

Name:
Roll No:

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](#), upload it to HDFS, and finally load the CSV file into a partition table.

```
hive> CREATE TABLE zipcodes(  
  > RecordNumber int,  
  > Country string,  
  > City string,  
  > Zipcode int, State string)  
  > ROW FORMAT DELIMITED  
  > FIELDS TERMINATED BY ',' ;  
OK  
Time taken: 0.039 seconds
```

Name:
Roll No:

Load Data into Partition Table

Download the [zipcodes.CSV from GitHub](#), 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.

```
hive> load data local inpath '/home/cloudera/Documents/zipcode.csv' into table zipcodes;
Loading data to table default.zipcodes
Table default.zipcodes stats: [numFiles=1, totalSize=591]
OK
Time taken: 0.538 seconds
hive> select * from zipcodes;
OK
NULL      Country City      NULL      State
1          US      PARC PARQUE 704      PR
2          US      PASEO COSTA DEL SUR 704      PR
10         US      BDA SAN LUIS 709      PR
61391     US      CINGULAR WIRELESS 76166    TX
61392     US      FORT WORTH 76177    TX
61393     US      FT WORTH 76177    TX
4         US      URB EUGENE RICE 704      PR
39827     US      MESA 85209 AZ
39828     US      MESA 85210 AZ
49345     US      HILLIARD 32046    FL
49346     US      HOLDER 34445    FL
49347     US      HOLT 32564    FL
49348     US      HOMOSASSA 34487    FL
3         US      SECT LANAUSS 704      PR
54354     US      SPRING GARDEN 36275    AL
54355     US      SPRINGVILLE 35146    AL
54356     US      SPRUCE PINE 35585    AL
76511     US      ASH HILL 27007    NC
76512     US      ASHEBORO 27203    NC
76513     US      ASHEBORO 27204    NC
NULL      NULL      NULL      NULL      NULL
Time taken: 0.345 seconds, Fetched: 22 row(s)
```

```
hive> create table zipcode(RecordNumber int, Country string, City string, Zipcode int) PARTITIONED BY(State string);
OK
Time taken: 0.053 seconds
hive> set hive.exec.dynamic.partition.mode=nonstrict;
```

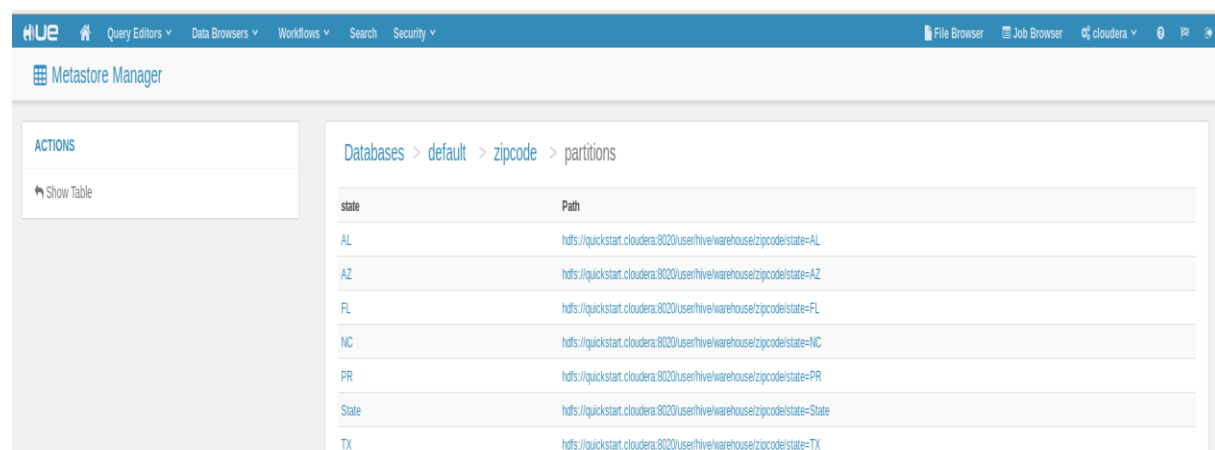
Add New Partition to the Hive Table

Name:
Roll No:

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

```
hive> insert overwrite table zipcode PARTITION(State) SELECT RecordNumber,Country,City,Zipcode,State from zipcodes;
Query ID = cloudera_20220322184444_4c8a901a-bbde-4aa1-8c04-26e6bc3e38aa
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1647952873179_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1647952873179_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1647952873179_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-03-22 18:44:27,035 Stage-1 map = 0%, reduce = 0%
2022-03-22 18:44:34,826 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.0 sec
MapReduce Total cumulative CPU time: 1 seconds 0 msec
Ended Job = job_1647952873179_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/.hive-staging_hive_2022-03-22_18-44-16_627_1195954405856007251-1/-ext-10000
Loading data to table default.zipcode partition (state=null)
Time taken for load dynamic partitions : 763
Loading partition {state= HIVE_DEFAULT_PARTITION_}
Loading partition {state=PR}
Loading partition {state=AZ}
Loading partition {state=FL}
Loading partition {state=State}
Loading partition {state=TX}
Loading partition {state=AL}
Loading partition {state=NC}
Time taken for adding to write entity : 9
Partition default.zipcode{state=AL} stats: [numFiles=1, numRows=3, totalSize=83, rawDataSize=80]
Partition default.zipcode{state=AZ} stats: [numFiles=1, numRows=2, totalSize=40, rawDataSize=38]
Partition default.zipcode{state=FL} stats: [numFiles=1, numRows=4, totalSize=91, rawDataSize=87]
Partition default.zipcode{state=NC} stats: [numFiles=1, numRows=3, totalSize=72, rawDataSize=69]
Partition default.zipcode{state=PR} stats: [numFiles=1, numRows=5, totalSize=121, rawDataSize=116]
Partition default.zipcode{state=State} stats: [numFiles=1, numRows=1, totalSize=19, rawDataSize=18]
Partition default.zipcode{state=TX} stats: [numFiles=1, numRows=3, totalSize=83, rawDataSize=80]
Partition default.zipcode{state= HIVE_DEFAULT_PARTITION_} stats: [numFiles=1, numRows=1, totalSize=12, rawDataSize=11]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 1.0 sec HDFS Read: 4423 HDFS Write: 930 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 0 msec
OK
Time taken: 22.12 seconds
```

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



state	Path
AL	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/state=AL
AZ	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/state=AZ
FL	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/state=FL
NC	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/state=NC
PR	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/state=PR
State	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/state=State
TX	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcode/state=TX

Name:
Roll No:

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](#).

Load Data into Bucket

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

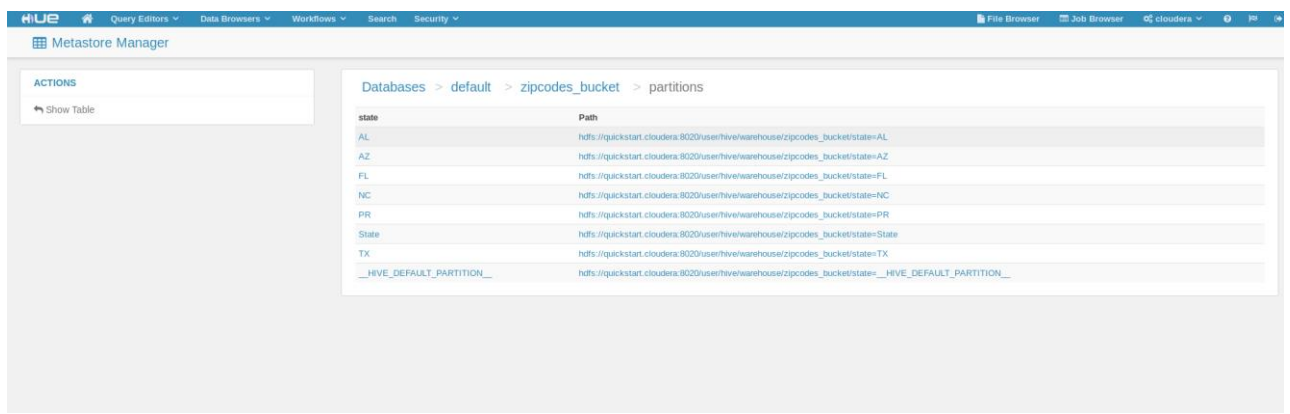
```
MapReduce Total cumulative CPU time: 35 seconds 950 msec
Ended Job = job_1646966376578_0003
Loading data to table default.zipcodes_bucket partition (state=null)
  Time taken for load dynamic partitions : 3203
  Loading partition {state= __HIVE_DEFAULT_PARTITION__}
  Loading partition {state=FL}
  Loading partition {state=PR}
  Loading partition {state=AZ}
  Loading partition {state=State}
  Loading partition {state=TX}
  Loading partition {state=NC}
  Loading partition {state=AL}
  Time taken for adding to write entity : 1
Partition default.zipcodes_bucket{state=AL} stats: [numFiles=32, numRows=3, totalSize=83, rawDataSize=80]
Partition default.zipcodes_bucket{state=AZ} stats: [numFiles=32, numRows=2, totalSize=40, rawDataSize=38]
Partition default.zipcodes_bucket{state=FL} stats: [numFiles=32, numRows=4, totalSize=91, rawDataSize=87]
Partition default.zipcodes_bucket{state=NC} stats: [numFiles=32, numRows=3, totalSize=72, rawDataSize=69]
Partition default.zipcodes_bucket{state=PR} stats: [numFiles=32, numRows=5, totalSize=121, rawDataSize=116]
Partition default.zipcodes_bucket{state=State} stats: [numFiles=32, numRows=1, totalSize=19, rawDataSize=18]
Partition default.zipcodes_bucket{state=TX} stats: [numFiles=32, numRows=3, totalSize=83, rawDataSize=80]
Partition default.zipcodes_bucket{state=__HIVE_DEFAULT_PARTITION__} stats: [numFiles=32, numRows=2, totalSize=24, rawDataSize=22]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 32 Cumulative CPU: 35.95 sec HDFS Read: 119079 HDFS Write: 2102 SUCCESS
Total MapReduce CPU Time Spent: 35 seconds 950 msec
OK
Time taken: 204.824 seconds
hive> █
```

Name:
Roll No:

Altering the table : Renaming the State name AL to 'NY'

```
hive> alter table zipcode partition(State='AL') rename to partition(State='NY');  
OK  
Time taken: 0.325 seconds  
hive> █
```

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



The screenshot shows the Hive Metastore Manager interface. On the left, there's a sidebar with 'Metastore Manager' and an 'ACTIONS' section containing a 'Show Table' button. The main area displays the 'Databases > default > zipcodes_bucket > partitions' view. It contains a table with two columns: 'state' and 'Path'.

state	Path
AL	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=AL
AZ	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=AZ
FL	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=FL
NC	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=NC
PR	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=PR
State	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=State
TX	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=TX
__HIVE_DEFAULT_PARTITION__	hdfs://quickstart.cloudera:8020/user/hive/warehouse/zipcodes_bucket/state=__HIVE_DEFAULT_PARTITION__