**Subject: Big Data Engineering (DJ19DSL604)**

# AY: 2023-24

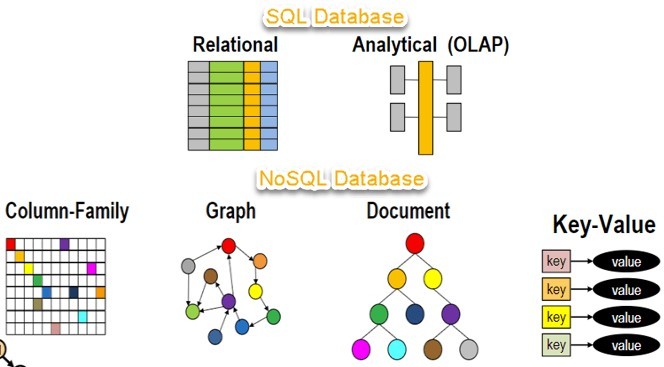
**Experiment 7 (No SQL Data Store)**

**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:

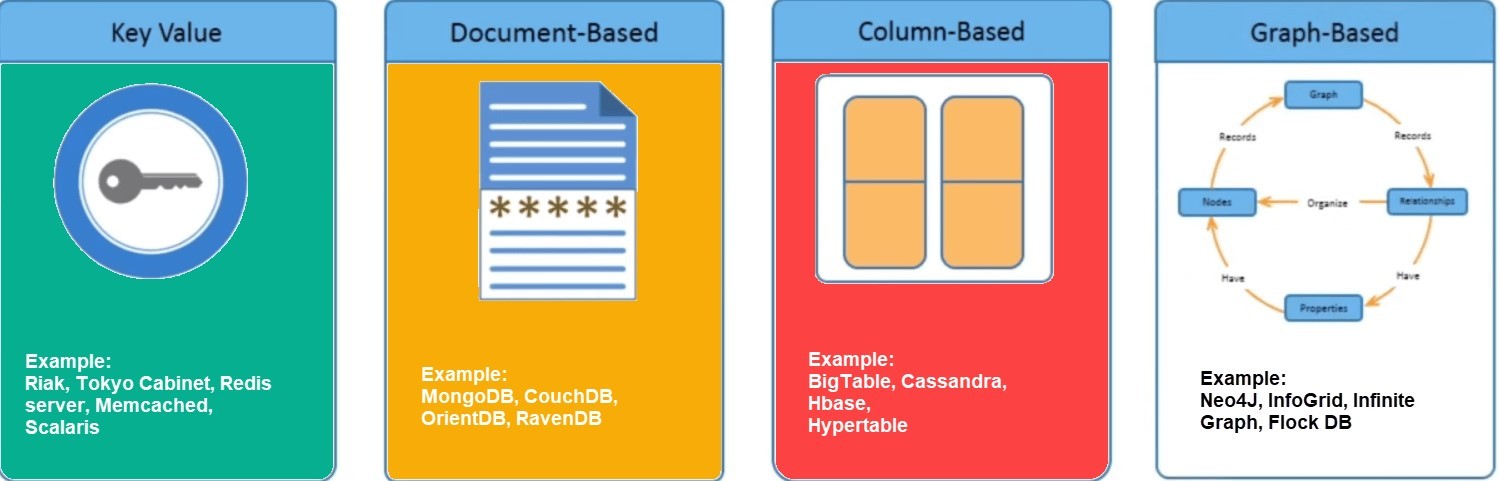
* SQL databases are relational, and NoSQL databases are non-relational.
* SQL databases use structured query language (SQL) and have a predefined schema. NoSQL databases have dynamic schemas for unstructured data.
* SQL databases are vertically scalable, while NoSQL databases are horizontally scalable.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

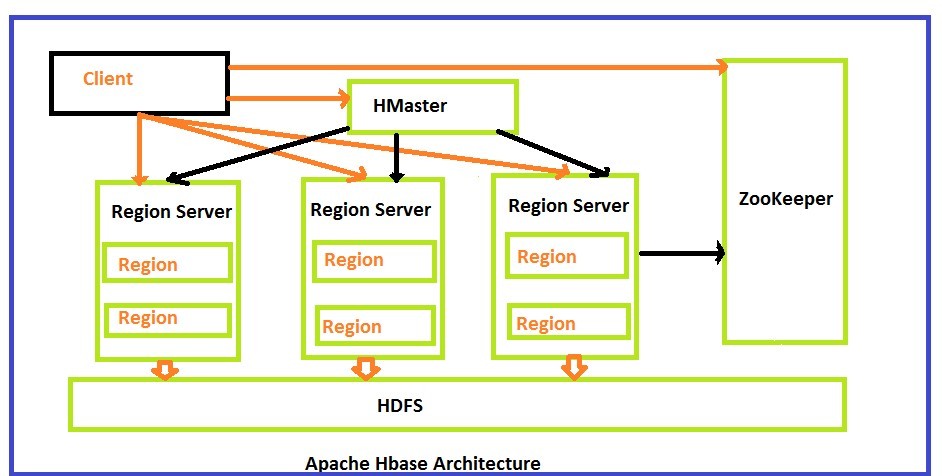
HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.

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).

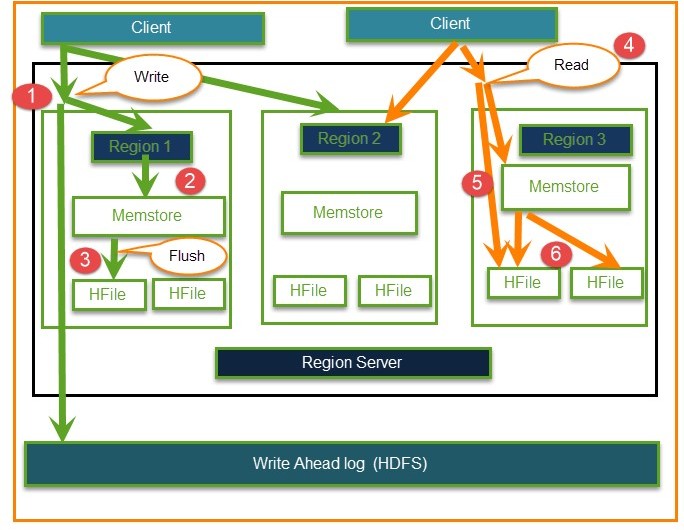
It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.

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.

# HBASE Architecture

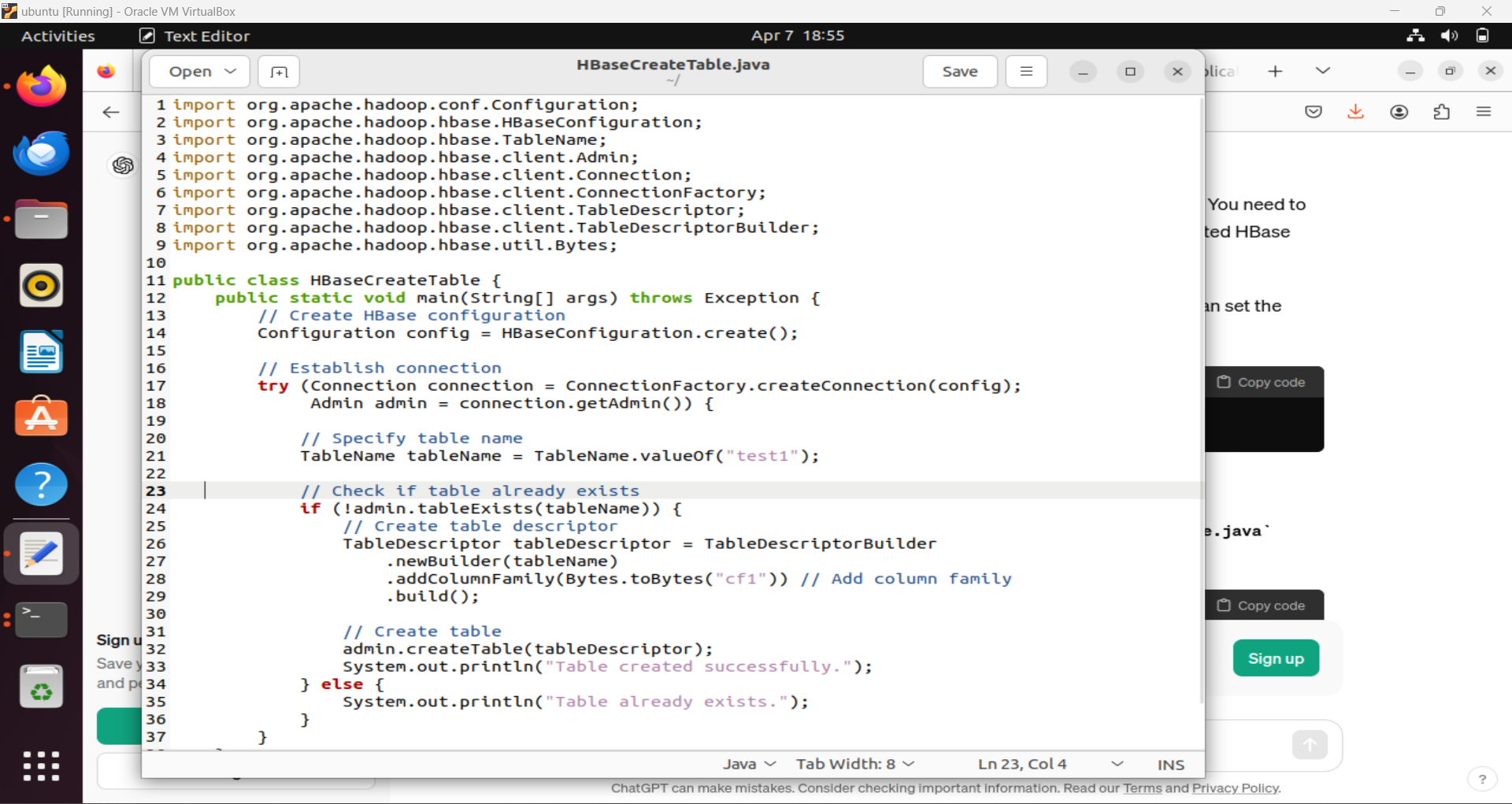


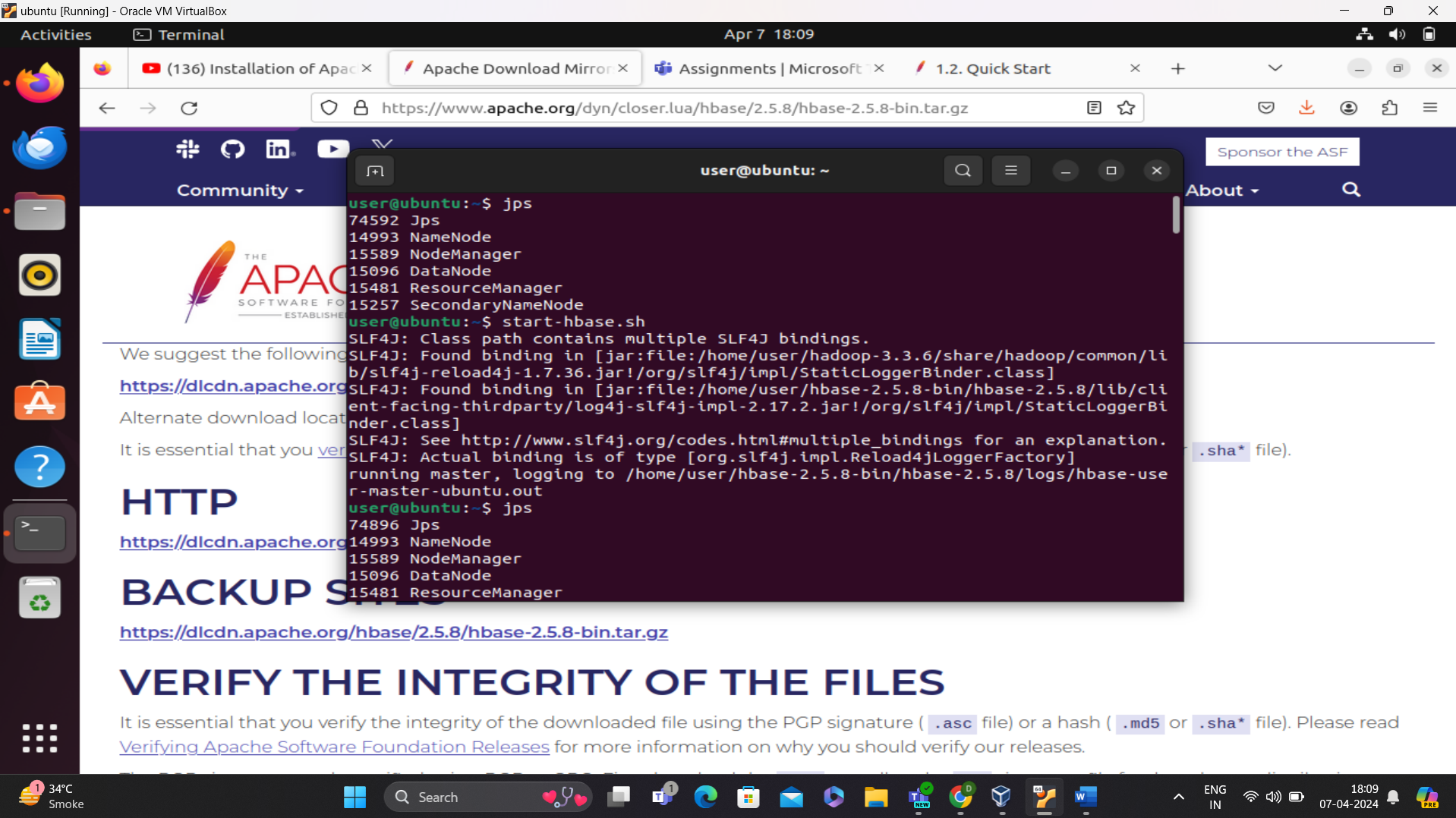
**HBase Read and Write Data**



**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:
   1. General commands
   2. Tables Managements commands
   3. Data manipulation commands
   4. Cluster Replication Commands





GENERAL COMMANDS:

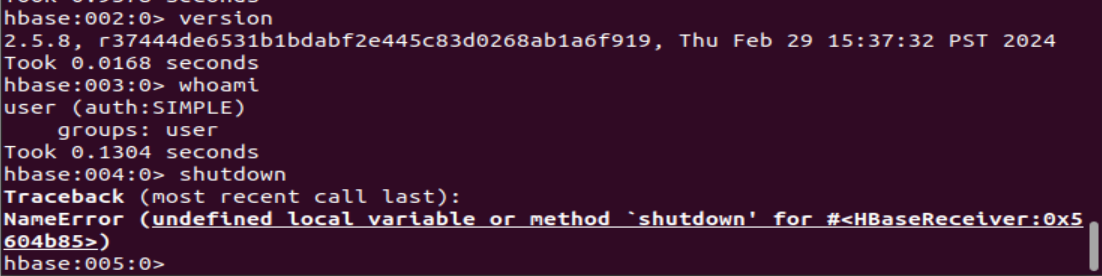
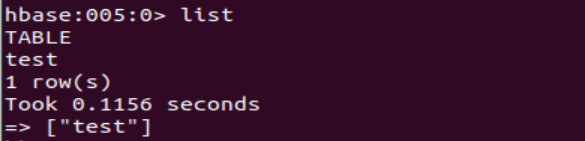
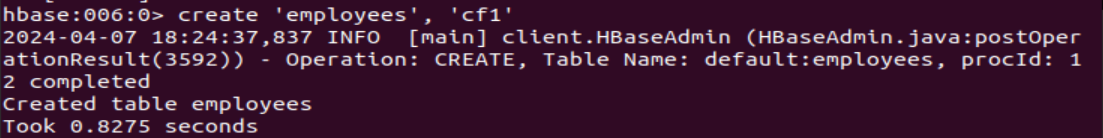
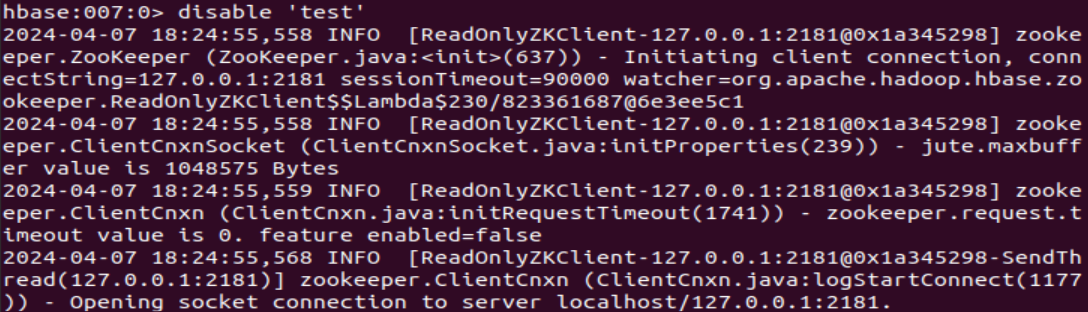
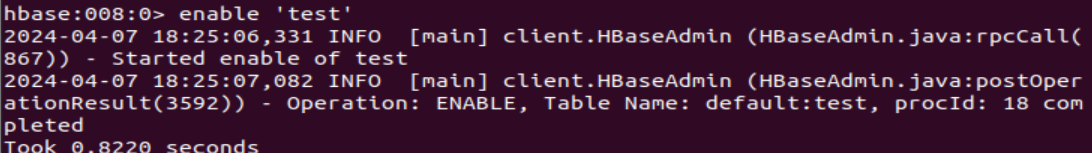


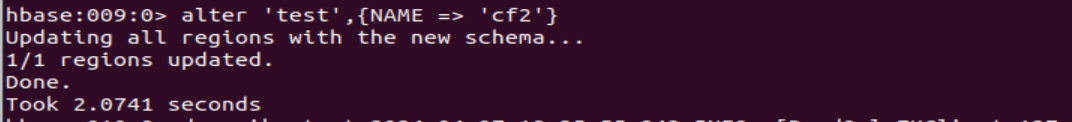
TABLE MANAGEMENT COMMANDS:

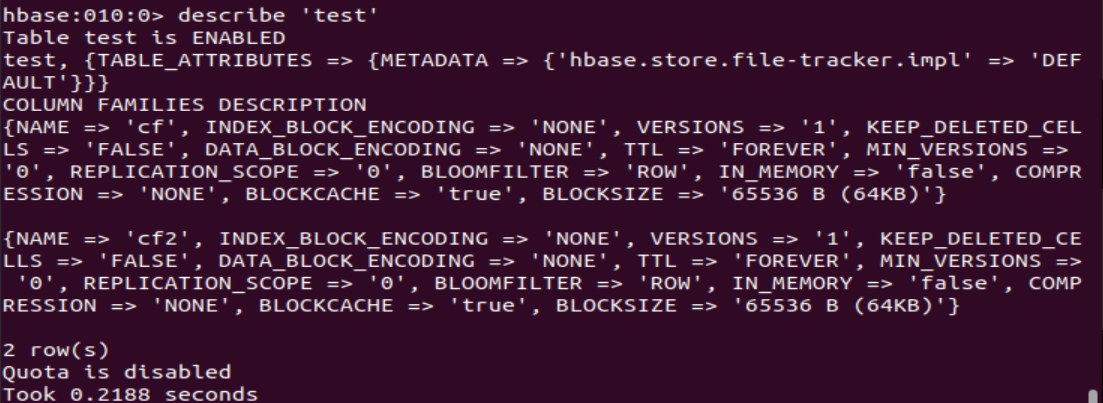


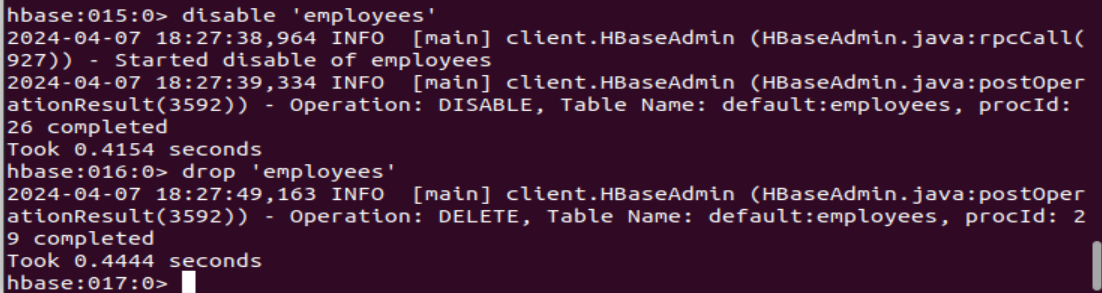




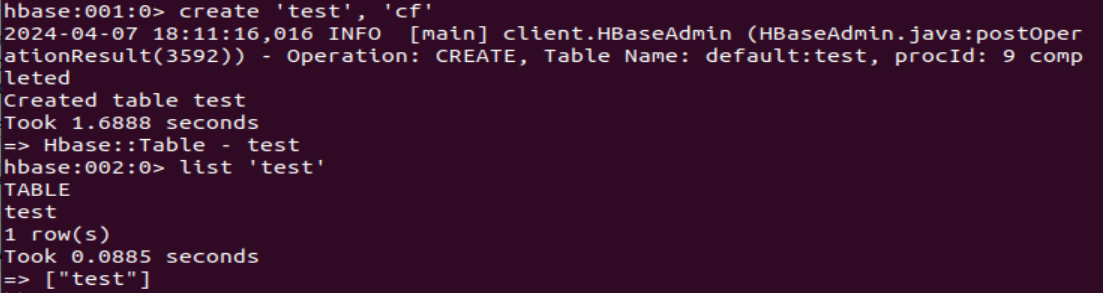


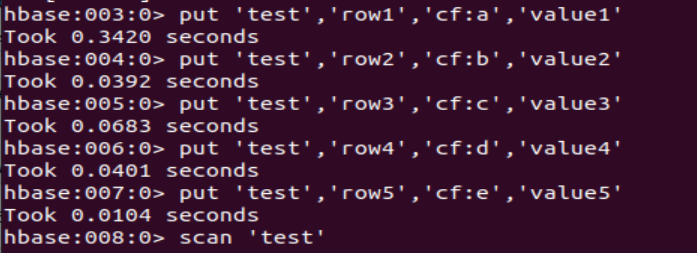


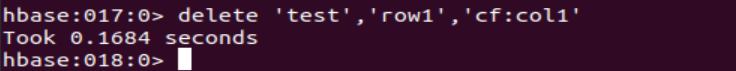




DATA MANUPULATION COMMANDS:







**Conclusion:**

In this experiment, we focused on implementing a NoSQL data store using HBase, a distributed column-oriented database built on top of the Hadoop ecosystem. This experiment equips users with the knowledge and skills necessary to harness the capabilities of HBase for building scalable and efficient NoSQL data storage solutions.