Hive

Create table for partitioning , bucketing & partitioning with bucketing for project

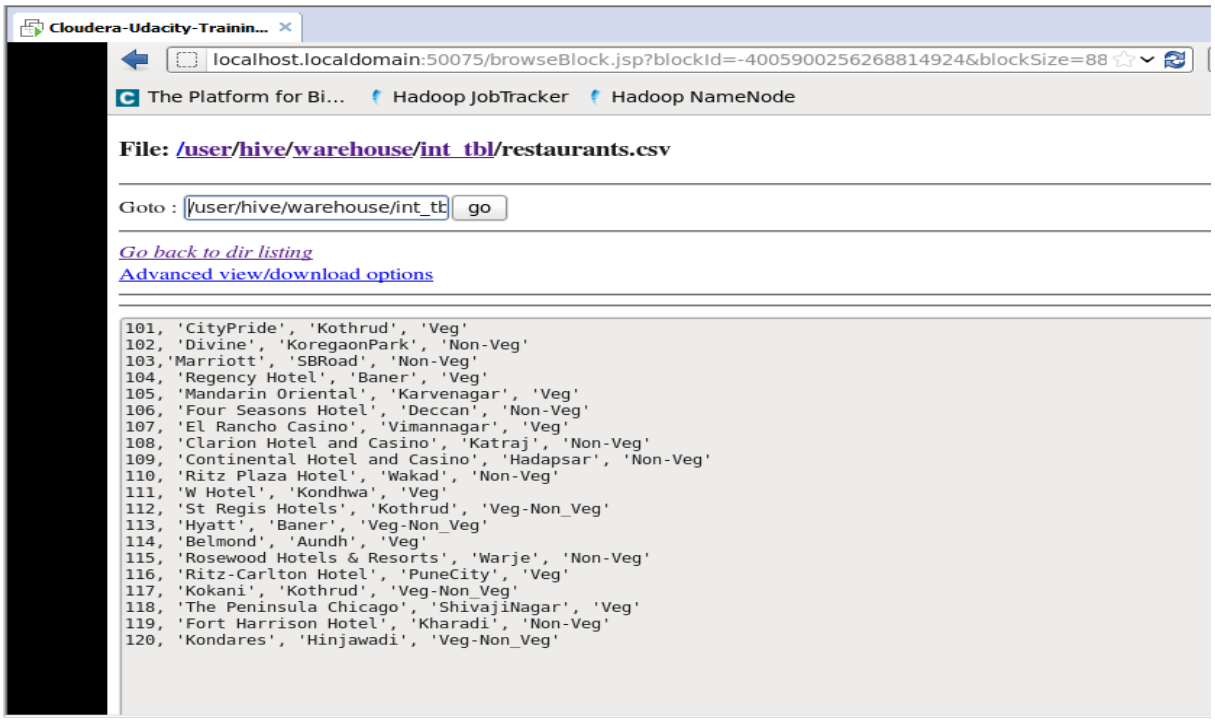
1.create table insert data in operating system

```
[root@localhost ~]# cat restaurants.csv
101, 'CityPride', 'Kothrud', 'Veg'
102, 'Divine', 'KoregaonPark', 'Non-Veg'
103,'Marriott', 'SBRoad', 'Non-Veg'
104, 'Regency Hotel', 'Baner', 'Veg'
105, 'Mandarin Oriental', 'Karvenagar', 'Veg'
106, 'Four Seasons Hotel', 'Deccan', 'Non-Veg'
107, 'El Rancho Casino', 'Vimannagar', 'Veg'
108, 'Clarion Hotel and Casino', 'Katraj', 'Non-Veg'
109, 'Continental Hotel and Casino', 'Hadapsar', 'Non-Veg'
110, 'Ritz Plaza Hotel', 'Wakad', 'Non-Veg'
111, 'W Hotel', 'Kondhwa', 'Veg'
112, 'St Regis Hotels', 'Kothrud', 'Veg-Non_Veg'
113, 'Hyatt', 'Baner', 'Veg-Non_Veg'
114, 'Belmond', 'Aundh', 'Veg'
115, 'Rosewood Hotels & Resorts', 'Warje', 'Non-Veg'
116, 'Ritz-Carlton Hotel', 'PuneCity', 'Veg'
117, 'Kokani', 'Kothrud', 'Veg-Non_Veg'
118, 'The Peninsula Chicago', 'ShivajiNagar', 'Veg'
119, 'Fort Harrison Hotel', 'Kharadi', 'Non-Veg'
120, 'Kondares', 'Hinjawadi', 'Veg-Non_Veg'
[root@localhost ~]#
```

Create internal data:-

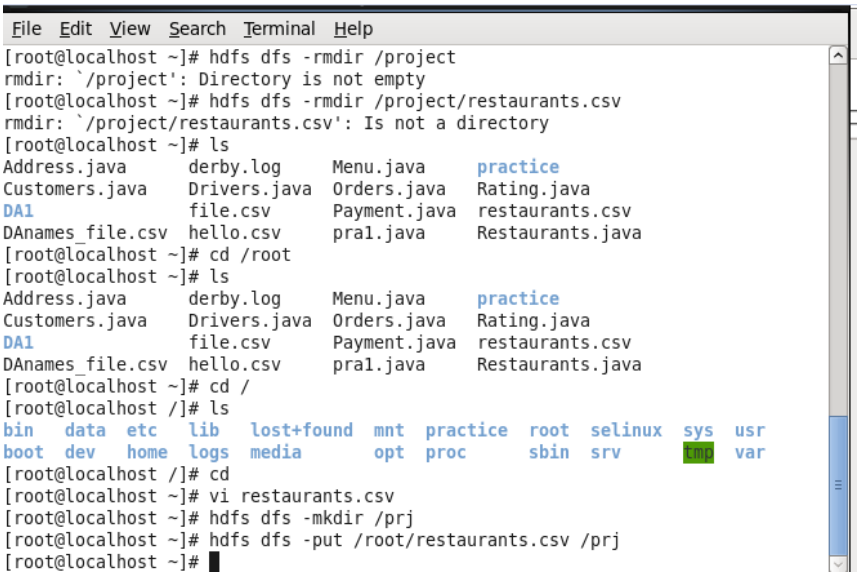
```
Cloudera-Udacity-Trainin...
hive> drop table parte_tbl;
OK
Time taken: 0.435 seconds
hive>  create table int_tbl(id int,rest_name string,city string,rest_type string)row format delimited fields terminated by ',
'lines terminated by '\n';
OK
Time taken: 0.071 seconds
hive> load data local inpath '/root/restaurants.csv'into table int_tbl;
Copying data from file:/root/restaurants.csv
Copying file: file:/root/restaurants.csv
Loading data to table default.int_tbl
OK
Time taken: 0.226 seconds
hive> select * from int_tbl;
OK
101     'CityPride'     'Kothrud'       'Veg'
102     'Divine'        'KoregaonPark'  'Non-Veg'
103    'Marriott'       'SBRoad'        'Non-Veg'
104     'Regency Hotel'         'Baner'         'Veg'
105     'Mandarin Oriental'     'Karvenagar'    'Veg'
106     'Four Seasons Hotel'    'Deccan'        'Non-Veg'
107     'El Rancho Casino'      'Vimannagar'    'Veg'
108     'Clarion Hotel and Casino'      'Katraj'        'Non-Veg'
109     'Continental Hotel and Casino'  'Hadapsar'      'Non-Veg'
110     'Ritz Plaza Hotel'      'Wakad'         'Non-Veg'
111     'W Hotel'       'Kondhwa'       'Veg'
112     'St Regis Hotels'       'Kothrud'       'Veg-Non_Veg'
113     'Hyatt'         'Baner'         'Veg-Non_Veg'
114     'Belmond'       'Aundh'         'Veg'
115     'Rosewood Hotels & Resorts'     'Warje'         'Non-Veg'
116     'Ritz-Carlton Hotel'    'PuneCity'      'Veg'
117     'Kokani'        'Kothrud'       'Veg-Non_Veg'
118     'The Peninsula Chicago'         'ShivajiNagar'  'Veg'
119     'Fort Harrison Hotel'   'Kharadi'       'Non-Veg'
120     'Kondares'      'Hinjawadi'     'Veg-Non_Veg'
Time taken: 0.23 seconds
```

Then create table on hive (internal table):- File Location



Create external table on HDFS

Create directory on hdfs on operating system



Data transfer from restaurants table to prj directory

For external table -File Location

Contents of directory /prj

Goto : /prj go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
restaurants.csv	file	0.86 KB	1	64 MB	2023-08-30 07:44	rw-r--r--	root	supergroup

Go back to DFS home

Cloudera-Udacity-Trainin...

The Platform for Bi... Hadoop JobTracker Hadoop NameNode

File: /prj/restaurants.csv

Goto : /prj go

Go back to dir listing
Advanced view/download options

```
101, 'CityPride', 'Kothrud', 'Veg'
102, 'Divine', 'KoregaonPark', 'Non-Veg'
103,'Marriott', 'SBRoad', 'Non-Veg'
104, 'Regency Hotel', 'Baner', 'Veg'
105, 'Mandarin Oriental', 'Karvenagar', 'Veg'
106, 'Four Seasons Hotel', 'Deccan', 'Non-Veg'
107, 'El Rancho Casino', 'Vimannagar', 'Veg'
108, 'Clarion Hotel and Casino', 'Katraj', 'Non-Veg'
109, 'Continental Hotel and Casino', 'Hadapsar', 'Non-Veg'
110, 'Ritz Plaza Hotel', 'Wakad', 'Non-Veg'
111, 'W Hotel', 'Kondhwa', 'Veg'
112, 'St Regis Hotels', 'Kothrud', 'Veg-Non_Veg'
113, 'Hyatt', 'Baner', 'Veg-Non_Veg'
114, 'Belmond', 'Aundh', 'Veg'
115, 'Rosewood Hotels & Resorts', 'Warje', 'Non-Veg'
116, 'Ritz-Carlton Hotel', 'PuneCity', 'Veg'
117, 'Kokani', 'Kothrud', 'Veg-Non_Veg'
118, 'The Peninsula Chicago', 'ShivajiNagar', 'Veg'
119, 'Fort Harrison Hotel', 'Kharadi', 'Non-Veg'
120, 'Kondares', 'Hinjawadi', 'Veg-Non_Veg'
```

Data partitioning

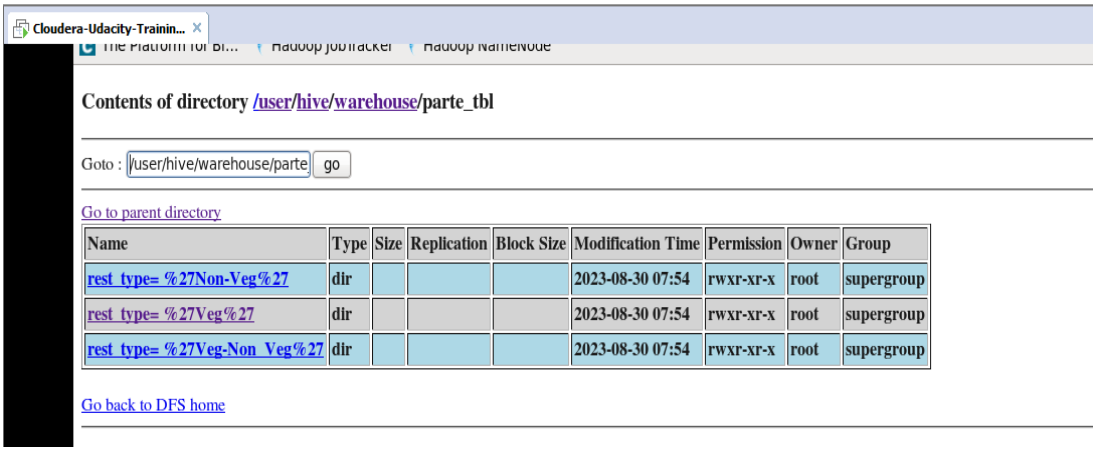
First we create normal table.

```
hive> create table npr_tbl(id int,rest_name string,city string,rest_type string)row format delimited fields terminated by ','
lines terminated by '\n';
OK
Time taken: 0.088 seconds
hive> select * from npr_tbl;
OK
Time taken: 0.183 seconds
hive> load data local inpath '/root/restaurants.csv'into table npr_tbl;
Copying data from file:/root/restaurants.csv
Copying file: file:/root/restaurants.csv
Loading data to table default.npr_tbl
OK
Time taken: 0.231 seconds
```

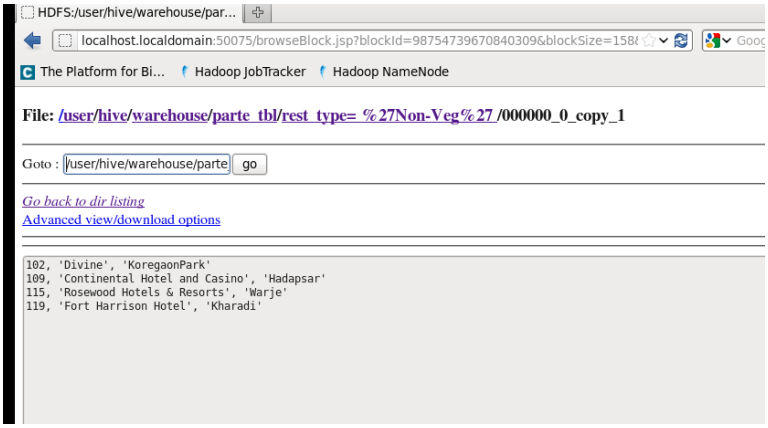
Then partitioning the table with help of normal table.

```
hive> create table parte_tbl(id int,rest_name string,city string)partitioned by(rest_type string)row format delimited fields
terminated by ','lines terminated by '\n';
OK
Time taken: 0.066 seconds
hive> insert into table parte_tbl partition(rest_type) select * from npr_tbl;
Total MapReduce jobs = 2
Launching Job 1 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1693333757154_0005, Tracking URL = http://localhost.localdomain:8088/proxy/application_1693333757154_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:9101 -kill job_1693333757154_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-08-30 07:54:16,900 Stage-1 map = 0%,  reduce = 0%
2023-08-30 07:54:24,843 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.07 sec
MapReduce Total cumulative CPU time: 2 seconds 70 msec
Ended Job = job_1693333757154_0005
Ended Job = 836649352, job is filtered out (removed at runtime).
Moving data to: hdfs://localhost:8020/tmp/hive-root/hive_2023-08-30_07-54-00_036_820156862947241l001/-ext-10000
Loading data to table default.parte_tbl partition (rest_type=null)
        Loading partition {rest_type= 'Veg' }
        Loading partition {rest_type= 'Veg-Non_Veg'}
        Loading partition {rest_type= 'Non-Veg' }
        Loading partition {rest_type= 'Veg'}
        Loading partition {rest_type= 'Non-Veg'}
20 Rows loaded to parte_tbl
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 2.07 sec   HDFS Read: 1079 HDFS Write: 672 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 70 msec
OK
Time taken: 26.147 seconds
hive>
```

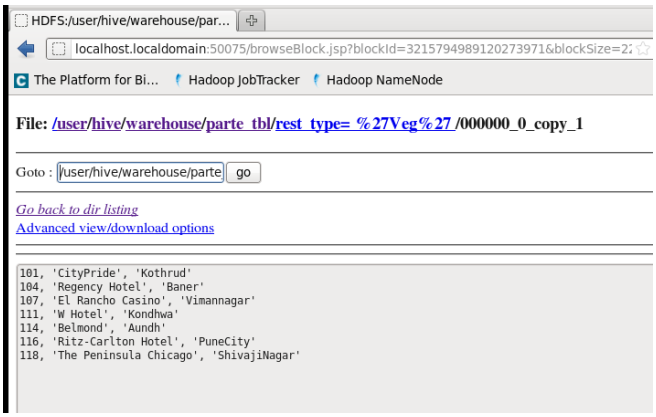
Partitioning file location:-



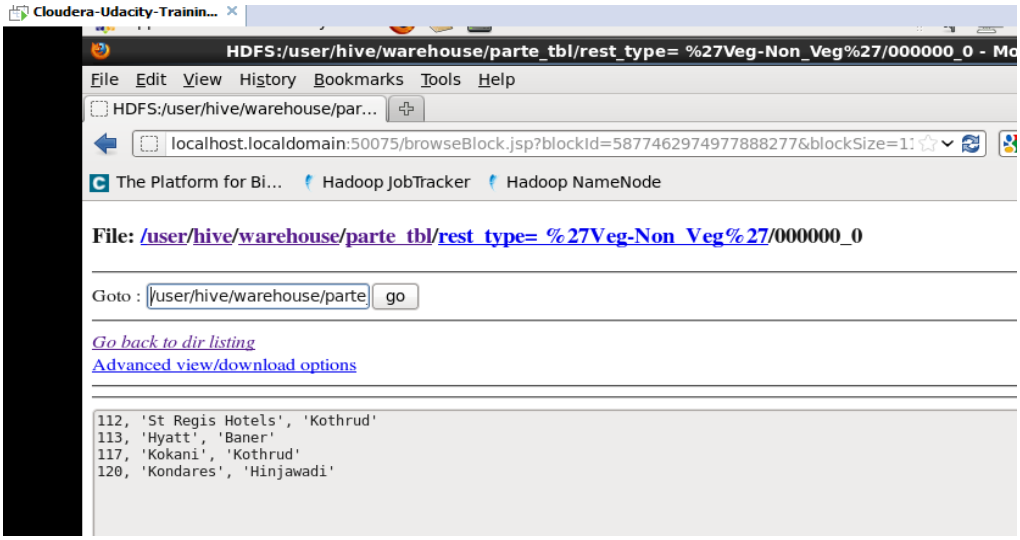
Nonveg hotels:-



Veg hotels:-



Veg & Non-veg hotels:-



Bucketing the data:-

Bucketing the data on customers data

1st we create data operating system

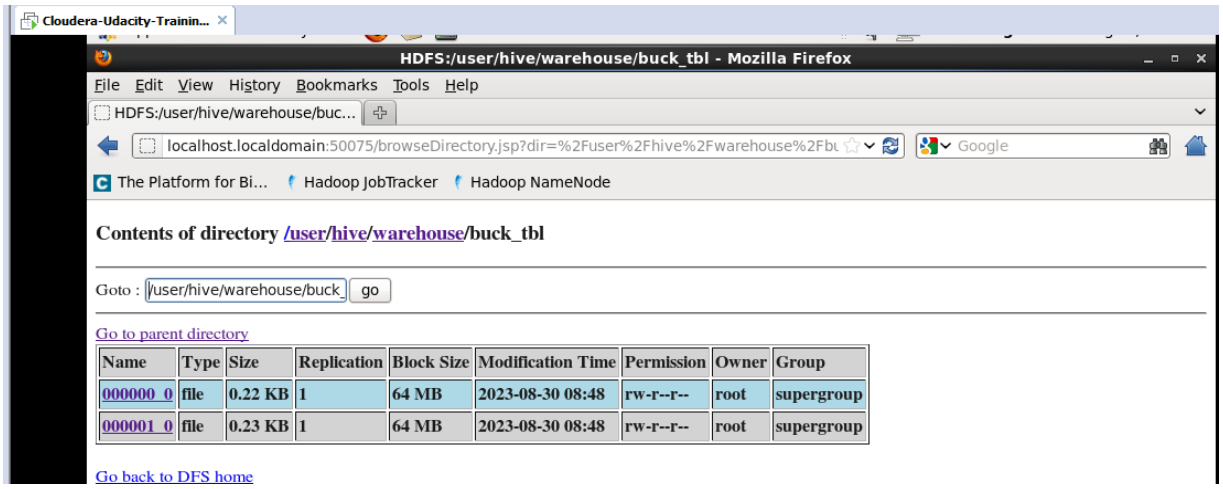
```
[root@localhost ~]# cat customers.csv
1, 'Aditya Gawade', 'adityagawade@gmail.com'
2, 'Ashish Kondare' , 'ashishkondare@gmail.com'
3, 'Viraj Walanj' , 'virajwalanj@gmail.com'
4, 'Akash Chavan' , 'akashchavan@gmail.com'
5, 'Sourabh Gawade', 'sourabhgawade@gmail.com'
6, 'Manasi Zagade', 'manasizagade@gmail.com'
7, 'Pooja Vaddepalli', 'poojawadepalli@gmail.com'
8, 'Ashwini Khade', 'ashwinikhade@gmail.com'
9, 'Ashwini Patil', 'ashwinipatil@gmail.com'
10, 'Jitesh Deore', 'jiteshdeore@gmail.com'
[root@localhost ~]#
```

Then we transfer the data from operating system to hive

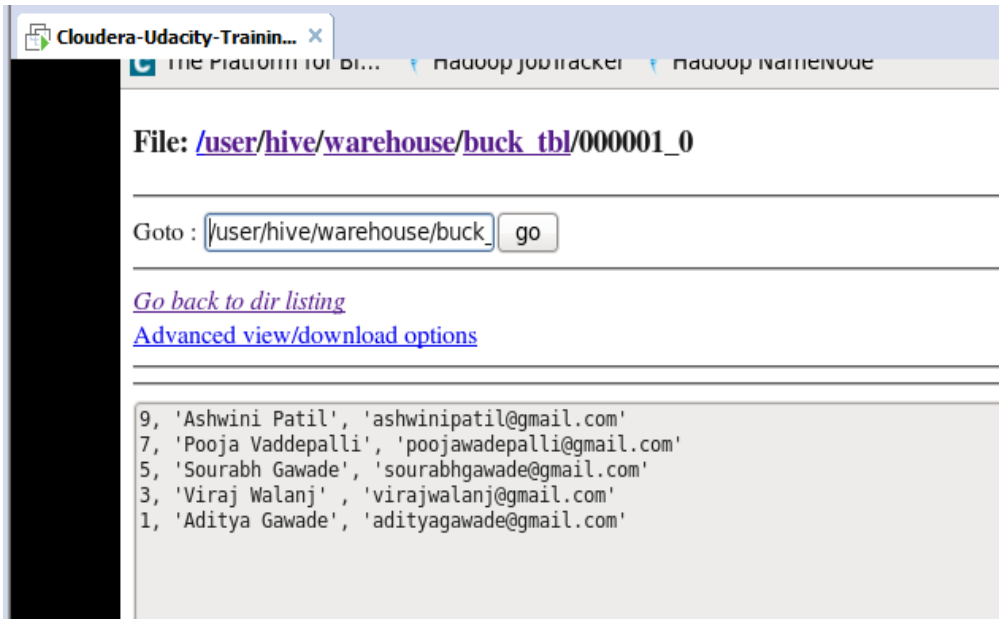
```
hive> create table npr_tbl1(id int,name string,email string)row format delimited fields terminated by ','lines terminated by
'\n';
OK
Time taken: 0.089 seconds
hive> load data local inpath '/root/customers.csv'into table npr_tbl1;
Copying data from file:/root/customers.csv
Copying file: file:/root/customers.csv
Loading data to table default.npr_tbl1
OK
Time taken: 0.244 seconds
```

```
Time taken: 22.538 seconds
hive> set hive.enforce.bucketing=true;
hive> drop table buck_tbl;
OK
Time taken: 0.157 seconds
hive> create table buck_tbl(id int,name string,email string)clustered by(id)into 2 buckets row format delimited fields termin
ated by ',';
OK
Time taken: 0.091 seconds
hive> insert into table buck_tbl select * from npr_tbl1;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 2
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_1693333757154_0009, Tracking URL = http://localhost.localdomain:8088/proxy/application_1693333757154_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:9101 -kill job_1693333757154_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 2
2023-08-30 08:47:54,028 Stage-1 map = 0%,  reduce = 0%
2023-08-30 08:48:22,320 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.97 sec
2023-08-30 08:48:23,744 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.79 sec
2023-08-30 08:48:25,356 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.79 sec
2023-08-30 08:48:26,960 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 7.71 sec
2023-08-30 08:48:28,645 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 7.71 sec
2023-08-30 08:48:30,004 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.39 sec
MapReduce Total cumulative CPU time: 9 seconds 390 msec
Ended Job = job_1693333757154_0009
Loading data to table default.buck_tbl
10 Rows loaded to buck_tbl
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 2   Cumulative CPU: 9.39 sec   HDFS Read: 654 HDFS Write: 457 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 390 msec
```

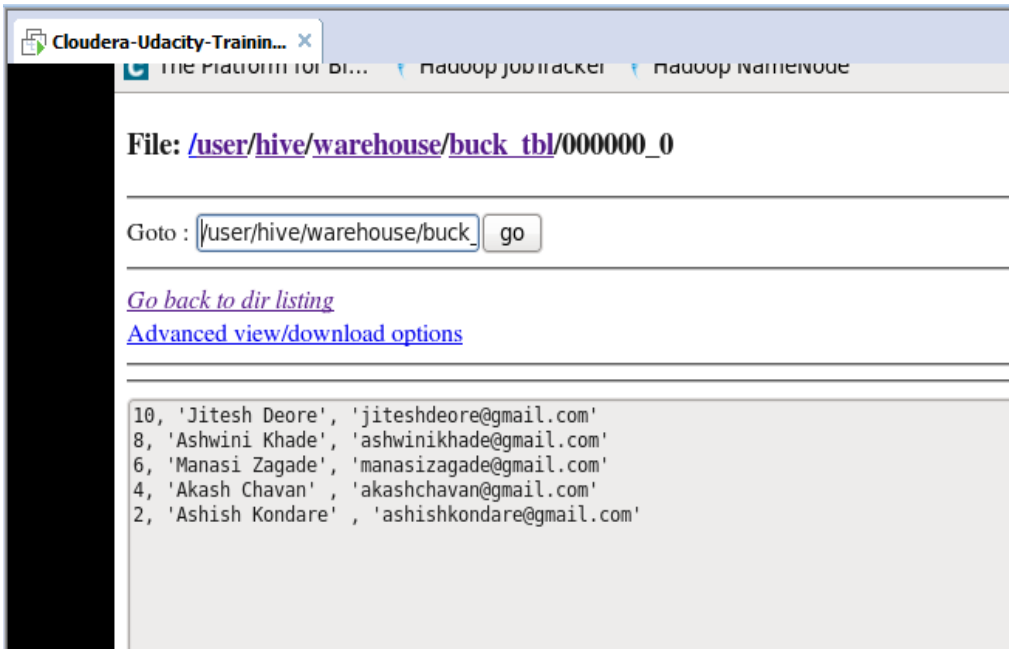
HDFS location:-



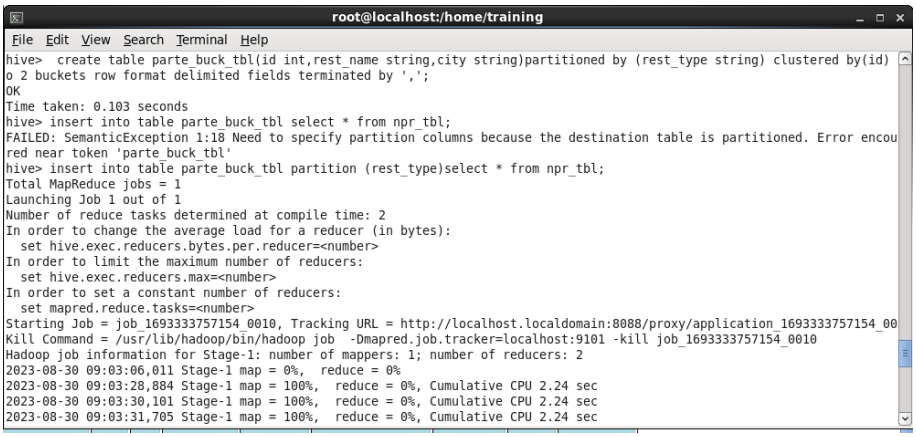
Bucket 1



Bucket 2:-

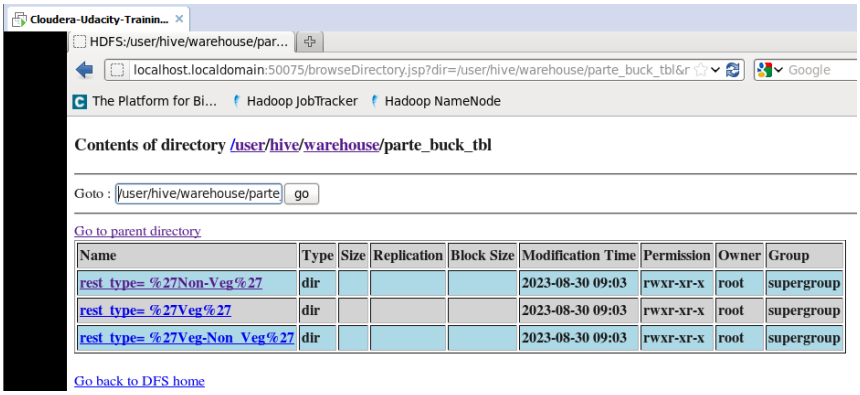


Then we do both at a time partitioning & bucketing



partitioning & bucketing location:- hive location

partitioning:-



Bucketing:-