**Optimisation In Hive**

Join is one of the most Important feature of Hive.Join Optimisation means the use of joins in such a manner to ensure efficient querying.

Join optimization could be done by

**1. Join table ordering (Largest table last).**

**2.Map Side Join**

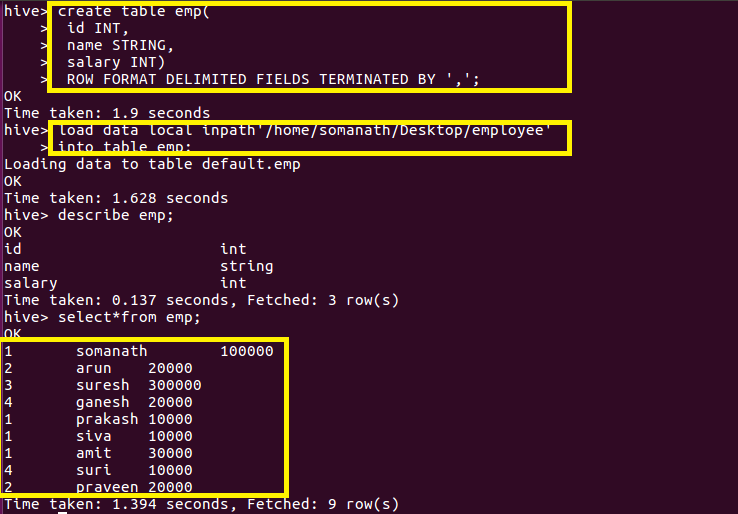
**3.Bucket map join**

**4.Sorted Merge Bucket Map Join**

In order to perform join in Hive I am creating the following two Table

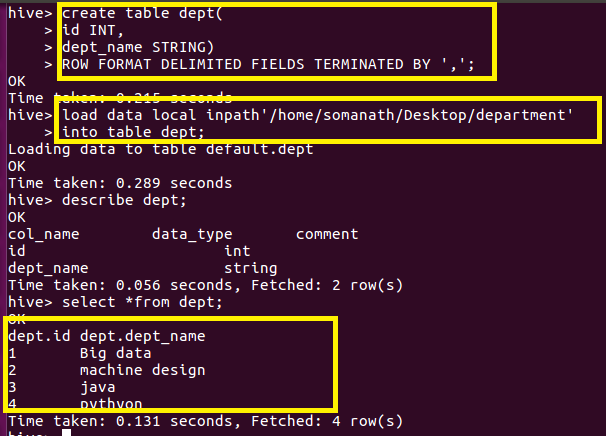
**Employee Table**

It consists of id(dept\_id), name and salary as shown



**Department Table**

It consists of dept id and department name

****

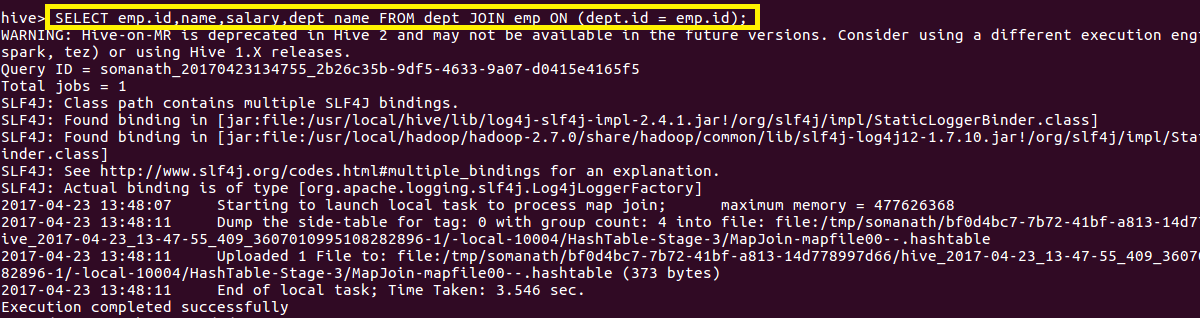
**Join table ordering (Largest table last).**

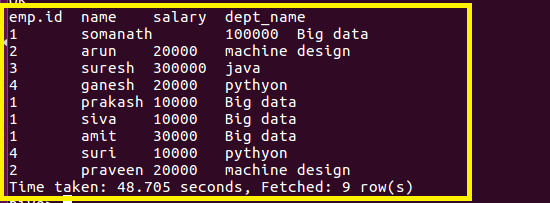
**In the normal join where joins takes place in reducer**

**In every map/reduce stage of the join, the last table in the sequence is streamed through the reducers where as the others are buffered. Therefore, it helps to reduce the memory needed in the reducer for buffering the rows for a particular value of the join key by organizing the tables such that the largest tables appear last in the sequence**.

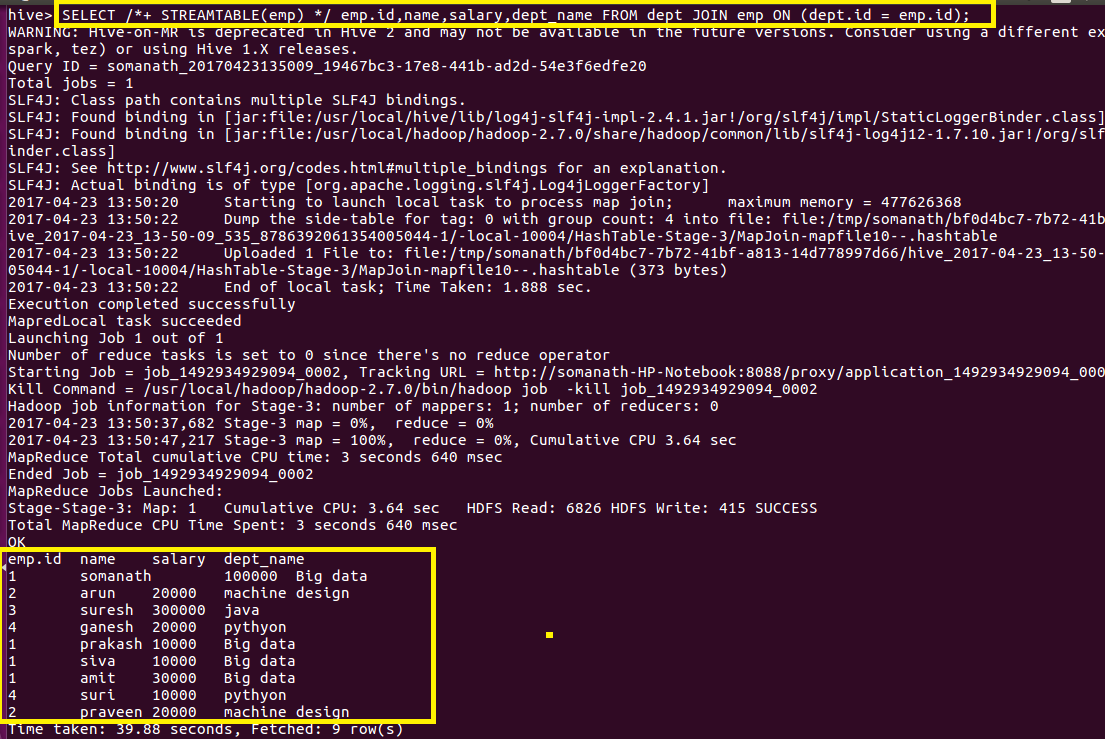
**It is general rule that the large table should be streamed**

**For example In our case the large table is employee table so it is joined as a last table**

****

****

**Instead we can also explicitly stream the table using streamable command as highlighted below**

****

**Map Side Join**

**In normal join, mappers read data of tables on which join needs to be performed and emit key as join key or column on which is expected to be performed . Thus MapReduce framework distributes join key data to single reducer**

**In normal Join in hive joining will take place is reducer but since in this joins joins will be made through buffer memory (which contains small dataset) and large dataset will be streamed in reducer which will actually takes long time**

**Instead of this we can use Map Side Join where joining will be take place in mapper itself**

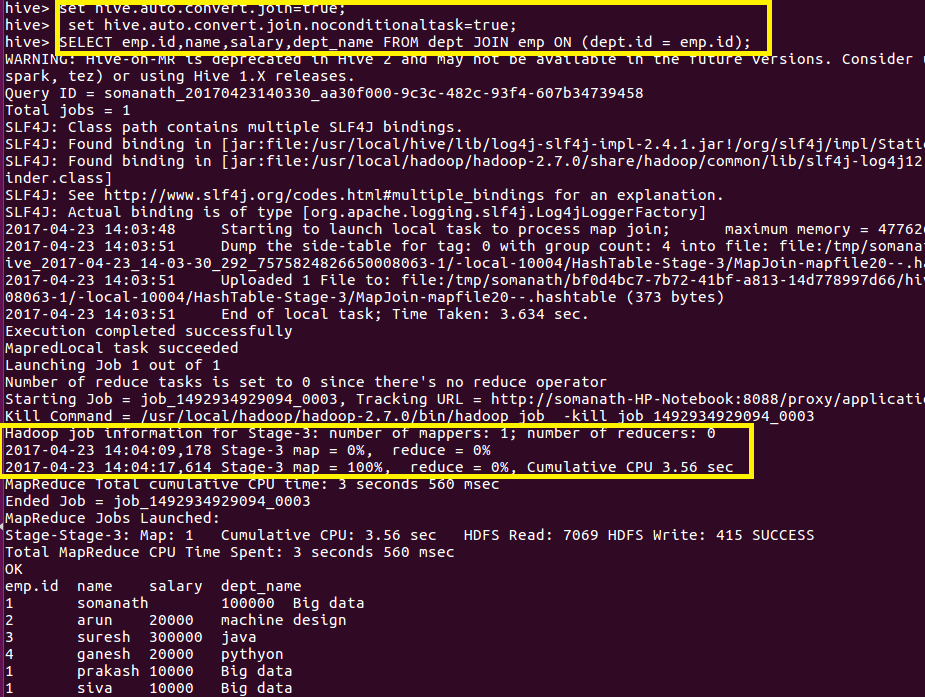
**Which will results in faster execution of query as this takes place in mapper through distributed cache**

**In order to do map side join set.hive.auto.convert=true;**

**set.hive.nonconditional.task=true;**

**OUTPUT:**

**WE can see that number of reducers=0 and reduce=0% which implies that Mapside Join occurs**

****

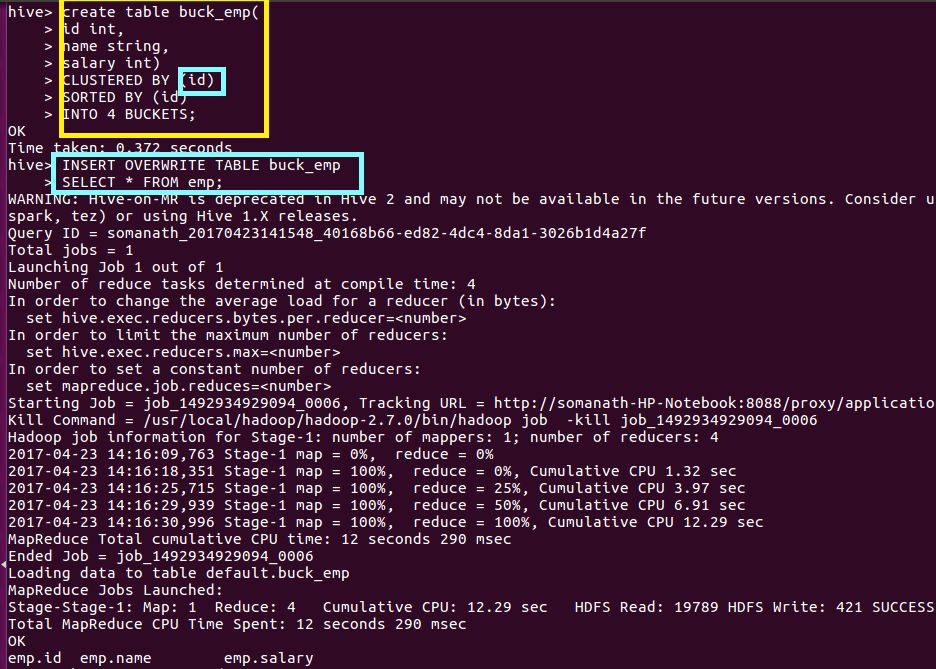
**Bucket Map Join**

**Since in ordinary map join join takes place in mapper and join will be made after reading the full record of small dataset in distributed cache and keys will be generated and will be joined with other dataset and It will take time**

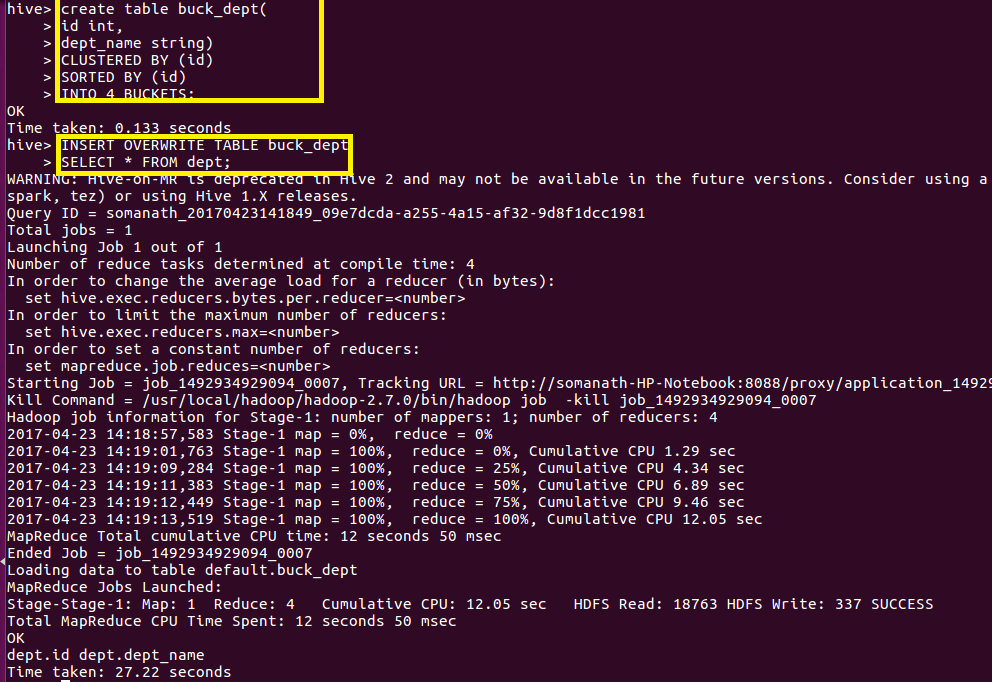
**But if we have bucketing is made on Id on which joins should be performed Then as both table contains bucket separate files will be created .hence while using map side join Hive will directly put the same bucket file on same mapper and join will be performed faster**

**For example,here id=1 bucketted file of buck\_emp and id=1 bucketted file of buck\_dept will be sent to mapper and join will be faster**

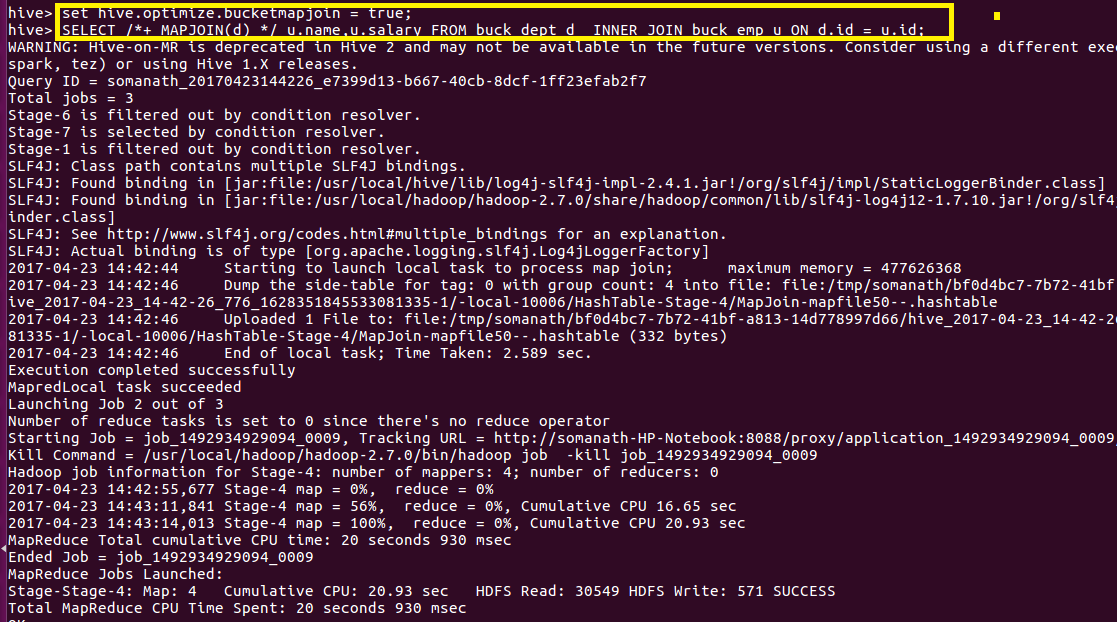
**Creating Bucket Table for employee**

****

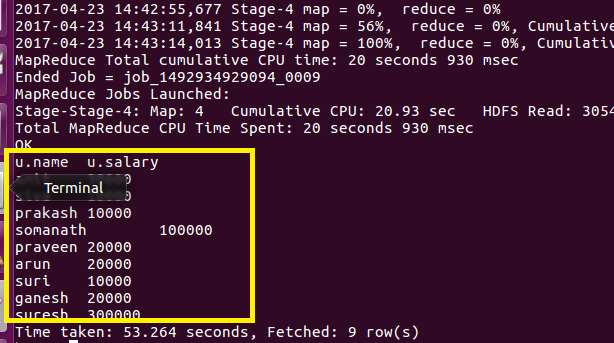
**Creating Bucket table for Department**

****

**Then in order to perform bucket map join set hive.optimize.bucketmapjoin=true and perform map join**

****

**Output**

****

**Sort-Merge-Bucket-Map Join**

**In this since the data is already sorted and bucketed based on key on which joining should be performed it will be easier for hive to perform join and it results in high optimization.It is a improvisation over bucket map join.**

**Setting configuration for Sort-Merge-Bucket-Map Join**

****

**Output**

**Here we are performing join on the above created bucketed tables**

