**Practical 7C : Partitioning the Table Using Hive**

**Partitioning the Table :**

Apache Hive is an open source data warehouse system used for querying and analyzing large datasets. Data in Apache Hive can be categorized into Table, Partition, and Bucket.The table in Hive is logically made up of the data being stored.

Hive provides way to categories data into smaller directories and files using partitioning or/and bucketing/clustering in order to improve performance of data retrieval queries and make them faster.

Main difference between Partitioning and Bucketing is that partitioning is applied directly on the column value and data is stored within directory named with column value whereas bucketing is applied using hash function on the column value MOD function with the number of buckets to store data in specific bucket file.

Hive table partition is a way to split a large table into smaller logical tables based on one or more partition keys. These smaller logical tables are not visible to users and users still access the data from just one table.

Partition eliminates creating smaller tables, accessing, and managing them separately.

To create a Hive table with partitions, you need to use PARTITIONED BY clause along with the column you wanted to partition and its type. Let’s create a table and Load the CSV file.

The data file that I am using to explain partitions can be downloaded from GitHub, It’s a simplified zipcodes codes where I have RecordNumber, Country, City, Zipcode, and State columns. I will be using State as a partition column.

## Load Data into Partition Table

Download the [zipcodes.CSV from GitHub](https://github.com/spark-examples/spark-scala-examples/blob/master/src/main/resources/zipcodes20.csv), upload it to HDFS, and finally load the CSV file into a partition table.



## Load Data into Partition Table

Download the [zipcodes.CSV from GitHub](https://github.com/spark-examples/spark-scala-examples/blob/master/src/main/resources/zipcodes20.csv), upload it to HDFS, and finally load the CSV file into a partition table.

## Show All Partitions on Hive Table

After loading the data into the Hive partition table, you can use SHOW PARTITIONS command to see all partitions that are present.





## Add New Partition to the Hive Table

A new partition can be added to the table using the ALERT TABLE statement, you can also specify the location where you wanted to store partition data on HDFS.



From the below image we can see that 6 partition have been created based on the name of the States.



**Bucketing the Table**

Hive Bucketing is a way to split the table into a managed number of clusters with or without partitions. With partitions, Hive divides(creates a directory) the table into smaller parts for every distinct value of a column whereas with bucketing you can specify the number of buckets to create at the time of [creating a Hive table](https://sparkbyexamples.com/apache-hive/hive-create-table-syntax-and-usage-with-examples/).

## Load Data into Bucket

Loading/inserting data into the Bucketing table would be the same as inserting data into the table.

****

Altering the table : Renaming the State name AL to ‘NY’

****

Now we can see from the below image ,the state name ‘AL’ is renamed to ‘NY’.

****