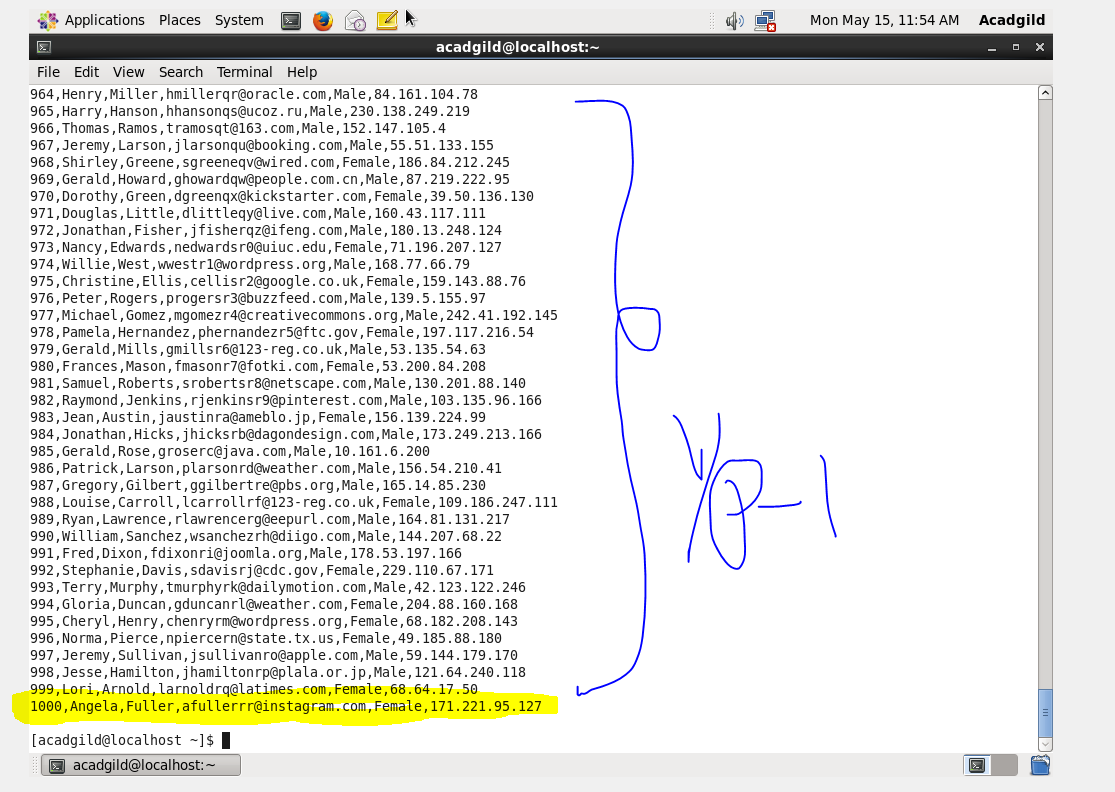
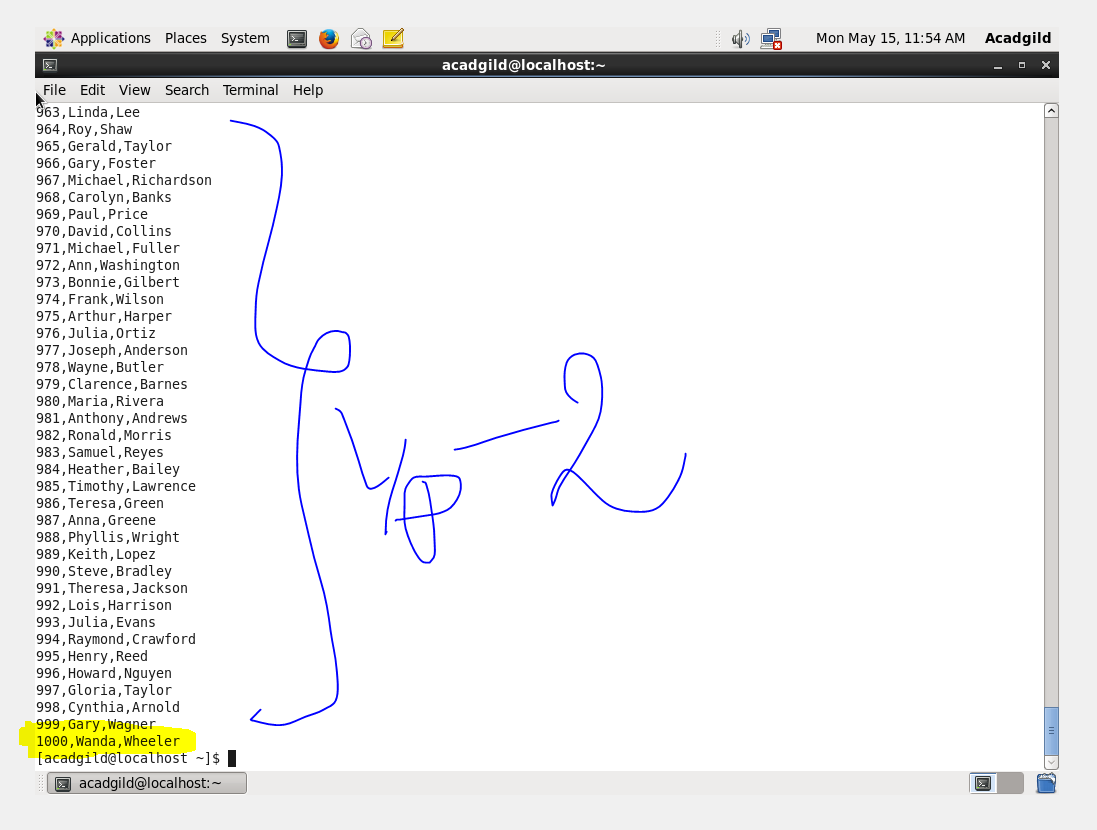
**Give a demo on the below joins in hive.**

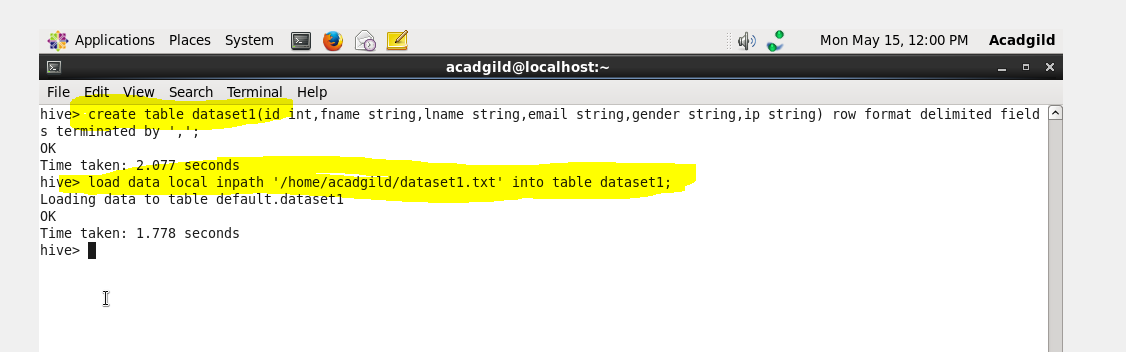
**Input Dataset 1**

****

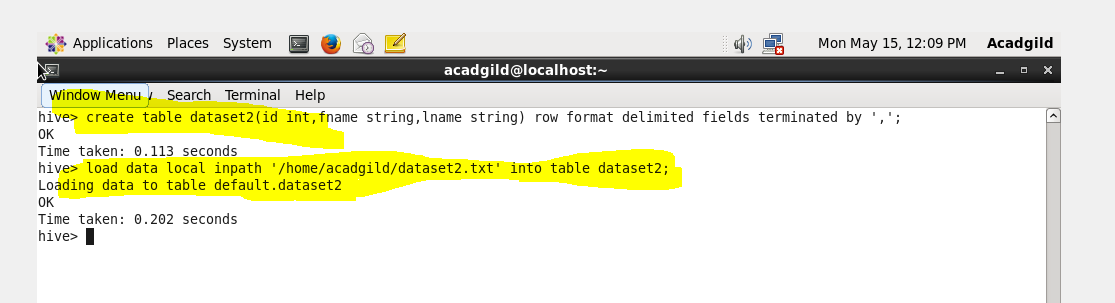
**Input Dataset 2**

****

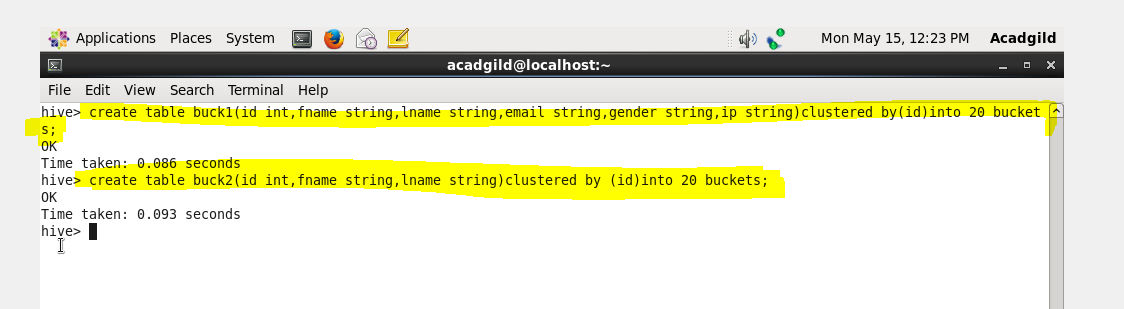
**Creating a table & loading the dataset1**

****

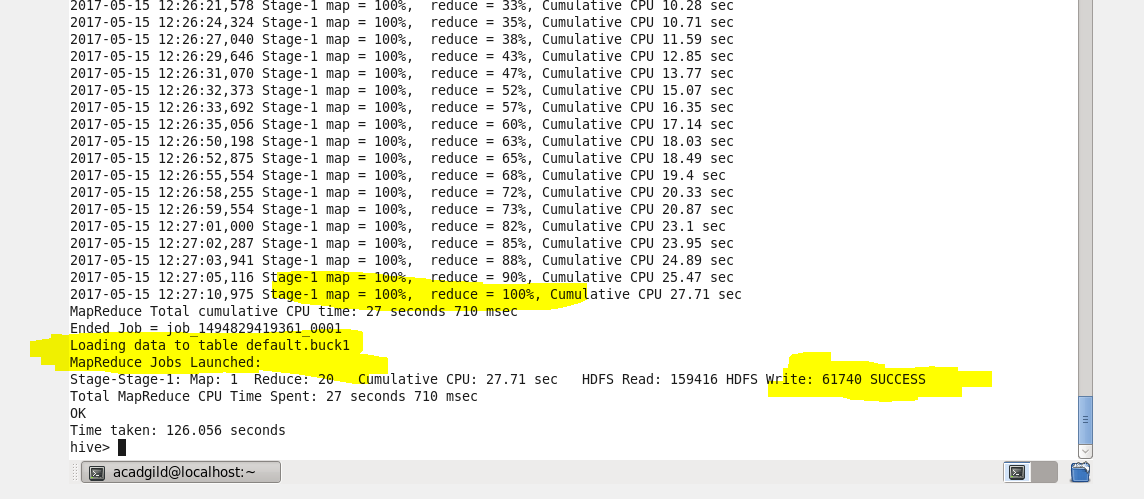
**Creating a table & loading the dataset2**

****

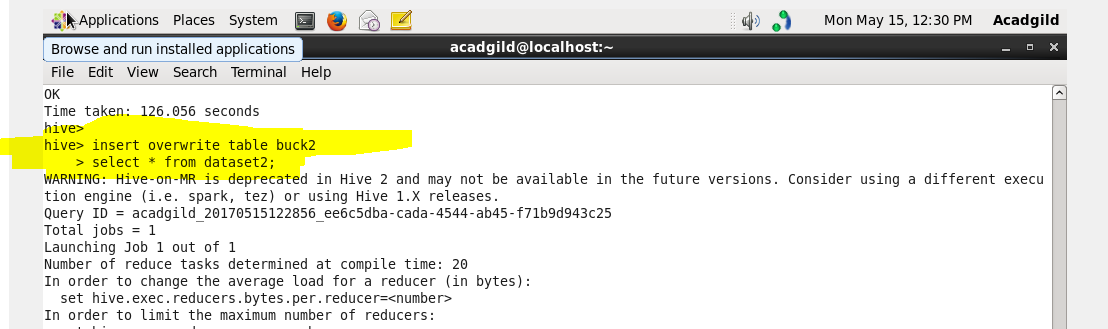
**Creating a bucket1 and inserting dataset1**

****

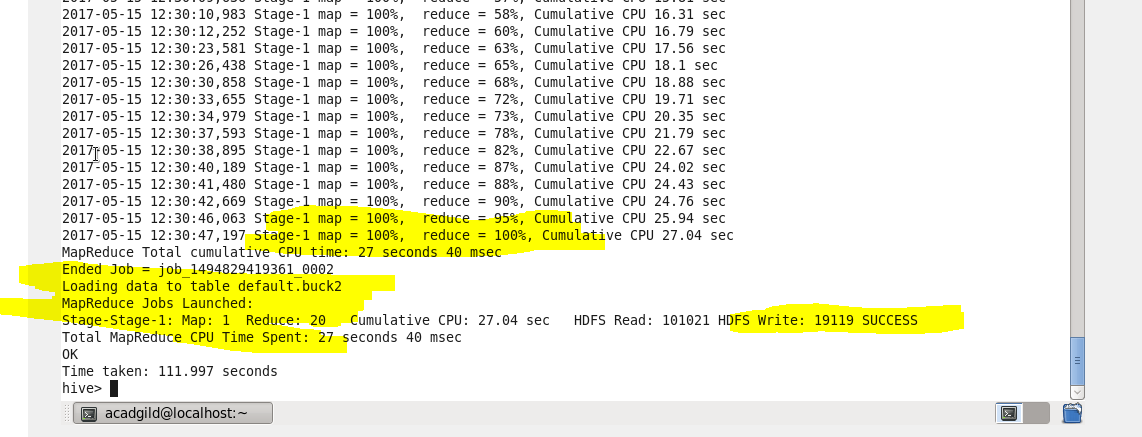
**Output**

****

**Creating a bucket2 and inserting dataset2**

****

**Output**

****

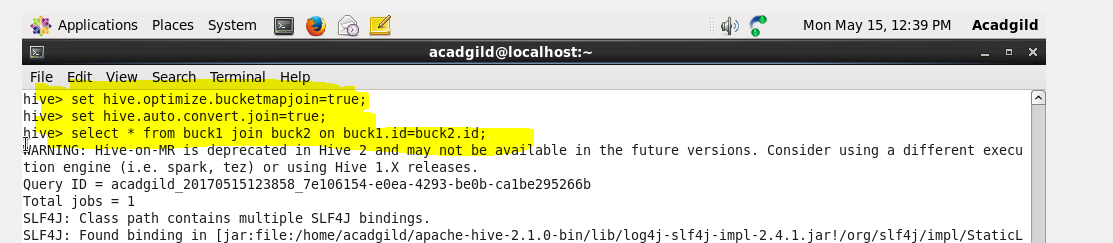
**Bucket Map join**

* Total table/partition size is big, not good for mapjoin
* Non-sorted the same
* Bucketd the same
* Joining on the sort/bucketing on the multiple columns

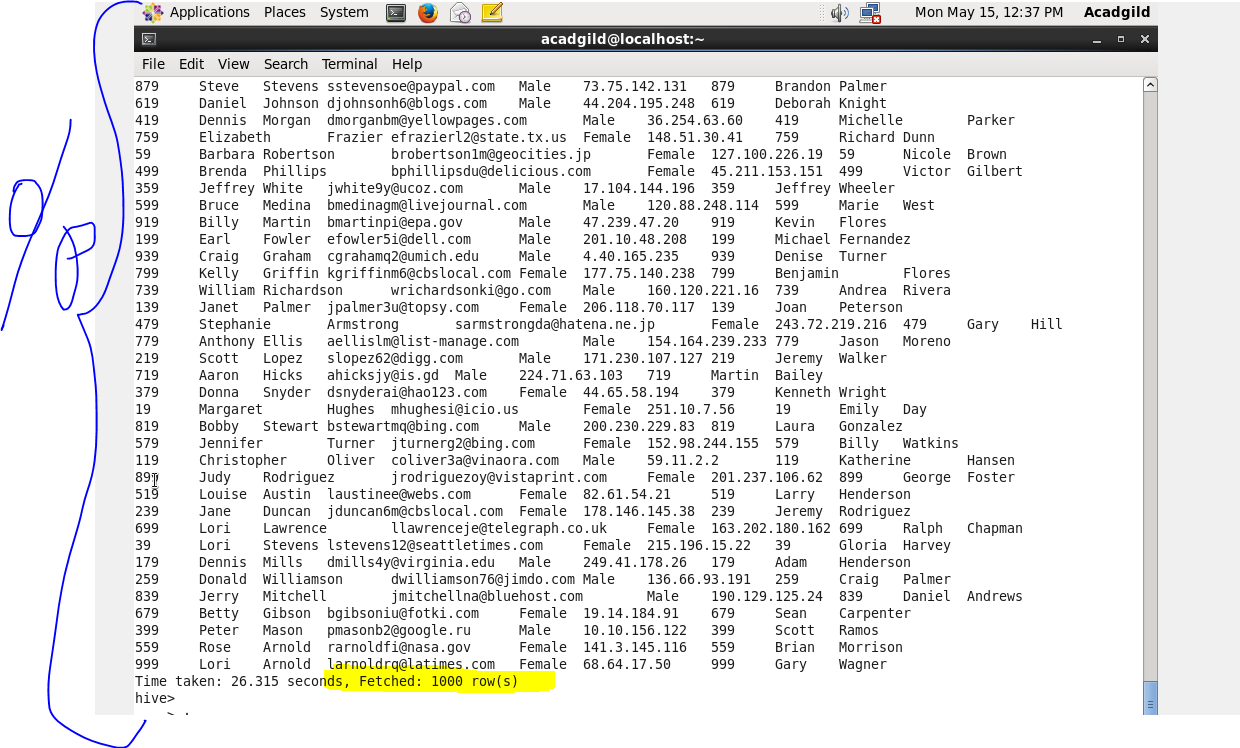
Pointers:

* set hive.optimize.bucketmapjoin; If true  then Bucket Map Join is activated

**Commands**



**Output**



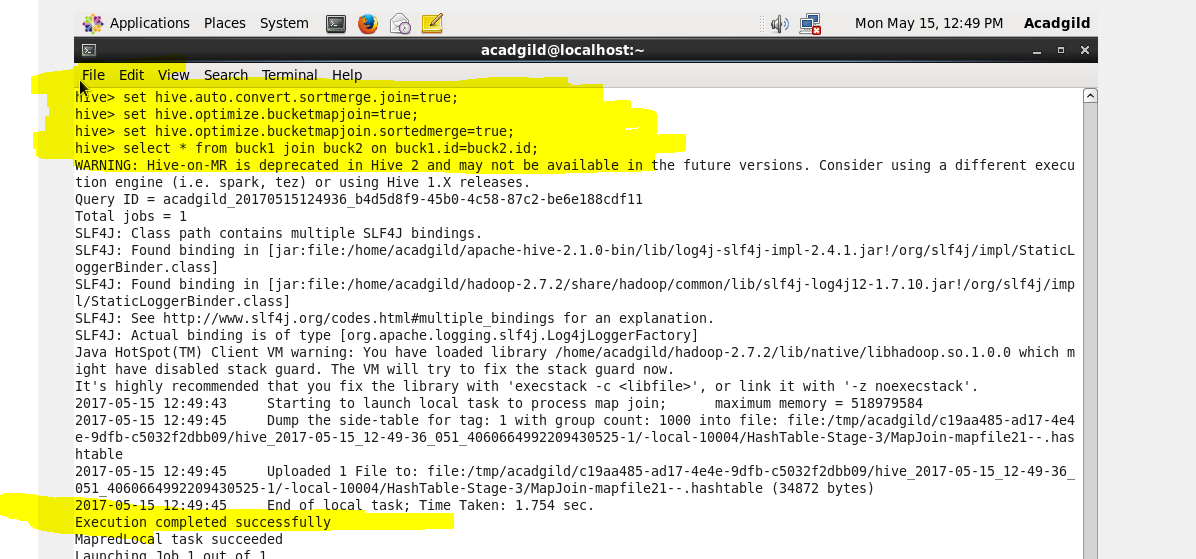
**Sort-Merge Bucket join**

* Sorted the same
* Bucketed the same
* Joining on the sort/bucket on the same/equal columns

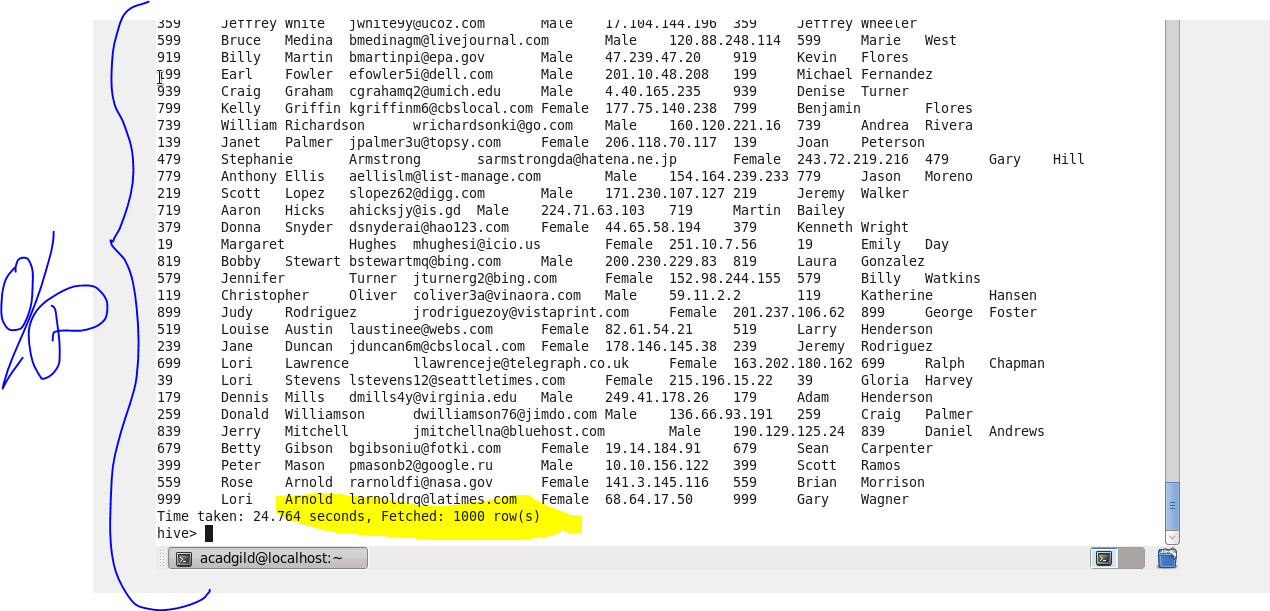
Pointers:

* Very efficient if applicable
* Both Mapt & Reduce task are used
* set hive.auto.convert.sortmerge.join=true
* set hive.optimize.bucketmapjoin = true
* set hive.optimize.bucketmapjoin.sortedmerge = true

**Commands**



**Output**



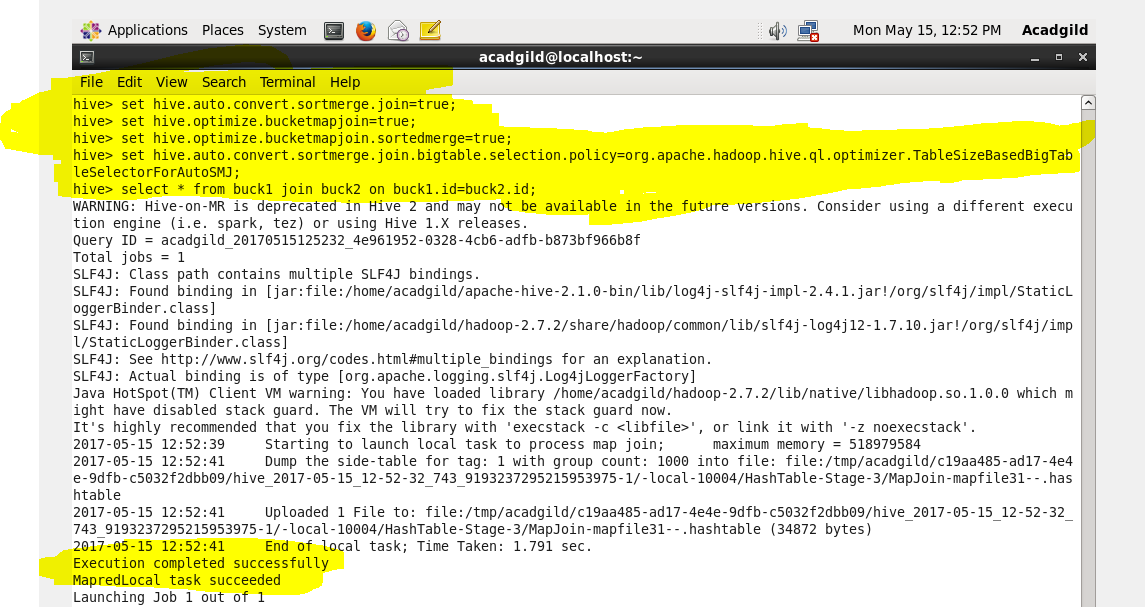
**Sort-Merge Bucket Map join**

* Sorted the same
* Bucketed the same
* Joining on the sort/bucket on the same/equal columns
* No limit on file/partition/table size

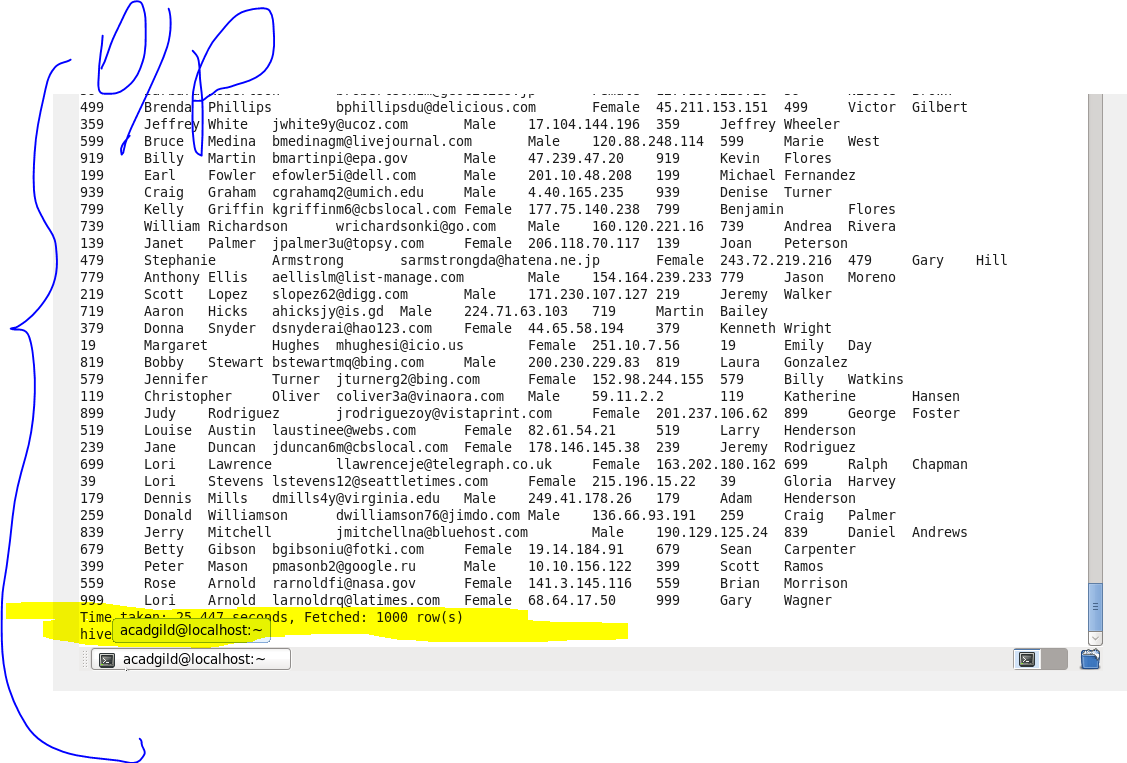
Pointers:

* set hive.auto.convert.sortmerge.join=true
* set hive.optimize.bucketmapjoin = true
* set hive.optimize.bucketmapjoin.sortedmerge = true
* set hive.auto.convert.sortmerge.join.bigtable.selection.policy=org.apache.hadoop.hive.ql.optimizer.TableSizeBasedBigTableSelectorForAutoSMJ

**Commands**



**Output**



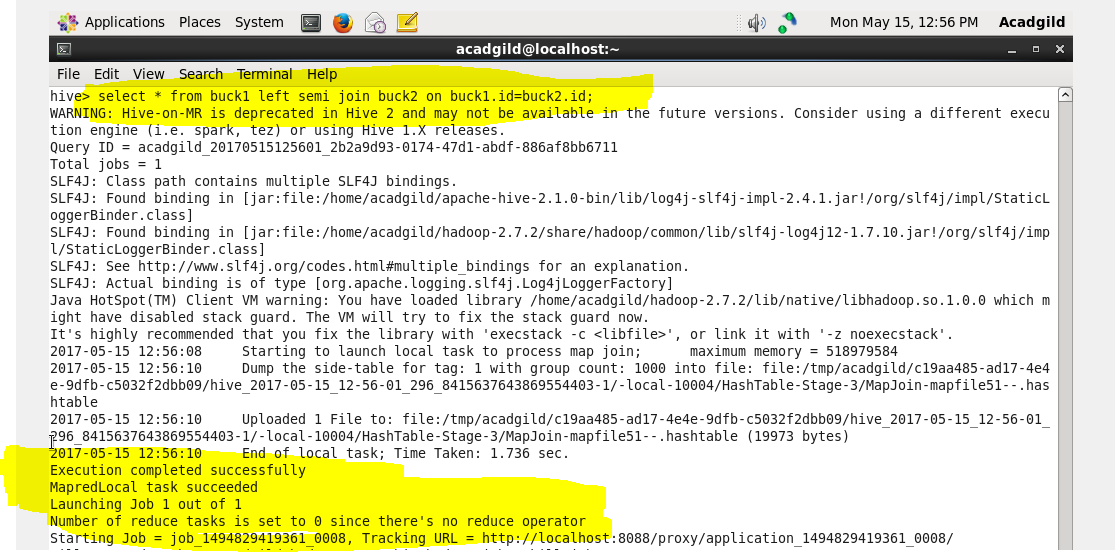
**Left semi join**

* To have functionality of IN/EXISTS subquery semantics

Pointers:

* Right hand side table should only be reference with ON clause, not with WHERE/SELECT clause
* Right semi-joins are not supported in Hive

**Commands**



**Output**

