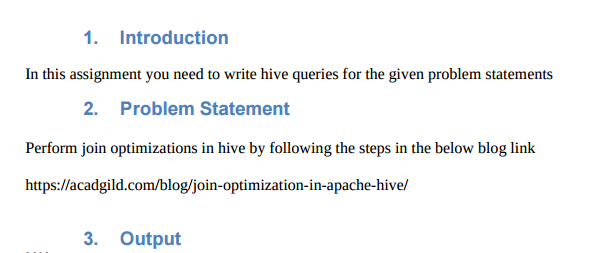
**Assignment 27.5**



**JOIN OPTIMIZATION :**

Join Optimisation means the use of joins in such a manner to ensure efficient querying.So as to speed up the query execution time and optimize our hive query.

Join optimization can be done by

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

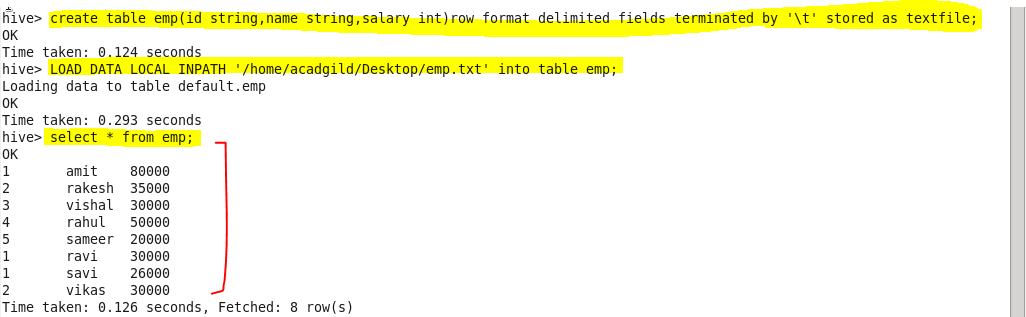
**2. Map Side Join**

**3. Bucket map join**

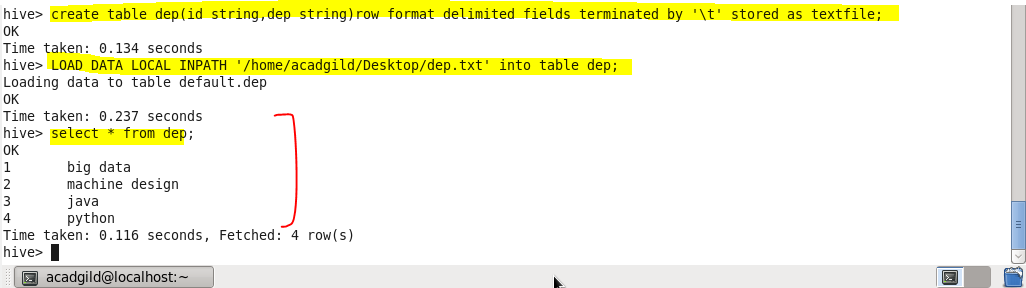
**4. Sorted Merge Bucket Map Join**

In order to perform join in Hive I am creating the following two table employee and department then will load the dataset into the created table and to show the data in our table we will use select command

**Employee Table**



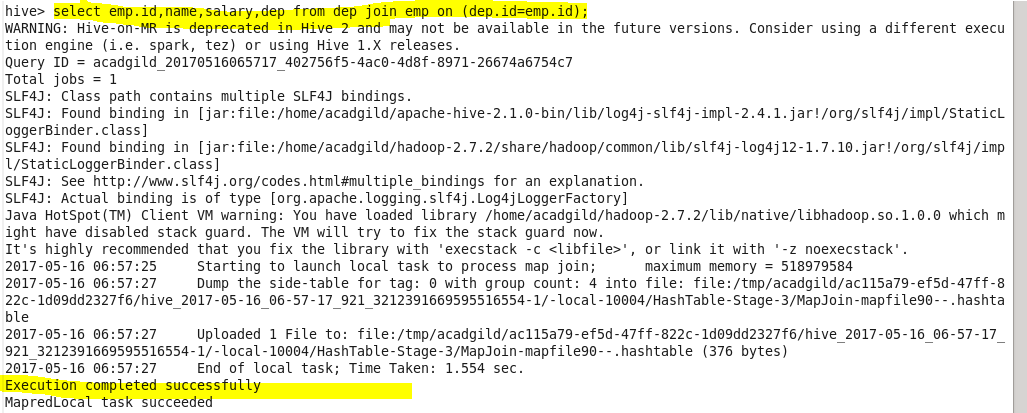
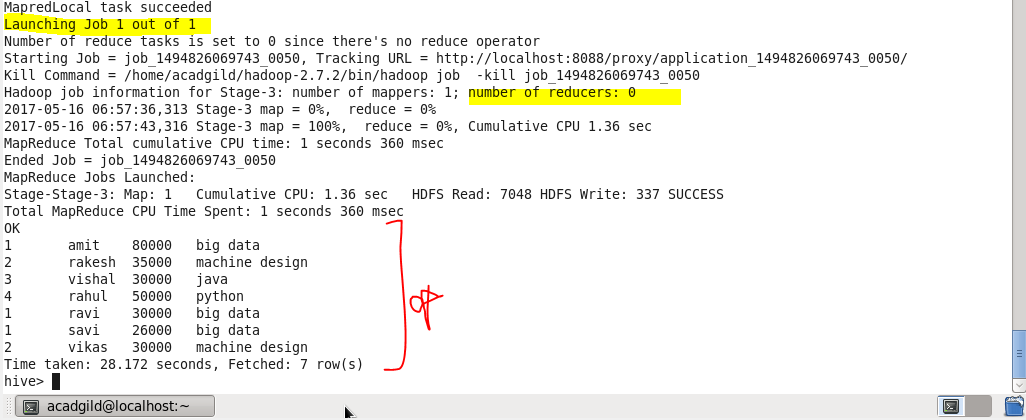
**Department Table**



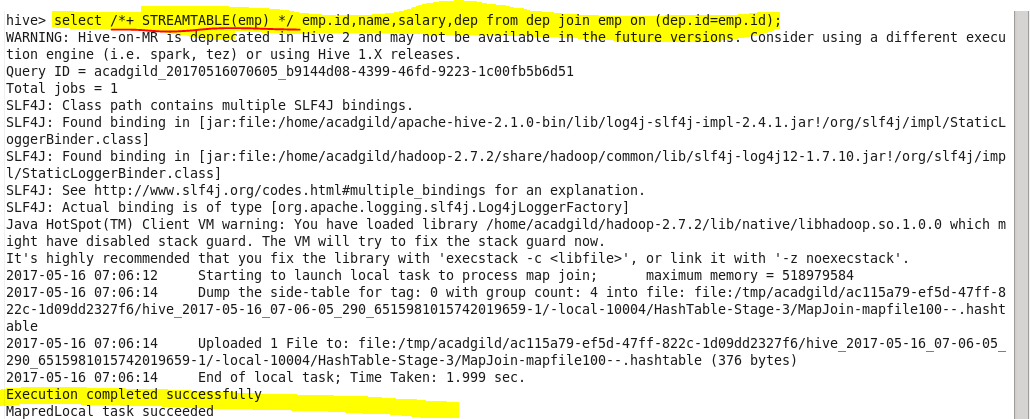
**Join table ordering (largest table last):**

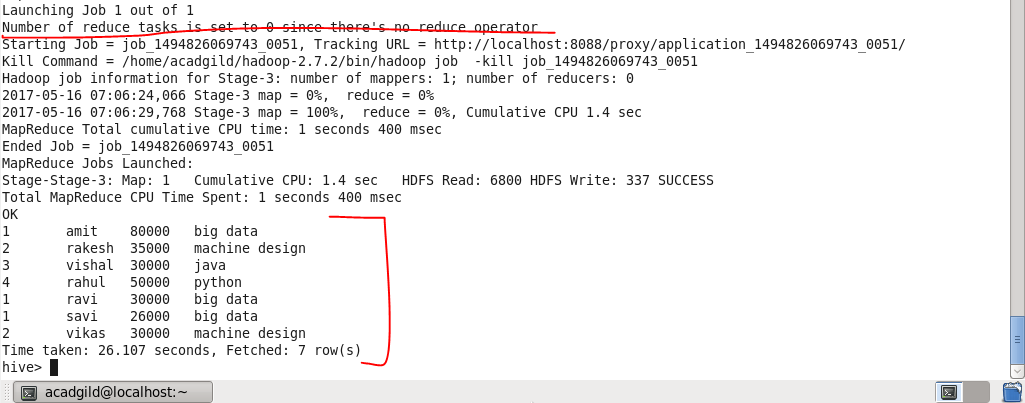
When Hive executes a join, it needs to select which table is streamed and which table is cached. Hive takes the last table in the JOIN statement for streaming, so we need to ensure that this streaming table is largest among the two. Here, largest table is employee that is table with larger data.

Or we can also explicitly tell Hive which table it should stream.

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





**Mapside join (join optimization also known as replicated join) :**

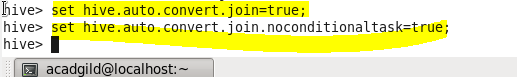
* In normal join, mappers read data of tables on which join needs to be performed and emit key as join key or column on which is expected to be performed . Thus MapReduce framework distributes join key data to single reducer.
* In normal Join in hive joining will take place is reducer but since in this join, join will be made through buffer memory (which contains small dataset) and large dataset will be streamed in reducer which will actually takes long time.

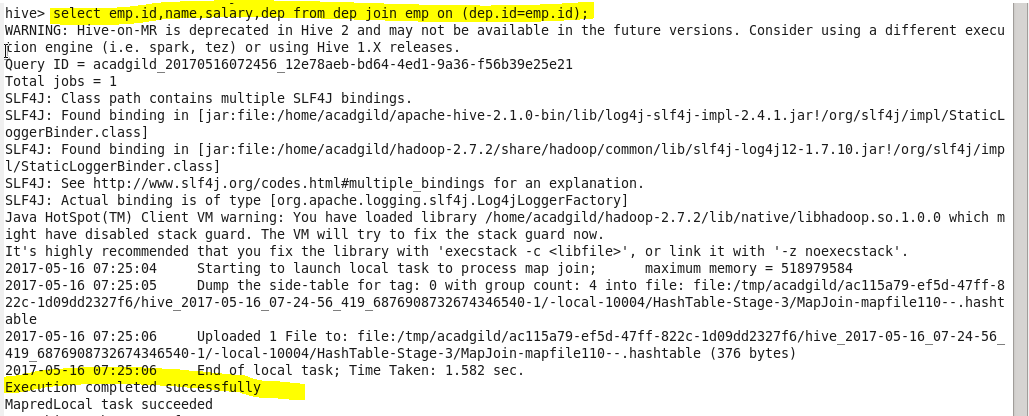
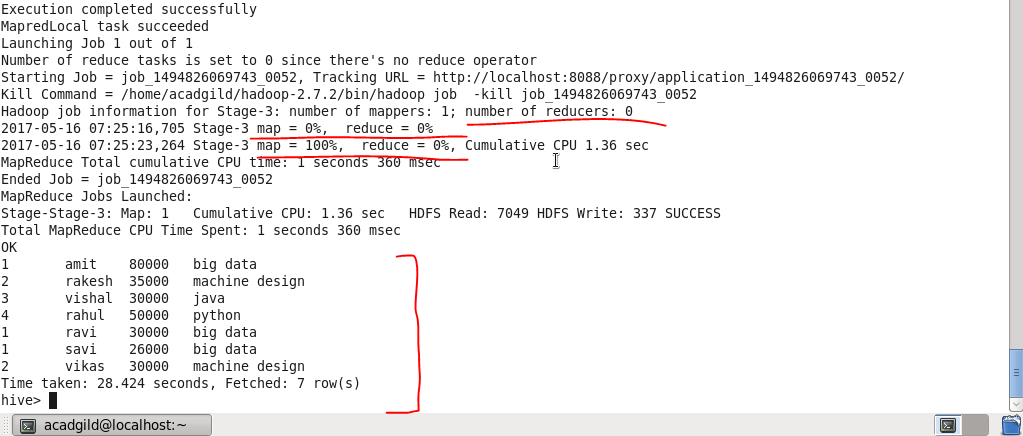
Instead of this we can use Map Side Join where joining will be take place in mapper itself.Which will results in faster execution of query as this takes place in mapper through distributed cache

* **In order to do map side join set**

Set hive.auto.convert.join=true;

Set hive.auto.convert.join.noconditionaltask=true;



Hence, the map-side join is faster than regular join operation since no reducer is used.

**Sort-Merge-Bucket (SMB) Map Join:**

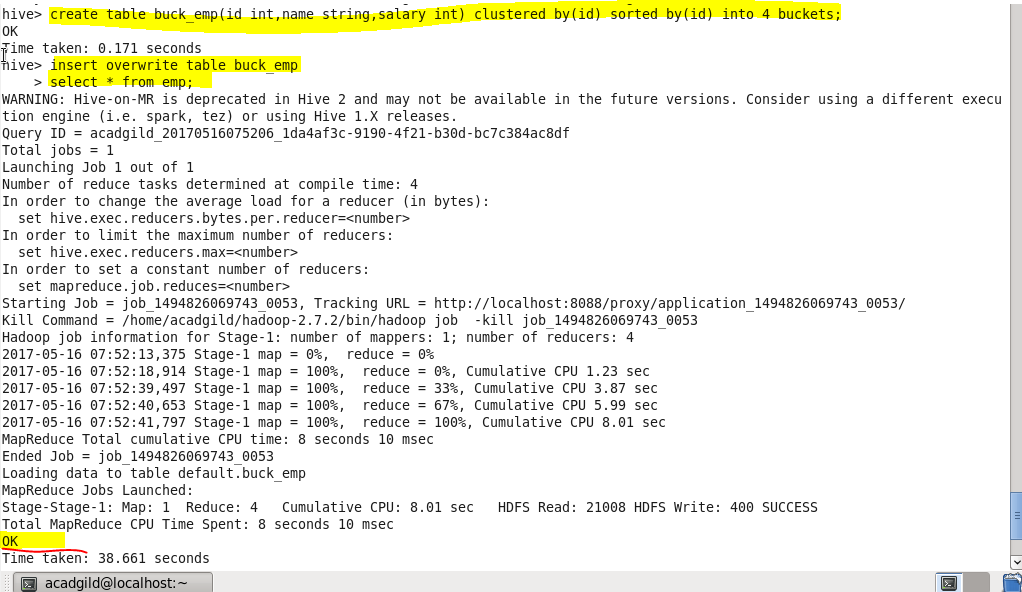
It is another Hive join optimization technique where all the tables need to be bucketed and sorted. In this case joins are very efficient because they require a simple merge of the presorted tables.

Creating bucketed tables from existing tables i.e , employee and department. Before that we need to set the configurations.

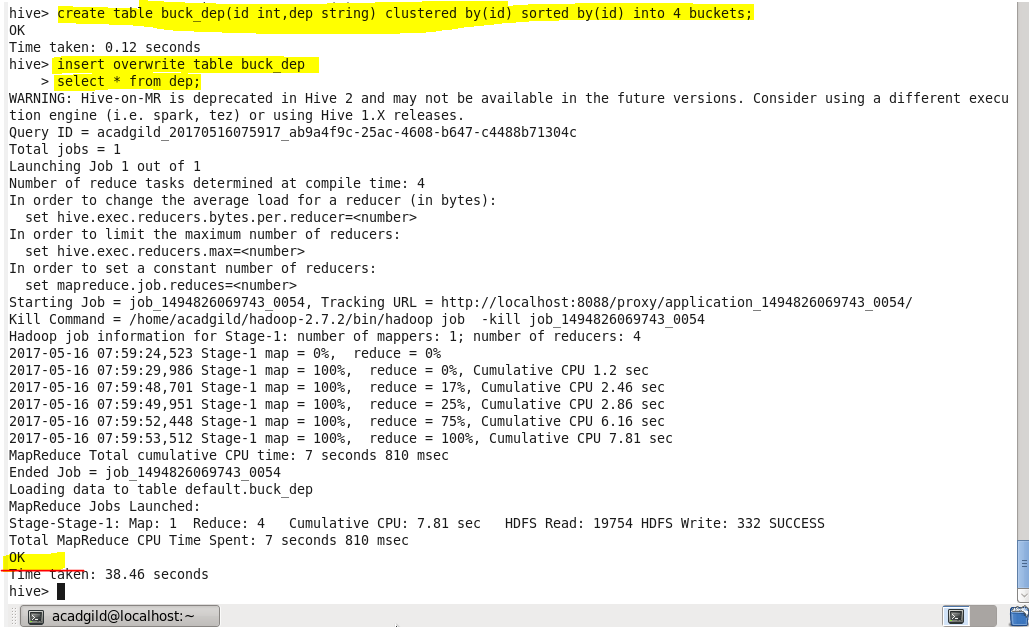
Set hive.enforce.bucketing=true;

Set hive.enforce.sorting=true;

Creating bucketed table for emp table as buck\_emp and inserting to it.



Creating bucketed table for dep table as buck\_dep and inserting to it.



Now the stage is set to perform SMB Map Join to optimize Hive joining. Again, make some changes in properties to perform SMB Map join.

hive> set hive.enforce.sortmergebucketmapjoin=false;

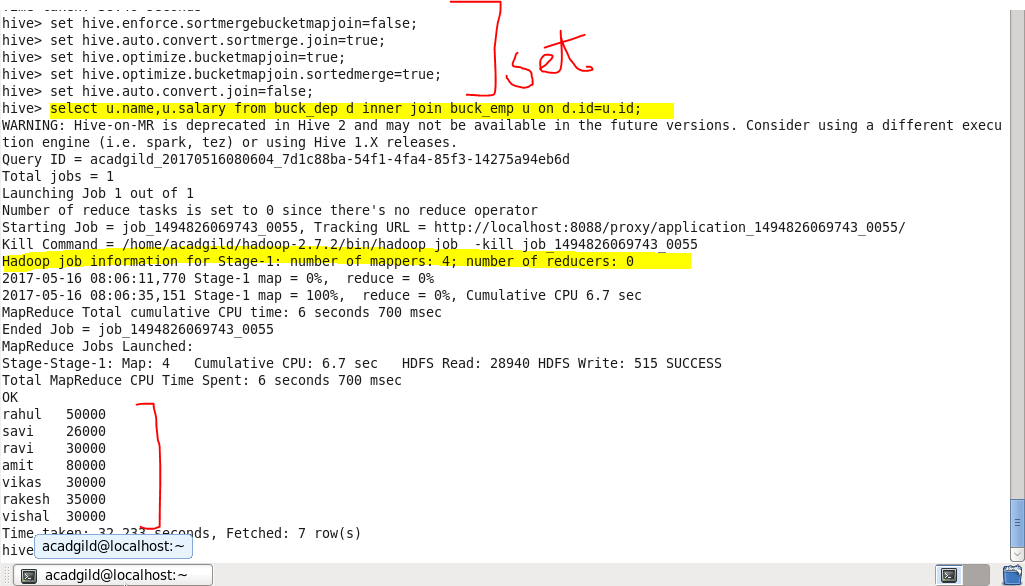
hive> set hive.auto.convert.sortmerge.join=true;

hive> set hive.optimize.bucketmapjoin = true;

hive> set hive.optimize.bucketmapjoin.sortedmerge = true;

hive> set hive.auto.convert.join=false;  // if we do not do this, automatically Map-Side Join will happen

SELECT u.name,u.salary FROM buck\_dept d  INNER JOIN buck\_emp u ON d.id = u.id;



See highlighted you will find that 4 mapper tasks are running (as we had 4 buckets). This helps in performing faster join operation when compared to regular.