SendThread(localhost:2181)] zookeeper.ClientCnxn:
Socket connection established
2018-08-06 15:17:52,789 INFO  [main-SendThread(localhost:2181)] zookeeper.ClientCnxn:
Session establishment complete00
2018-08-06 15:17:53,790 INFO  [main] Configuration.deprecation: io.bytes.per.checksum
is deprecated. Instead, use df
2018-08-06 15:17:53,868 INFO  [main]
client.ConnectionManager$HConnectionImplementation: Closing zookeeper sessionid
2018-08-06 15:17:53,871 INFO  [main-EventThread] zookeeper.ClientCnxn: EventThread
shut down
2018-08-06 15:17:53,871 INFO  [main] zookeeper.ZooKeeper: Session: 0x1650e05252e0014
closed
2018-08-06 15:17:53,959 INFO  [main] client.RMProxy: Connecting to ResourceManager at
localhost/127.0.0.1:8032
2018-08-06 15:17:54,181 INFO  [main] Configuration.deprecation: io.bytes.per.checksum
is deprecated. Instead, use df
2018-08-06 15:17:56,354 INFO  [main] input.FileInputFormat: Total input paths to
process : 1
2018-08-06 15:17:56,426 INFO  [main] mapreduce.JobSubmitter: number of splits:1
2018-08-06 15:17:56,441 INFO  [main] Configuration.deprecation: io.bytes.per.checksum
is deprecated. Instead, use df
2018-08-06 15:17:56,580 INFO  [main] mapreduce.JobSubmitter: Submitting tokens for
job: job_1533538524805_0003
2018-08-06 15:17:56,834 INFO  [main] impl.YarnClientImpl: Submitted application
application_1533538524805_0003
2018-08-06 15:17:56,927 INFO  [main] mapreduce.Job: The url to track the job:
http://localhost:8088/proxy/applicatio
2018-08-06 15:17:56,928 INFO  [main] mapreduce.Job: Running job:
job_1533538524805_0003
2018-08-06 15:18:05,115 INFO  [main] mapreduce.Job: Job job_1533538524805_0003 running
in uber mode : false
2018-08-06 15:18:05,117 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2018-08-06 15:18:11,234 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2018-08-06 15:18:11,246 INFO  [main] mapreduce.Job: Job job_1533538524805_0003
completed successfully
2018-08-06 15:18:11,360 INFO  [main] mapreduce.Job: Counters: 31
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=139423
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=249
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=2
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3776
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=3776
                Total vcore-seconds taken by all map tasks=3776
                Total megabyte-seconds taken by all map tasks=3866624
        Map-Reduce Framework
                Map input records=6
                Map output records=5
                Input split bytes=104
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=98
                CPU time spent (ms)=1540
                Physical memory (bytes) snapshot=177299456
                Virtual memory (bytes) snapshot=2098319360
                Total committed heap usage (bytes)=120061952
        ImportTsv
                Bad Lines=1
        File Input Format Counters
                Bytes Read=145
        File Output Format Counters
                Bytes Written=0
[acadgild.mmisra ~]$
[acadgild.mmisra ~]$
```

### Step#3 – We launch  hbase shell again and scan the ‘workers’ table and we should see records successfully imported in hbase

```
[acadgild.mmisra ~]$
[acadgild.mmisra ~]$ hbase shell
2018-08-06 15:18:25,187 WARN  [main] util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-
log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> scan 'workers'
ROW                            COLUMN+CELL
 100                           column=details:name, timestamp=1533548872174,
value=Mohit
 100                           column=details:salary, timestamp=1533548872174,
value=20000
 100                           column=location:location, timestamp=1533548872174,
value=Lucknow
 200                           column=details:name, timestamp=1533548872174,
value=Rajan
 200                           column=details:salary, timestamp=1533548872174,
value=30000
 200                           column=location:location, timestamp=1533548872174,
value=Bombay
 301                           column=details:name, timestamp=1533548872174,
value=Priya
 301                           column=details:salary, timestamp=1533548872174,
value=24000
 301                           column=location:location, timestamp=1533548872174,
value=Jaipur
 807                           column=details:name, timestamp=1533548872174,
value=Suman
 807                           column=details:salary, timestamp=1533548872174,
value=34000
 807                           column=location:location, timestamp=1533548872174,
value=Agara
 899                           column=details:name, timestamp=1533548872174,
value=Sujit
 899                           column=details:salary, timestamp=1533548872174,
value=29000
 899                           column=location:location, timestamp=1533548872174,
value=Mathura
5 row(s) in 0.3470 seconds

hbase(main):002:0>
```