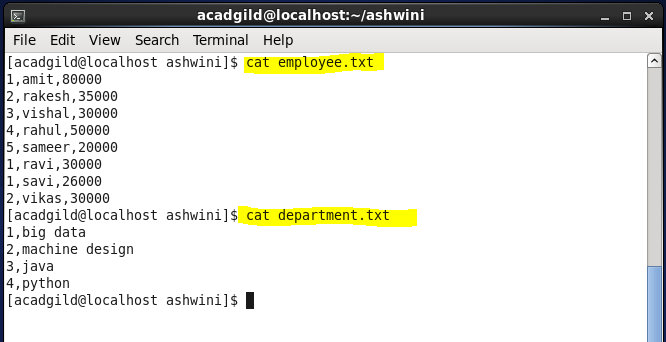


**Solution-**

* Hive, like other SQL databases, allows users to join various tables.
* However, Joins can be computationally expensive, especially on big tables.
* Hive on top of Hadoop makes data processing so straightforward and scalable that we can easily forget to optimize our Hive queries.
* For join optimization in Hive, we can use repartition joins, replication joins and semi joins.

**Example-**

**Input File-**

****

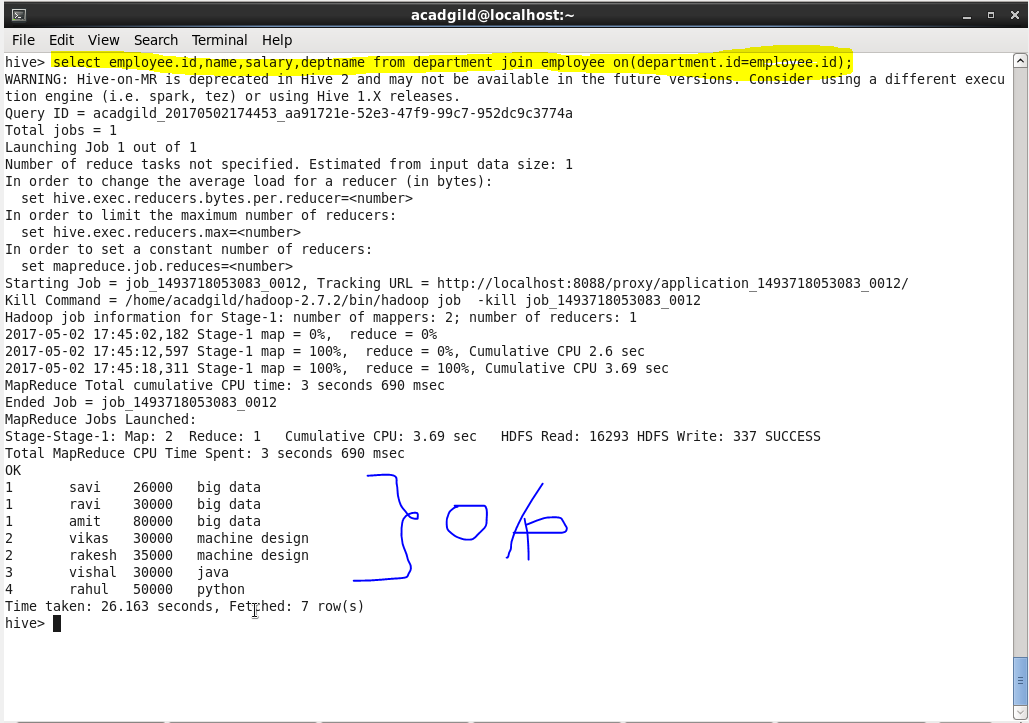
* **Creation of tables-**

****

**Join Table Ordering-**

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.

* In this example we have two tables employee and department. We know the employee table is not static, it will grow by the time, but the department table will be static for most of the time. Hence when these two tables are joined, it is important that the larger table comes last in the query.

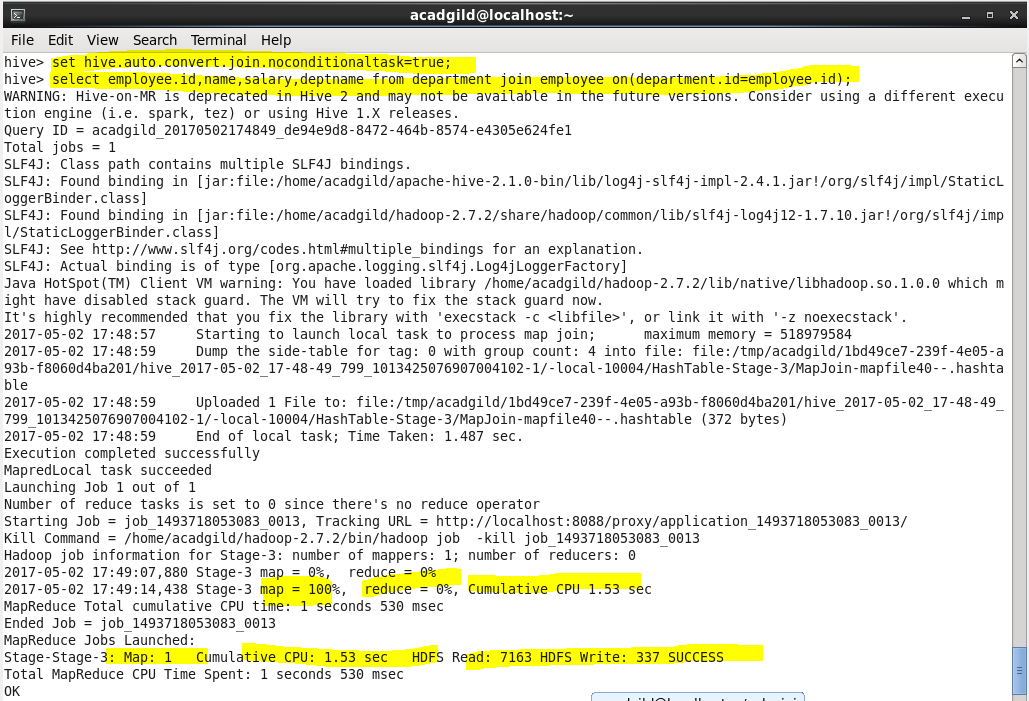


**Map Side Join-**

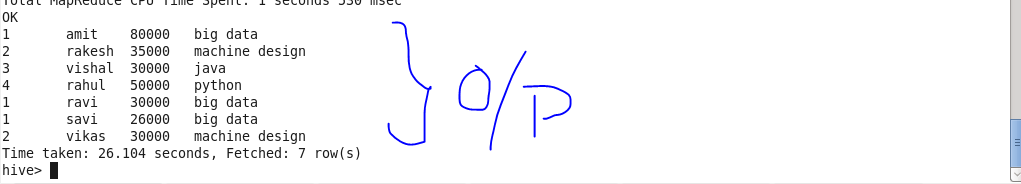
A map-side join is a special type of join where a smaller table is loaded in memory and join is performed in map phase of Map Reduce job. Since there is no reducer involved in the map-side join, it is much faster when compared to regular join.

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.

Once all the configurations are done, execute the same join operation as we performed. Since there is no reducer, the join operation will become faster.



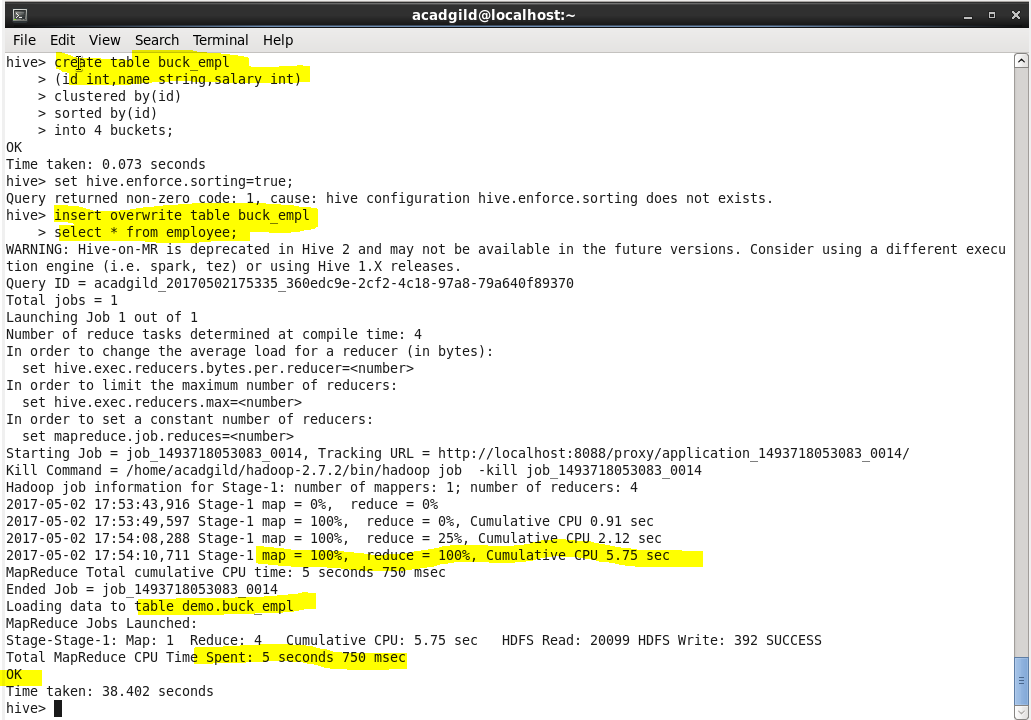
**Output-**

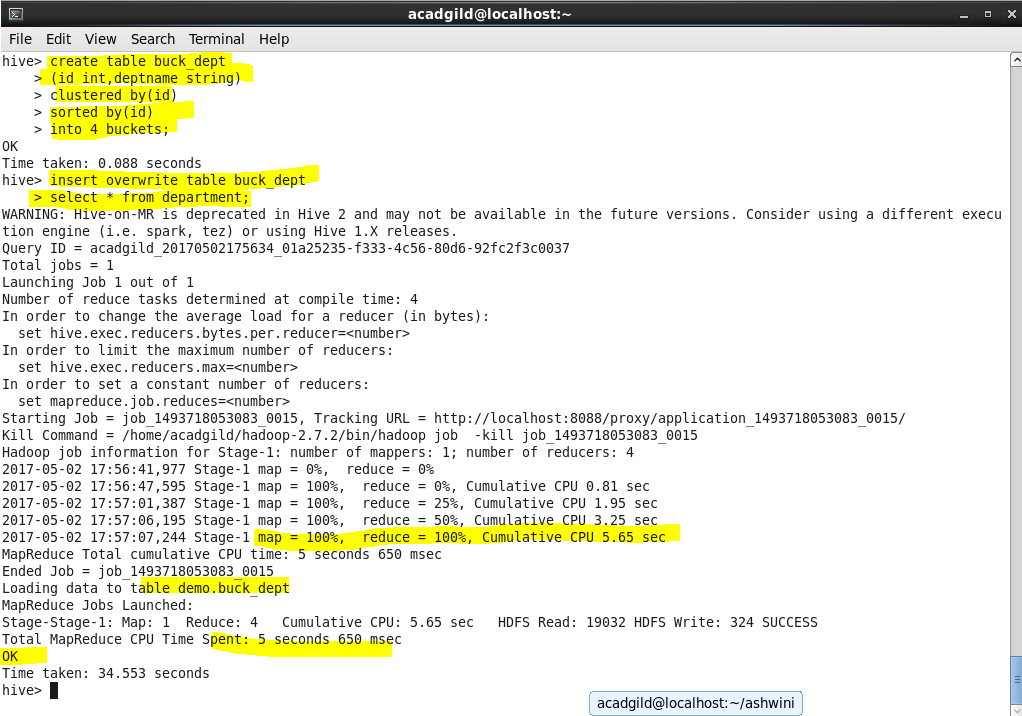


**Sort-Merge-Bucket Join-**

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.

* Creation of two bucket tables for employee and for department, which is clustered and sorted by id and divided into 4 buckets.
* Setting hive properties for bucketing.
* Loading data into bucketed tables.





* Select query to select data from bucketed tables by joining the two above created tables.

