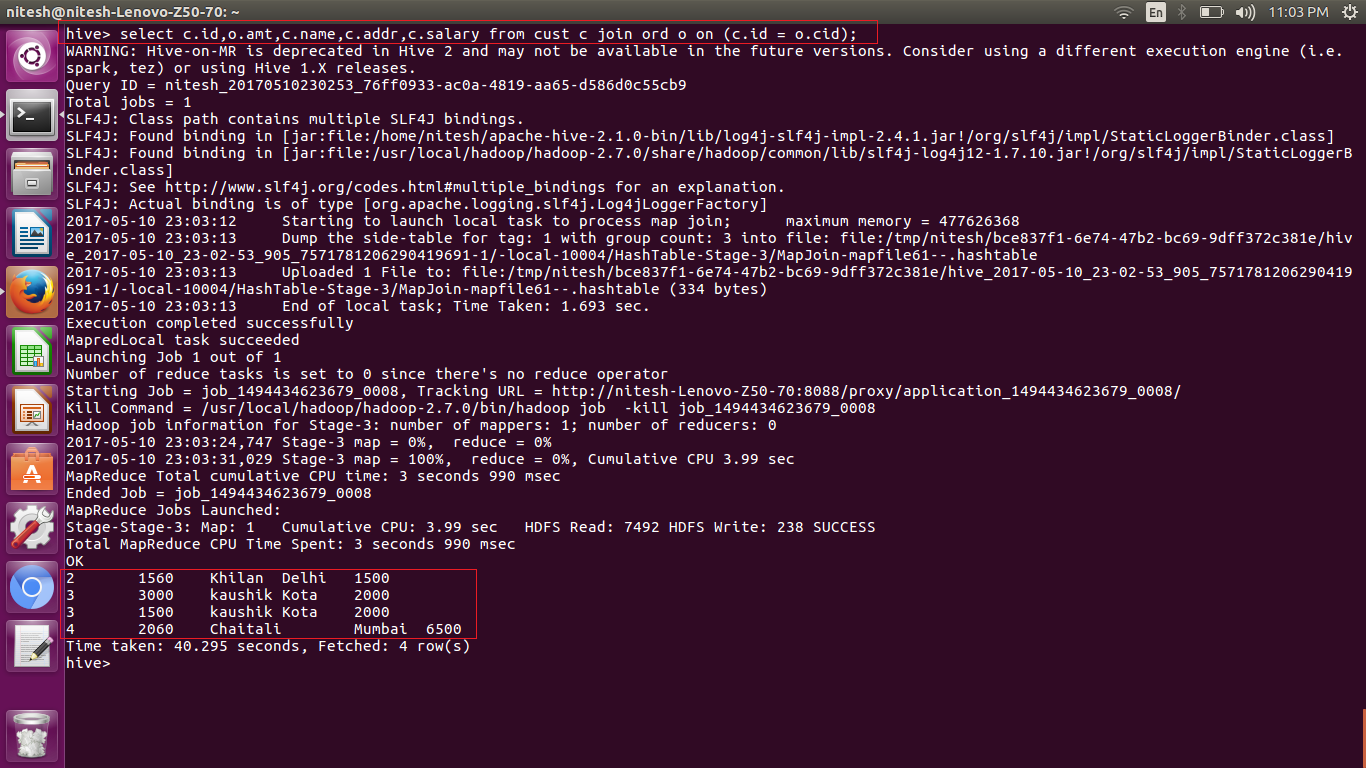
Perform join optimizations in hive by following the steps in the below blog link

Join table ordering (Largest table last).

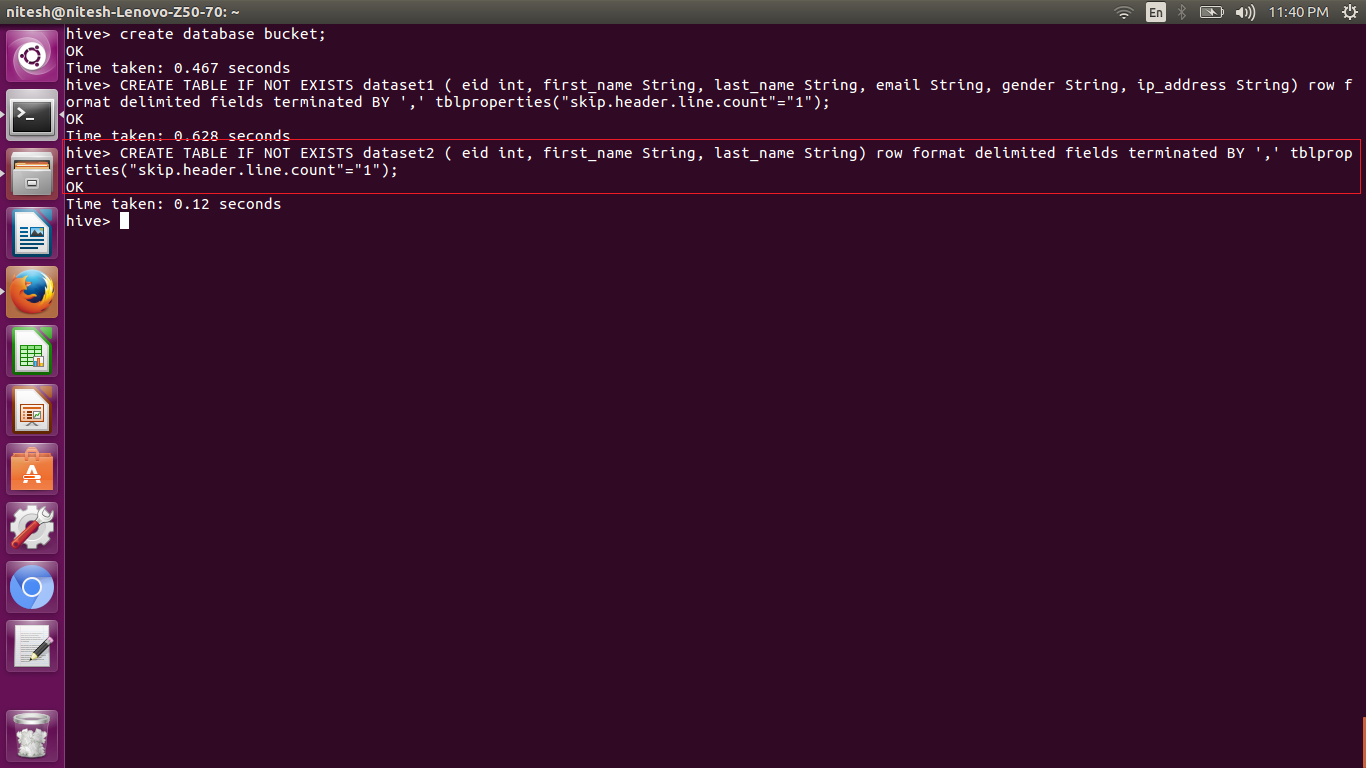
As with any type of tuning, it is important to understand the internal working of a system. When Hive executes a join, it needs to select which table is streamed and which table is cached. Hive takes the last table in the JOIN statement for streaming, so we need to ensure that this streaming table is largest among the two.

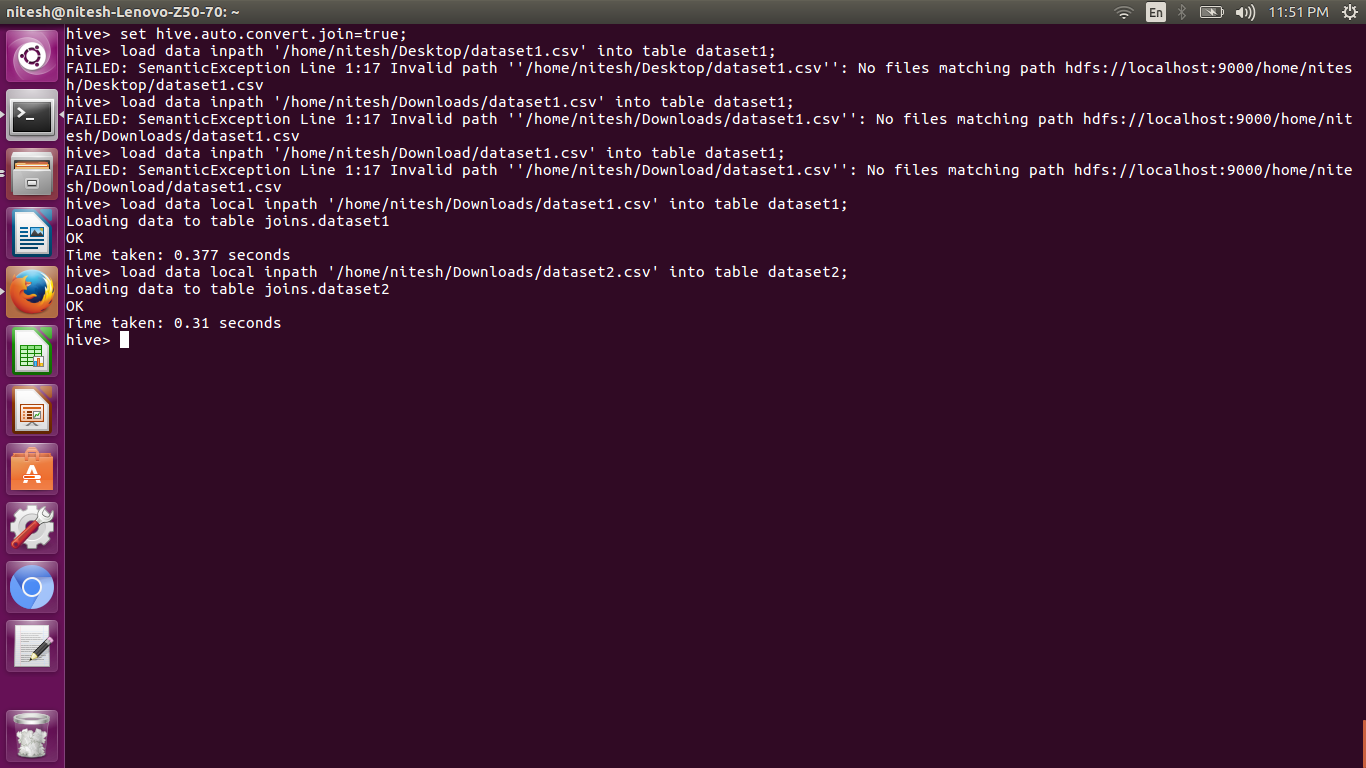


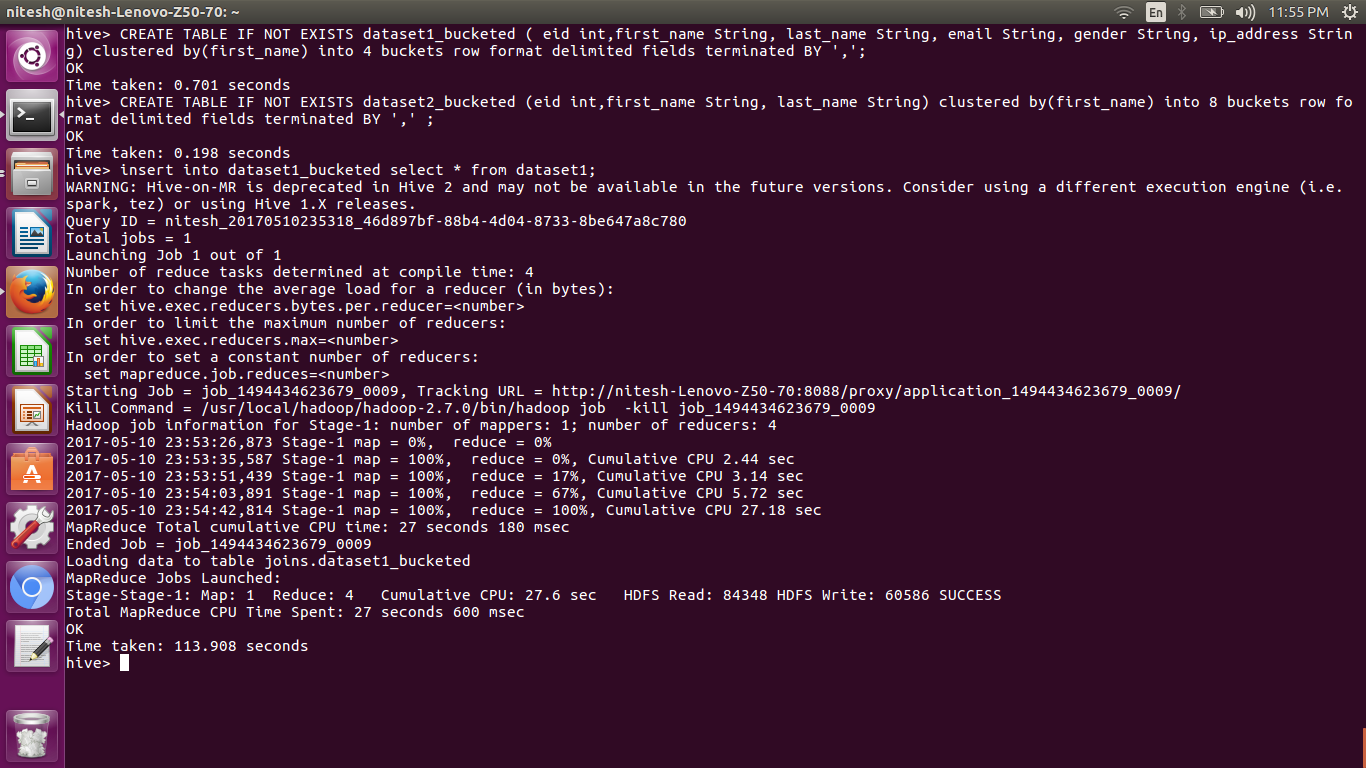
* Bucket Map join

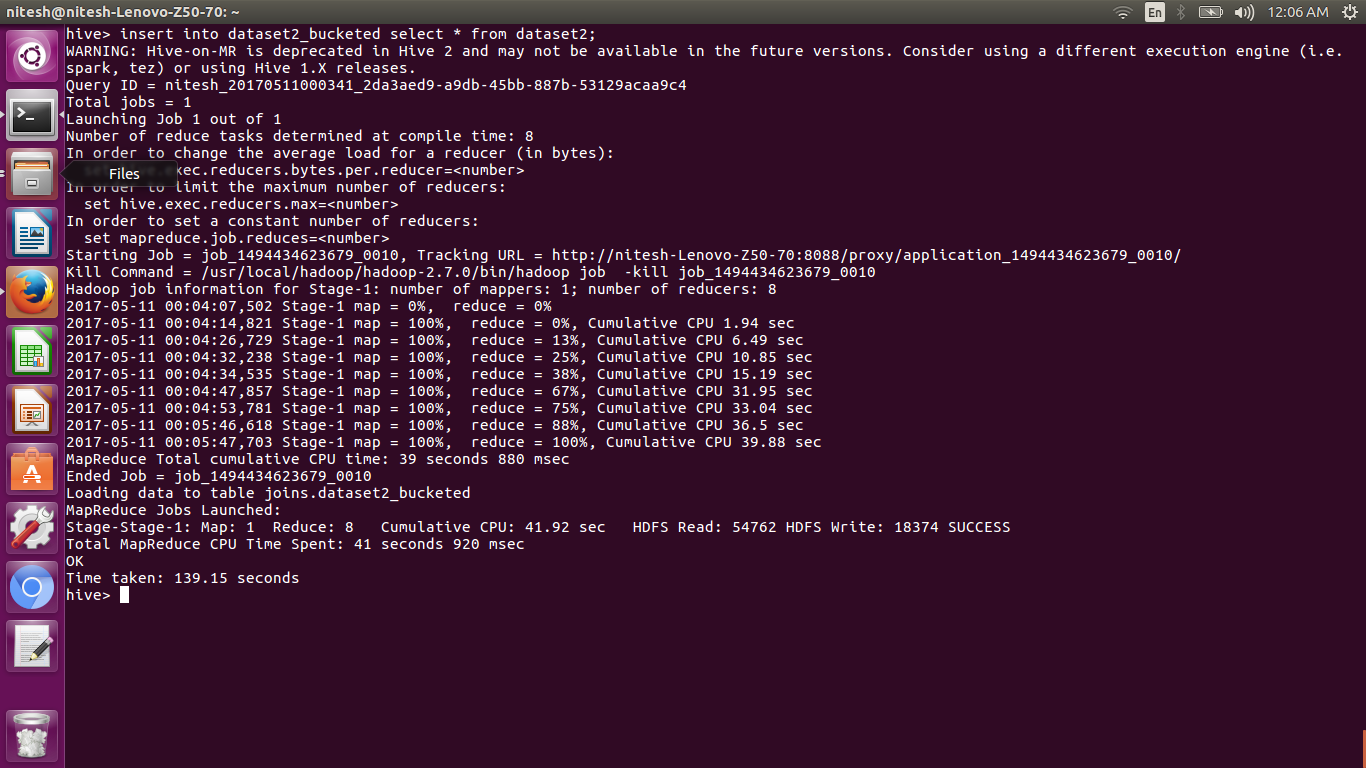
Also known as replicated join, a map-side join is a special type of join where a smaller table is loaded in memory and join is performed in map phase of MapReduce job. Since there is no reducer involved in the map-side join, it is much faster when compared to regular join.

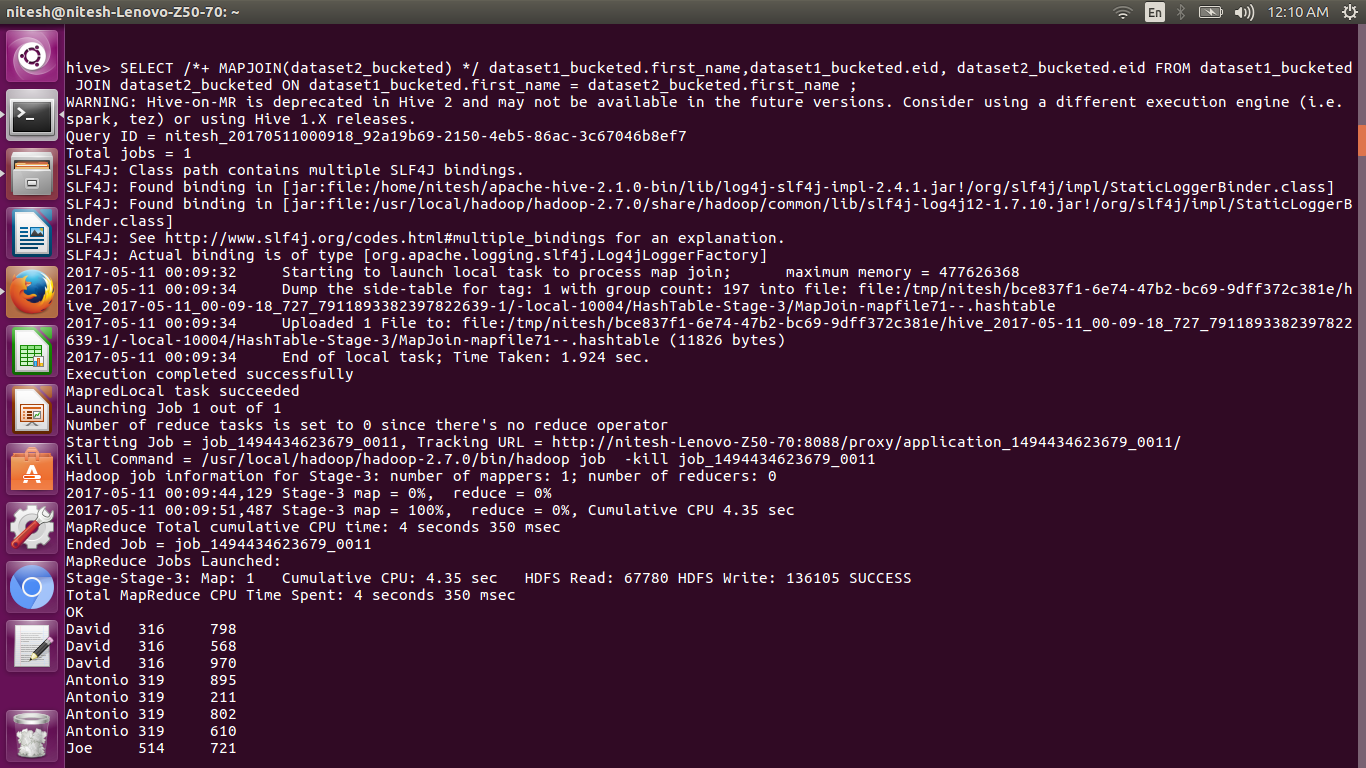
An important point to note is, one table must be small enough to fit into memory. It is recommended to have a proper configuration so that Hive automatically attempt to convert Joins into the map-side join. Below is a Hive join operation which is not a map-side join.





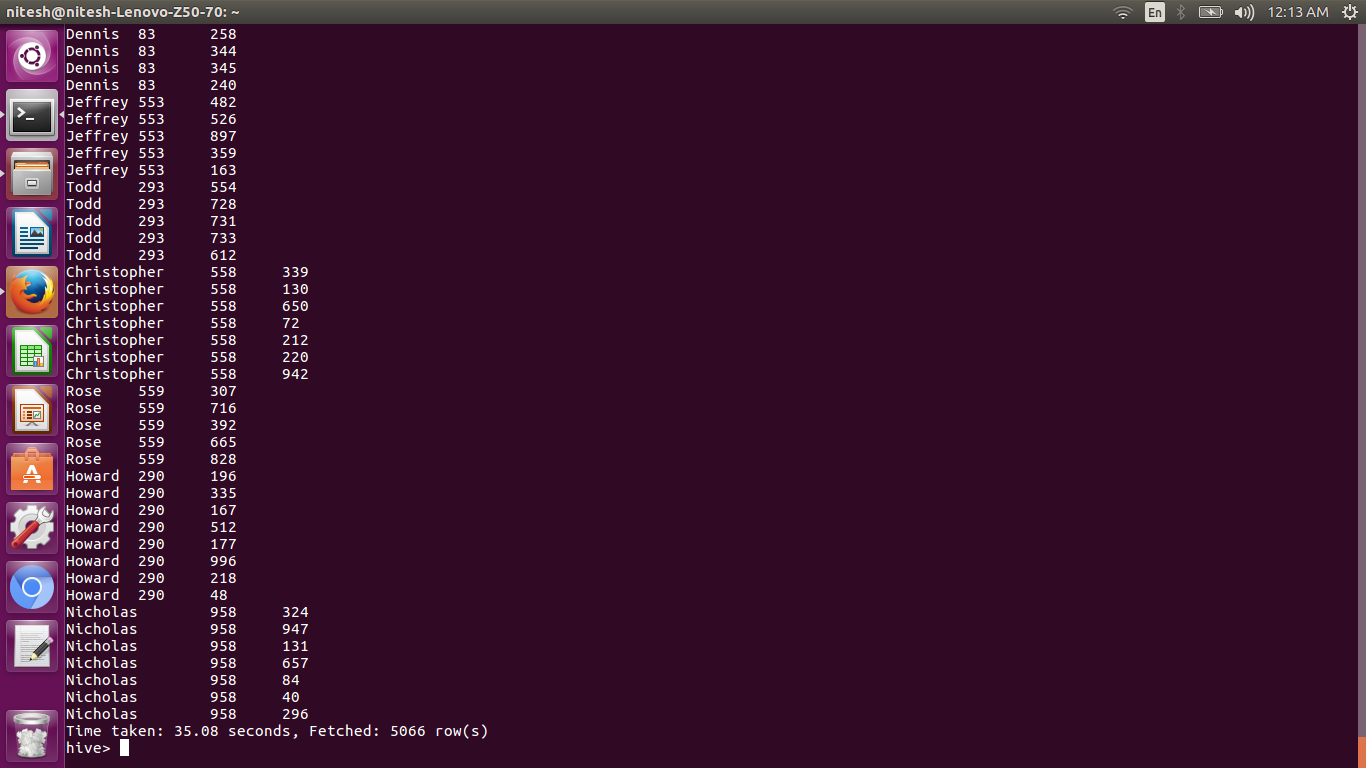












* Sort-Merge Bucket Map join

If the tables being joined are sorted and bucketized on the join columns and have the same number of buckets, a sort-merge join can be performed. The corresponding buckets are joined with each other at the mapper.

Here we have 4 buckets for dataset1 and 8 buckets for dataset2. Now, we will create another table with 4 buckets for dataset2.

For performing the SMB-Map join, we need to set the following properties:

Set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;

set hive.optimize.bucketmapjoin = true;

set hive.optimize.bucketmapjoin.sortedmerge = true;

To perform this join, we need to have the data in the bucketed tables sorted by the join column. Now, we will re-insert the data into the bucketed tables by using sorting the records.

insert overwrite table dataset1\_bucketed select \* from dataset1 sort by first\_name;

The above command will overwrite the data in the old table and insert the data as per the query. So now the data in the dataset1\_bucketed table is sorted by first\_name. You can see the same in the following screenshot:

