



Subject: Big Data Engineering (DJ19DSL604)

AY: 2022-23

Experiment 7

(No SQL Data Store)

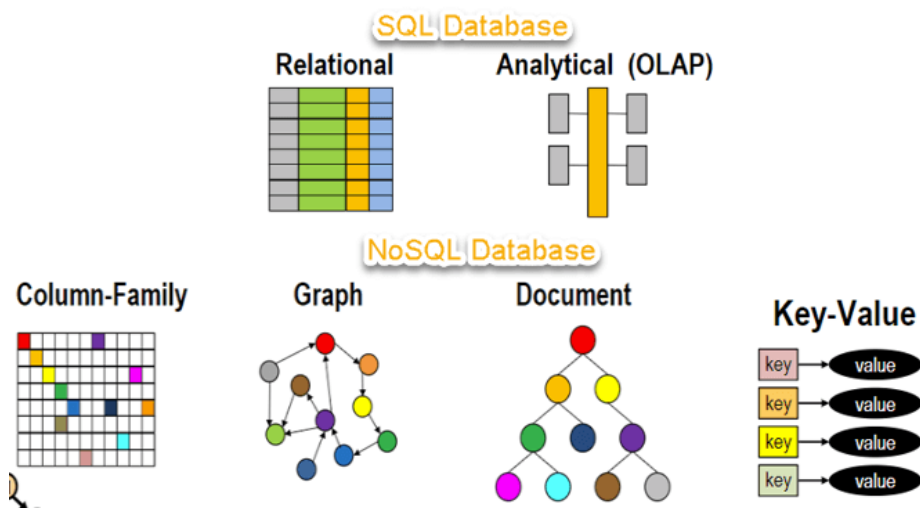
Name : Sarvagya Singh
SAPID : 60009200030
BATCH : K1

Aim: Implement No SQL Data Store using HBase.

Theory:

NoSQL:

NoSQL Database is a non-relational Data Management System, that does not require a fixed schema. It avoids joins, and is easy to scale. The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs. NoSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook and Google collect terabytes of user data every single day. **NoSQL database** stands for “Not Only SQL” or “Not SQL.”



Difference between SQL and NoSQL data stores:

- SQL databases are relational, and NoSQL databases are non-relational.
- SQL databases use structured query language (SQL) and have a predefined schema. NoSQL databases have dynamic schemas for unstructured data.
- SQL databases are vertically scalable, while NoSQL databases are horizontally scalable.



Department of Computer Science and Engineering (Data Science)

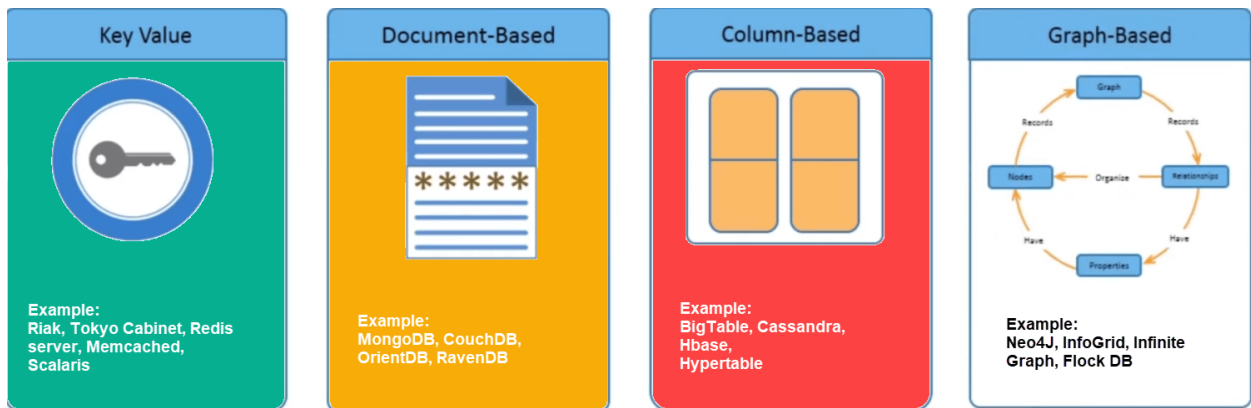
- SQL databases are table-based, while NoSQL databases are document, key-value, graph, or wide-column stores.
- SQL databases are better for multi-row transactions, while NoSQL is better for unstructured data like documents or JSON.

Types of NoSQL Databases

NoSQL Databases are mainly categorized into four types: Key-value pair, Column-oriented, Graph-based and Document-oriented. Every category has its unique attributes and limitations. None of the above-specified database is better to solve all the problems. Users should select the database based on their product needs.

Types of NoSQL Databases:

- Key-value Pair Based
- Column-oriented Graph
- Graphs based
- Document-oriented



Introduction to HBase

HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.

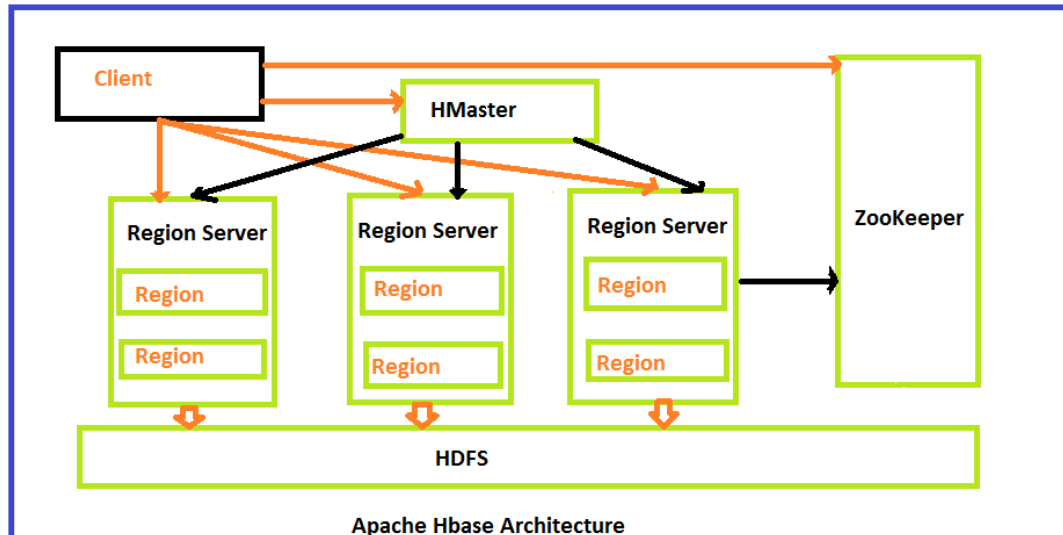
HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS).

It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.

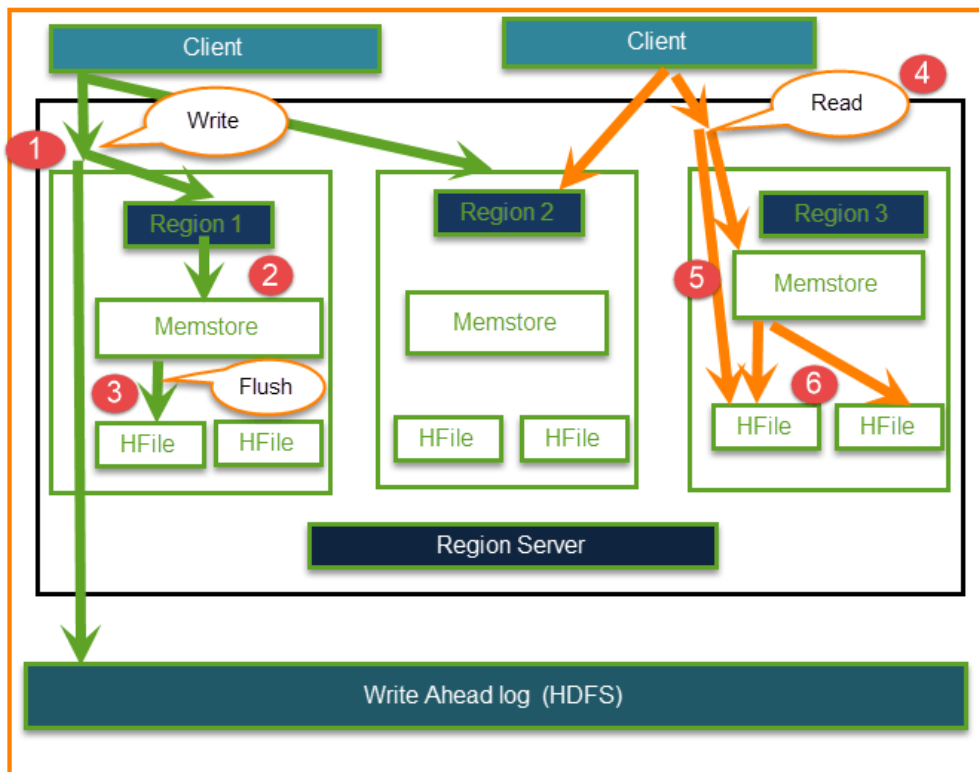
One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.



Department of Computer Science and Engineering (Data Science)
HBase Architecture



HBase Read and Write Data





Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Computer Science and Engineering (Data Science)

Lab Assignment:

1. Installation of HBase on standalone mode.
2. Implementation of HBase Create Table with Java API & Shell.
3. Implement HBase Shell Commands and dynamic scaling:
 - a. General commands
 - b. Tables Managements commands
 - c. Data manipulation commands
 - d. Cluster Replication Commands

```
hadoop@csedsstudent-cv115:/home/cseds-student/hadoop/hbase-2.4.17-bin/hbase-2.4.17/bin$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/cseds-student/hadoop/hbase-2.4.17-bin/hbase-2.4.17/lib/client-facing-thirdparty/slf4j-reload4j-1.7.33.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.Reload4jLoggerFactory]
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.hbase.unsafe.HBasePlatformDependent (file:/home/cseds-student/hadoop/hbase-2.4.17-bin/hbase-2.4.17/lib/hbase-unsafe-4.1.4.jar) to method java.nio.Bits.unaligned()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.hbase.unsafe.HBasePlatformDependent
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.17, r7fd096f39b4284da9a71da3ce67c48d259ffa79a, Fri Mar 31 18:10:45 UTC 2023
Took 0.0008 seconds
```

```
hbase:001:0> create 'bde', 'k2'
Created table bde
Took 0.8622 seconds
=> Hbase::Table - bde
hbase:002:0> put 'bde', '1', 'k2:name', 'Bhavik'
Took 0.1001 seconds
hbase:003:0> put 'bde', '1', 'k2:marks', '15'
Took 0.0075 seconds
hbase:004:0> put 'bde', '2', 'k2:name', 'Ojas'
Took 0.0052 seconds
hbase:005:0> put 'bde', '2', 'k2:marks', '14'
Took 0.0054 seconds
hbase:006:0> put 'bde', '3', 'k2:name', 'OM'
Took 0.0031 seconds
hbase:007:0> put 'bde', '3', 'k2:marks', '14'
Took 0.0069 seconds
```

```
hbase:008:0> scan 'bde'
ROW COLUMN+CELL
1 column=k2:marks, timestamp=2023-04-10T11:58:22.693, value=15
1 column=k2:name, timestamp=2023-04-10T11:58:00.301, value=Bhavik
2 column=k2:marks, timestamp=2023-04-10T11:59:01.773, value=14
2 column=k2:name, timestamp=2023-04-10T11:58:51.688, value=Ojas
3 column=k2:marks, timestamp=2023-04-10T11:59:48.021, value=14
3 column=k2:name, timestamp=2023-04-10T11:59:24.996, value=OM
3 row(s)
Took 0.0396 seconds
hbase:009:0> █
```

```
hbase:009:0> get 'bde', '1'
COLUMN CELL
k2:marks timestamp=2023-04-10T11:58:22.693, value=15
k2:name timestamp=2023-04-10T11:58:00.301, value=Bhavik
1 row(s)
Took 0.0222 seconds
hbase:010:0> █
```

```
hbase:012:0> put 'bde', '3', 'k2:marks', '25'
Took 0.0149 seconds
```