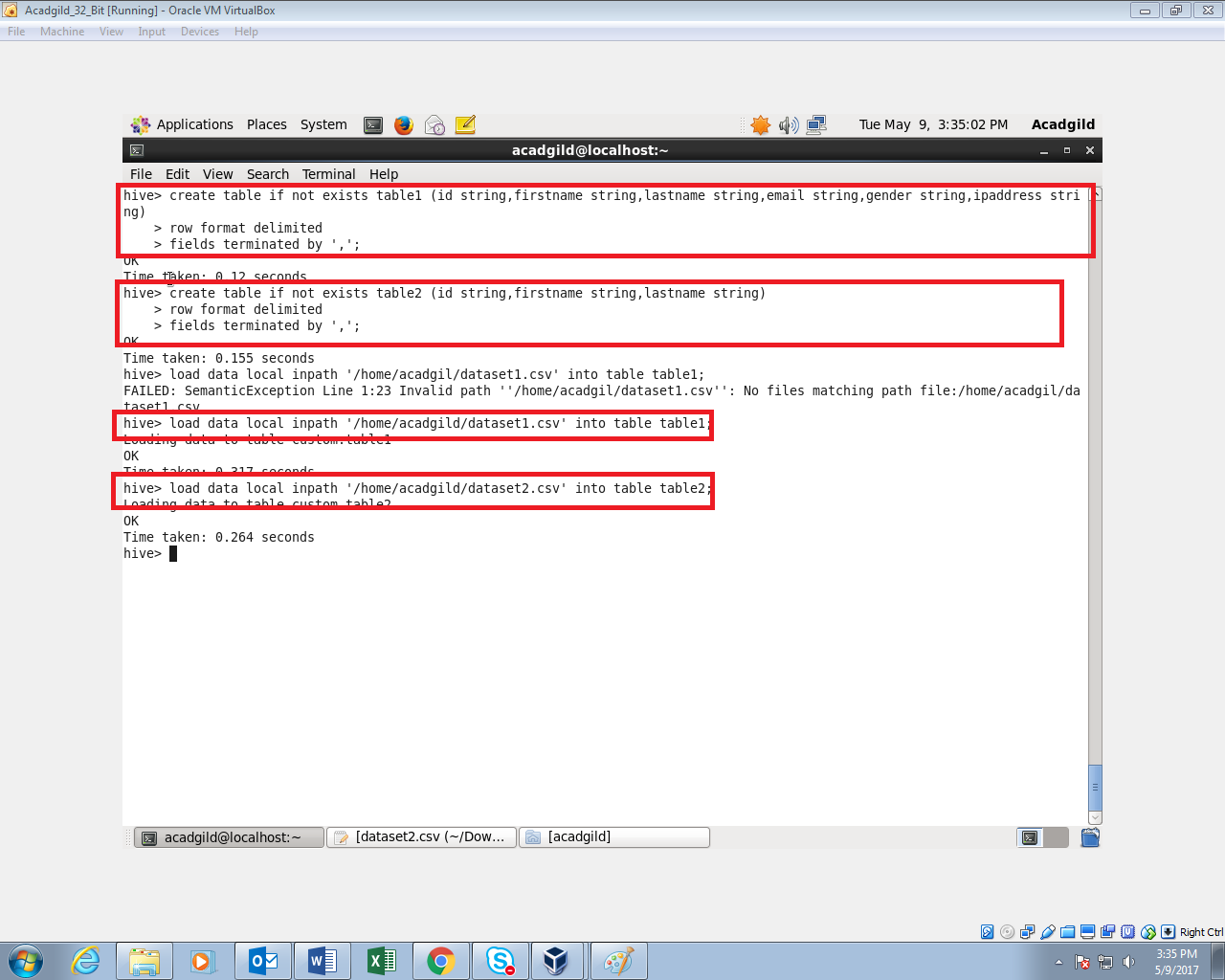
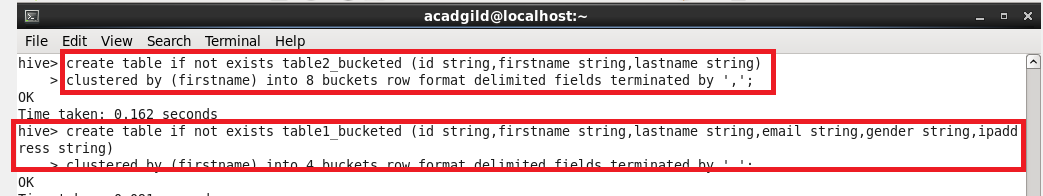
Using the below given two datasets you need to give a demo on the below joins in hive.

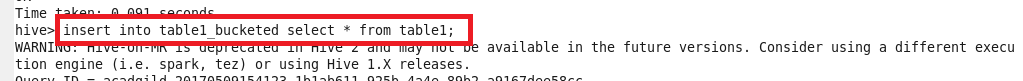
* Bucket Map join
* Sort-Merge Bucket join
* Sort-Merge Bucket Map join
* Left semi join

Bucket map join:

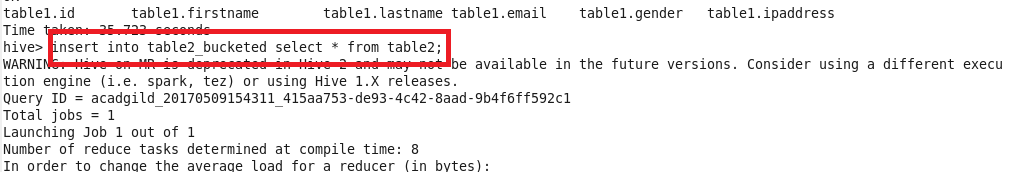
To perform bucketing, we need to have bucketed tables.so I’m creating 2 bucketed tables. One with 4 buckets and one with 8 buckets



Inserting the data in table1 into the bucketed table table1\_bucketed

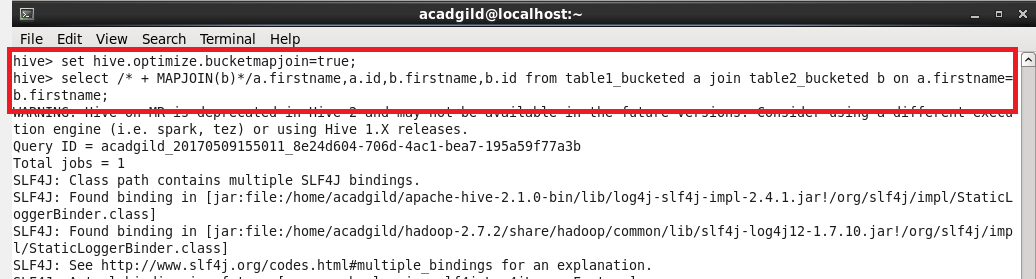


Inserting the data in table2 into the bucketed table table2\_bucketed

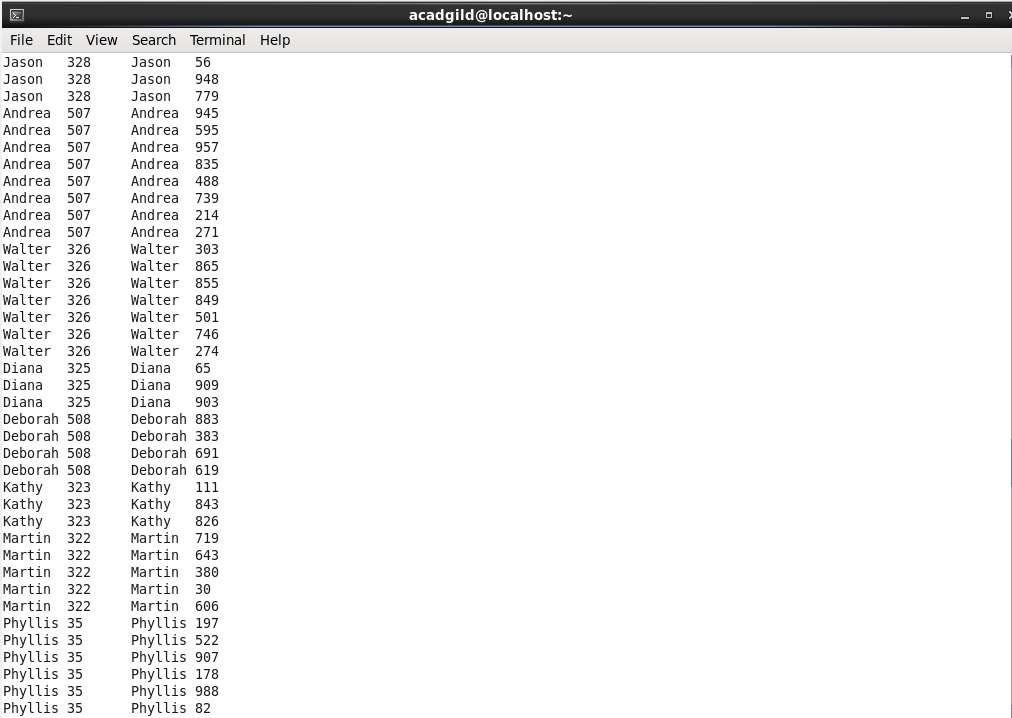


For performing Bucket-Map join, we need to set this property in the Hive shell.

set hive.optimize.bucketmapjoin = true



Output:

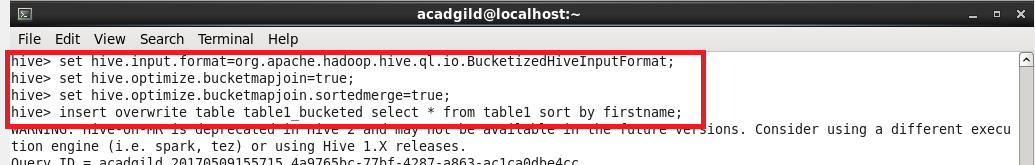


Sort-merge bucket join:

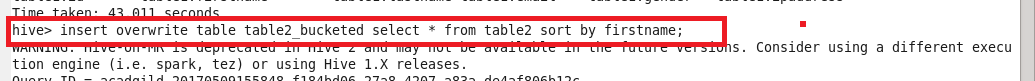
Sort-Merge bucket join is like reduce side join. Before joining we’ll have to sort the data and then only merge.

Moreover in this join we should set

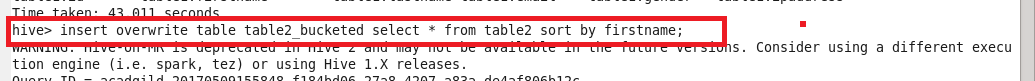
set hive.auto.convert.join= false;



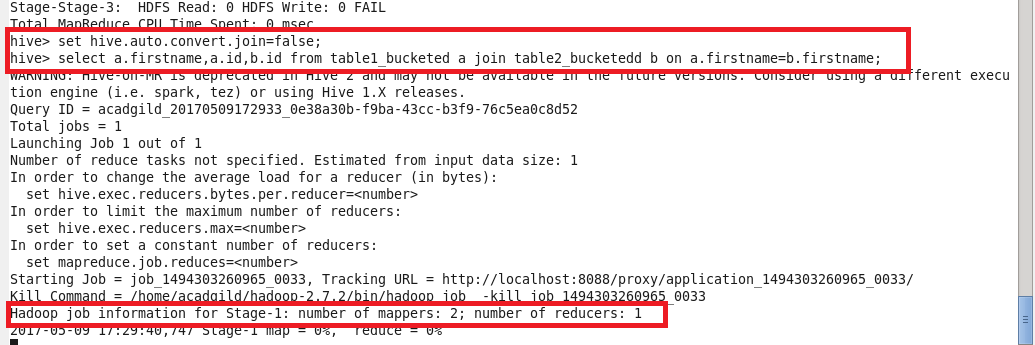
Sorting the contents of bucketed table 2



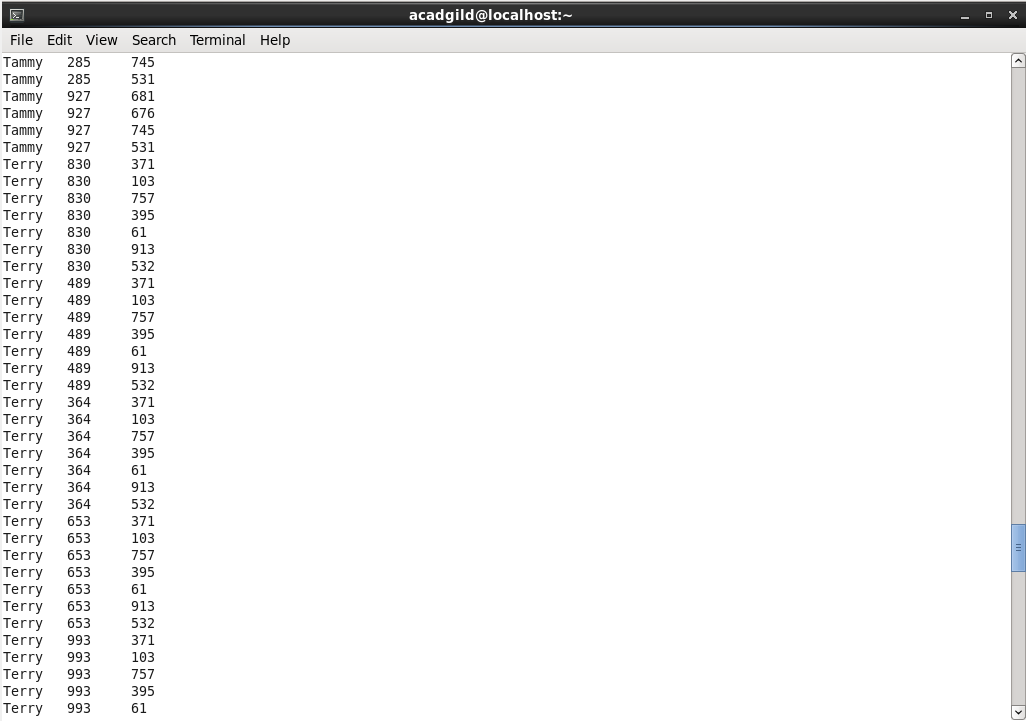
Sorting the contents of bucketed table 1



query for extracting data



Output:

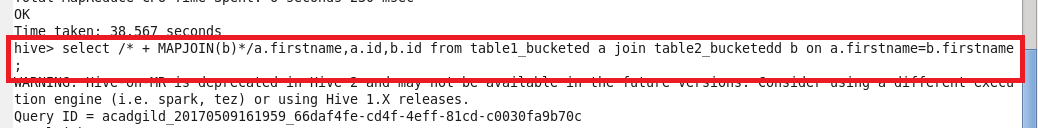


Sort-merge bucket map join:

since it is sort-merge bucket map join we need to set

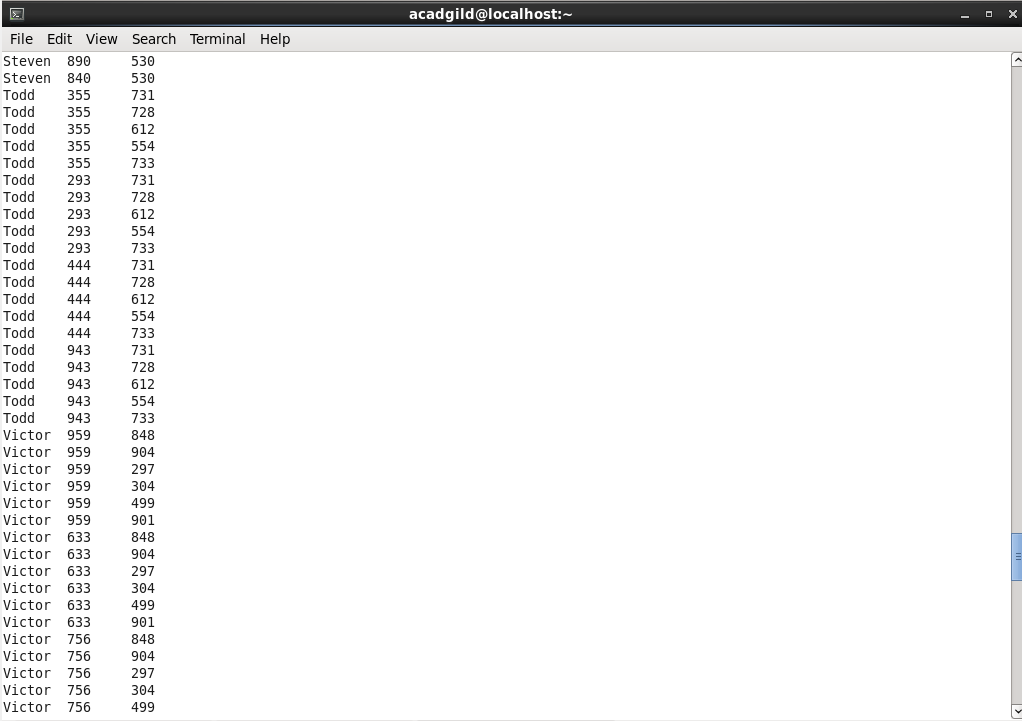
set hive.auto.convert.join= true;

but without setting this also we can run a map only job by typing like this

******

This query runs a map only job. I’ve used the same bucketed tables that I’ve created previously for this join

The output will be like this,

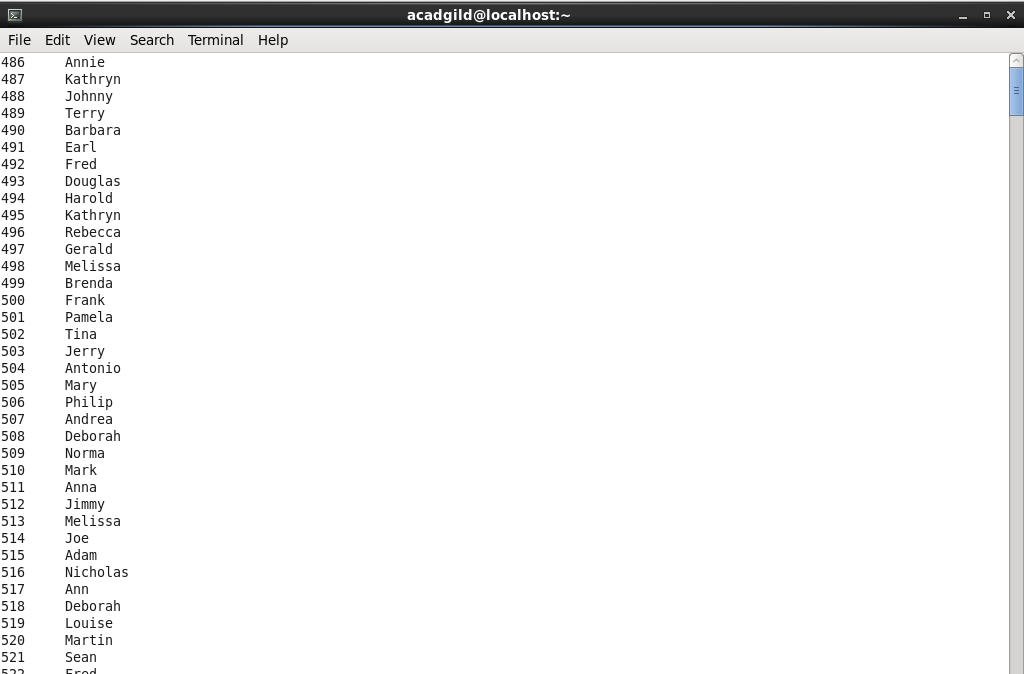
******

Left semi join:

Lef semi join eliminates the use of where and in keywords that we use in typical sub queries.

******

Output:

******