Explain the key concepts of Bucketing and perform bucketing operations using our attached Blog. Share and explain the commands used with the final result

**Solution:**

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.

If there is a huge [dataset](https://acadgild.com/big-data/big-data-development-training-certification) then 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.

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. But in Hive Buckets, each bucket will be created as file.

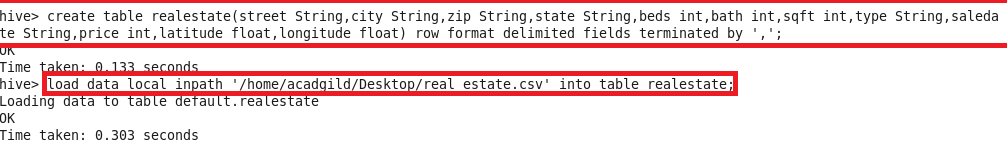
Bucketing can also be done even without partitioning on Hive tables.

**Advantages of Bucketing:**

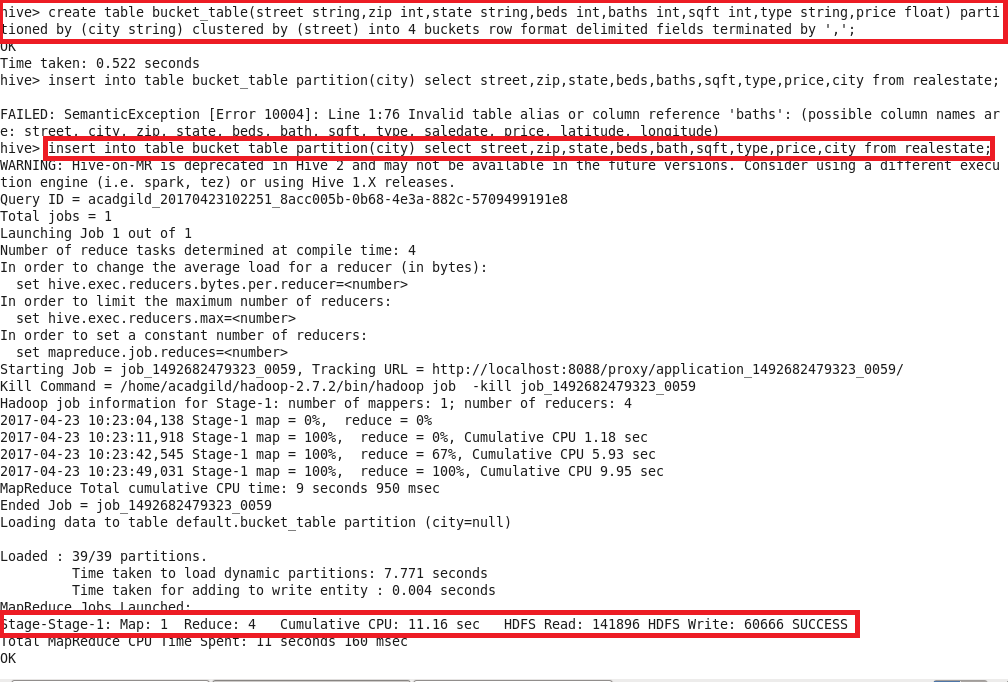
Bucketed tables allows much more efficient [sampling](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Sampling) than the non-bucketed tables. With sampling, we can try out queries on a section of data for testing and debugging purpose when the original data sets are very huge. Here, the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

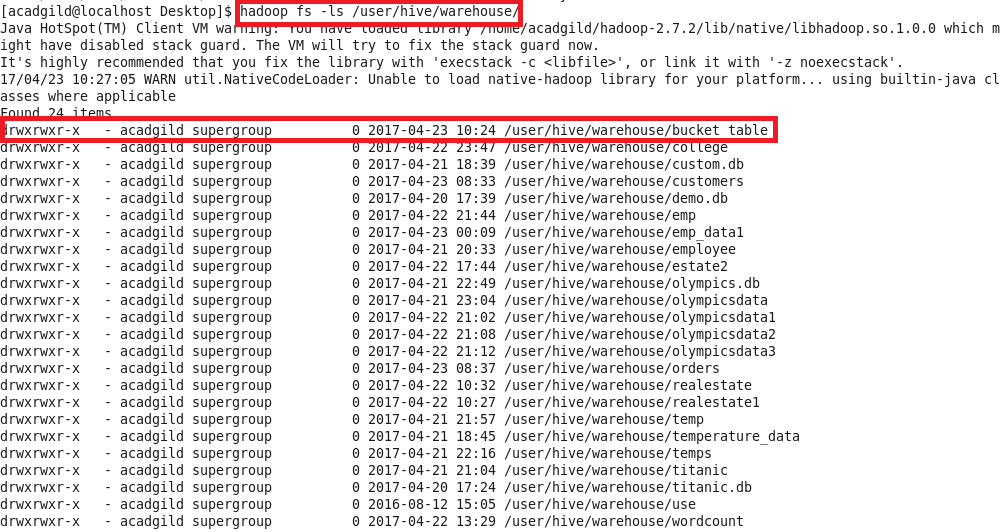
Creating table realestate and loading the data:

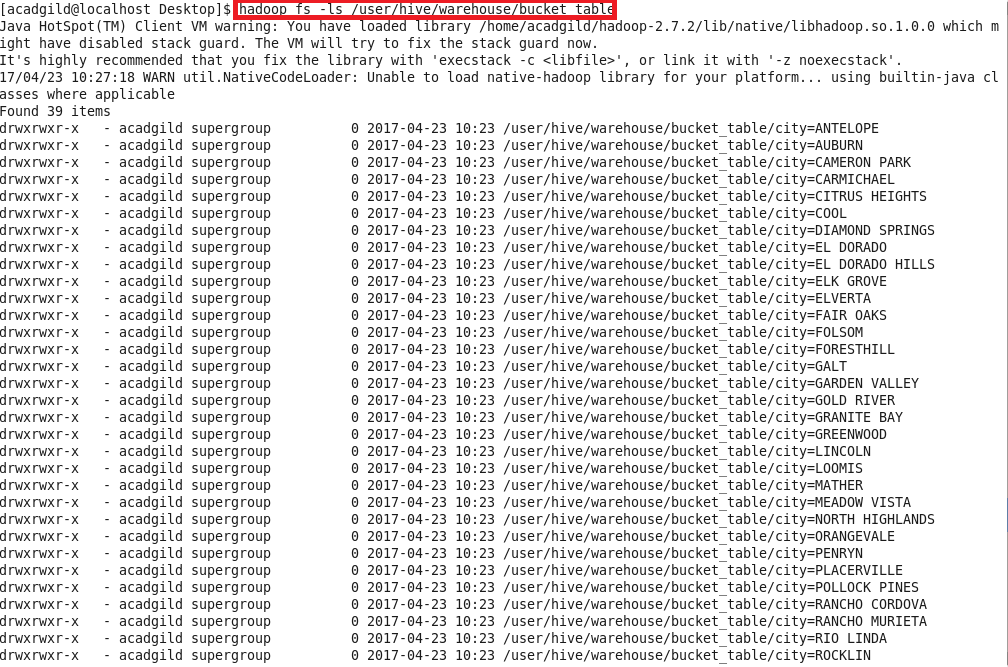


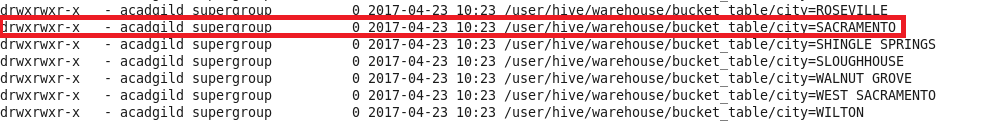
Creating bucket table and inserting the data:



Browsing directory:







4 buckets are created and displaying the contents of the file:

