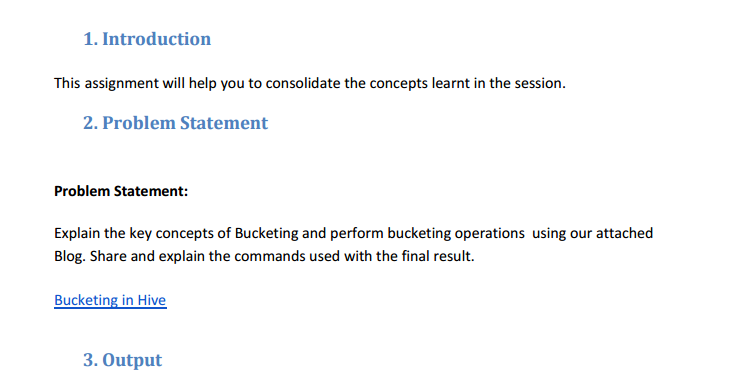
**Assignment 27.3**



**Concept of Bucketing in Hive** -

Bucketing gives a fine structure to Hive tables while performing queries on large datasets. As we all know, Partition helps in increasing the efficiency when performing a query on a table. Instead of scanning the whole table, it will only scan for the partitioned set and does not scan or operate on the unpartitioned sets, which helps us to provide results in lesser time and the details will be displayed very quickly because of Hive Partition.

**Bucketing** –

Bucketing is done when there is a huge [dataset](https://acadgild.com/big-data/big-data-development-training-certification). At times, even after partitioning on a particular field or fields, the partitioned file size doesn’t match with the actual expectation and remains huge and we want to manage the partition results into different parts. To overcome this problem of partitioning, Hive provides Bucketing concept, which allows user to divide table data sets into more manageable parts.

Thus, Bucketing helps user to maintain parts that are more manageable and user can set the size of the manageable parts or Buckets too.

**Bucketing Features in Hive-**

Hive partition divides table into number of partitions and these partitions can be further subdivided into more manageable parts known as Buckets or Clusters. The Bucketing concept is based on Hash function, which depends on the type of the bucketing column. Records which are bucketed by the same column will always be saved in the same bucket.

Here,**CLUSTERED BY** clause is used to divide the table into buckets.

* In [**Hive Partition**,](http://www.hadooptpoint.com/introduction-hive-partition-big-data/)each partition will be created as **directory**.
* In **Hive Buckets**, each bucket will be created as **file.**

Bucketing can also be done even without partitioning on Hive tables.

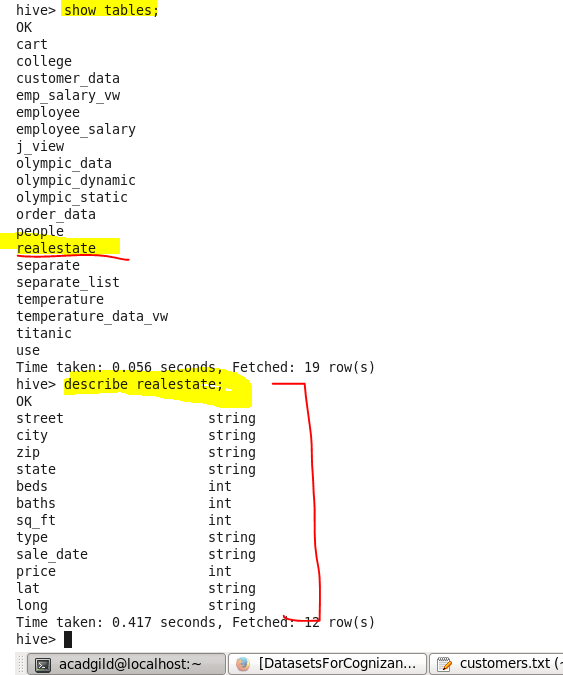
**Advantages of Bucketing:**

Bucketed tables allows much more efficient [sampling](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Sampling) than the non-bucketed tables. With sampling, we can try out queries on a section of data for testing and debugging purpose when the original data sets are very huge. Here, the user can fix the size of buckets according to the need.

Bucketing concept also provides the flexibility to keep the records in each bucket to be sorted by one or more columns. Since the data files are equal sized parts, map-side joins will be faster on the bucketed tables.

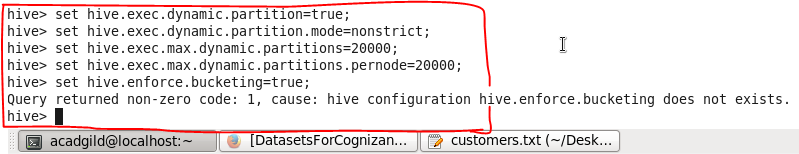
**Example :**  
> **We are going to use our realestate table we created earlier**

Showing tables and then, describing the table

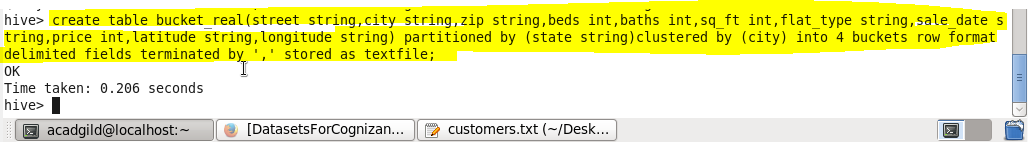


Before Proceeding Further for Bucketing Scripts:

Setting the required configurations:

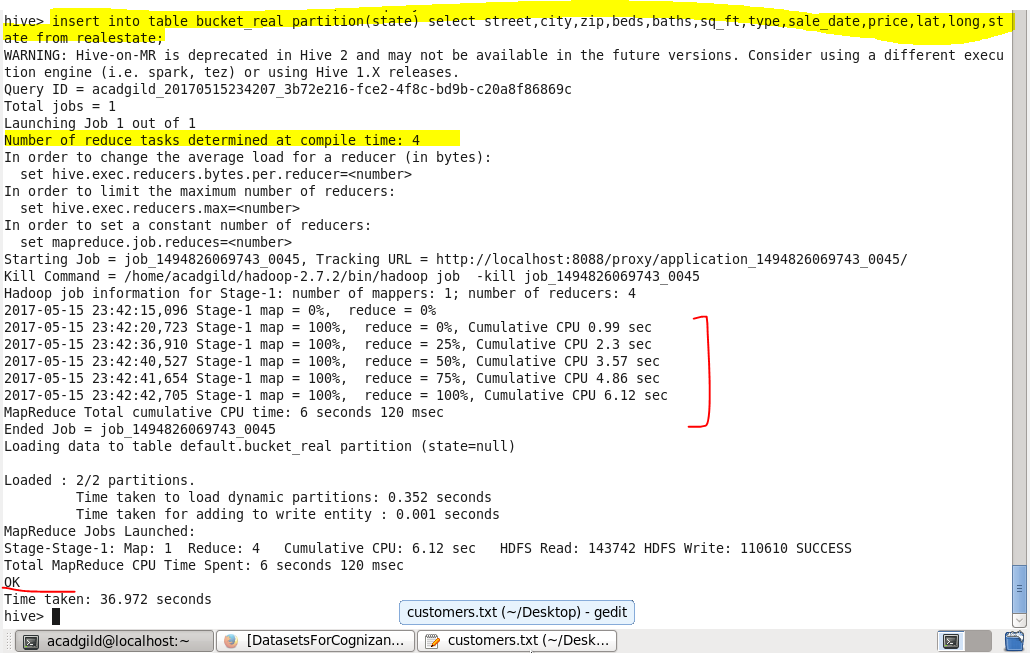


* **creating the bucket table:**

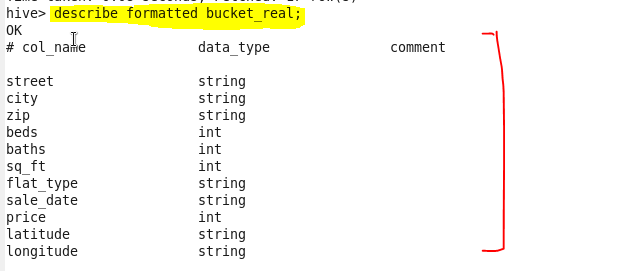


***CLUSTERED BY*** clause is used to divide the table into buckets.

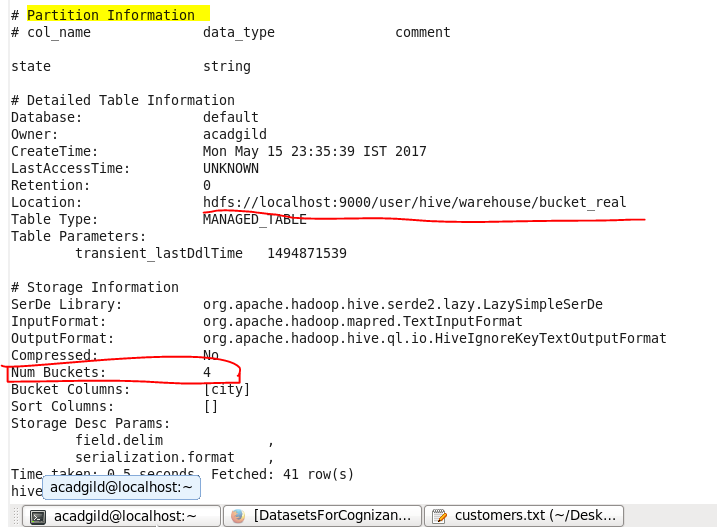
* **Inserting the data into bucketed table:**



**Describing** the bucketed table:

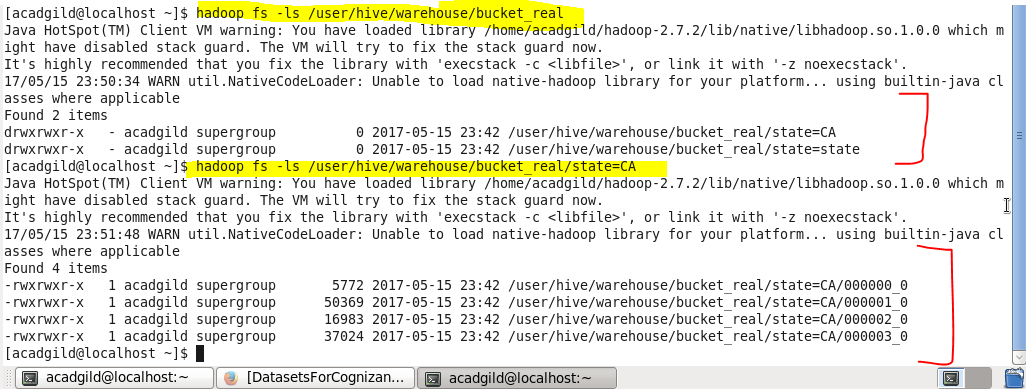


Here, we can see that number of buckets are 4.



We can see that buckets are created

**All four buckets in HDFS:**

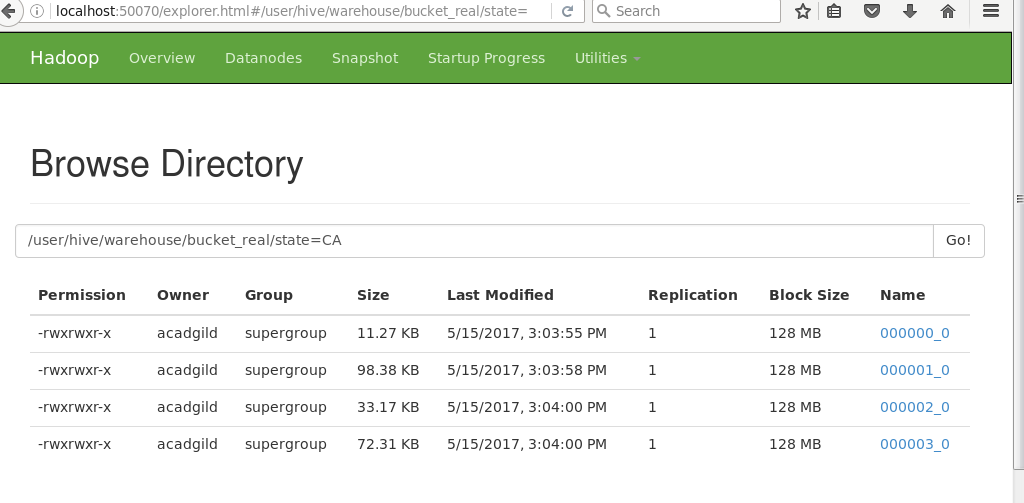


**Displaying the contents of it using cat command.**



Similarly we can check other three bucket files using cat command.

**In browser we can check our resultant files :**

****

**We can access the files by downloading**

**Conclusion:**

Bucketed tables allows much more efficient [sampling](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Sampling) than the non-bucketed tables. Bucketing concept allows user to divide table data sets into more manageable parts.