**ASSIGNMENT - 27.5**

**Problem Statement:**

Perform join optimizations in hive by following the steps in the below blog link:

<https://acadgild.com/blog/join-optimization-in-apache-hive/>

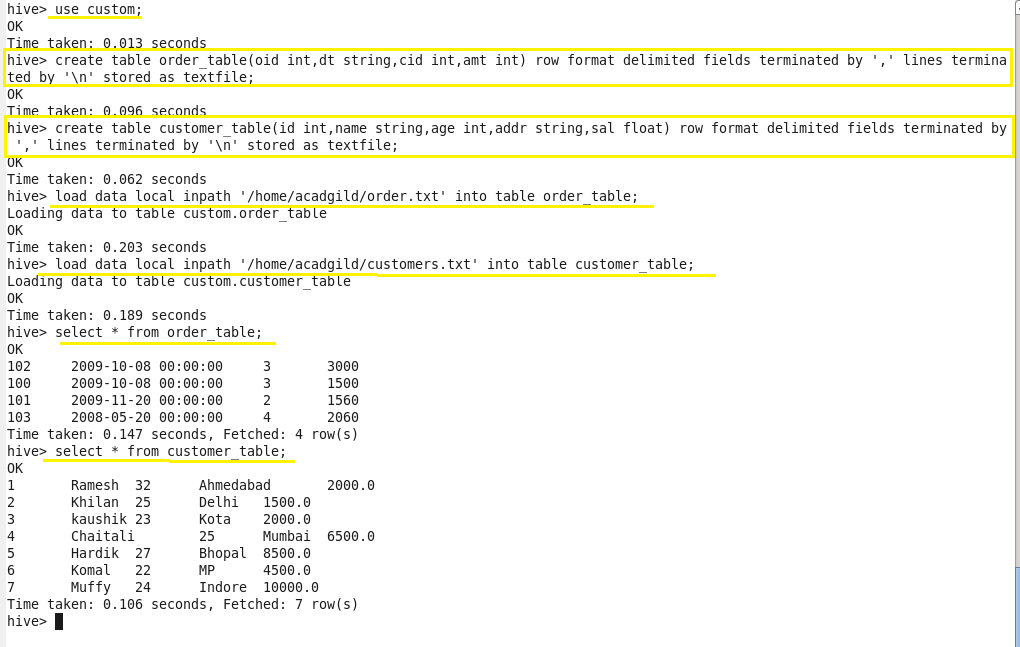
**Output:**

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.

**Table creation:**

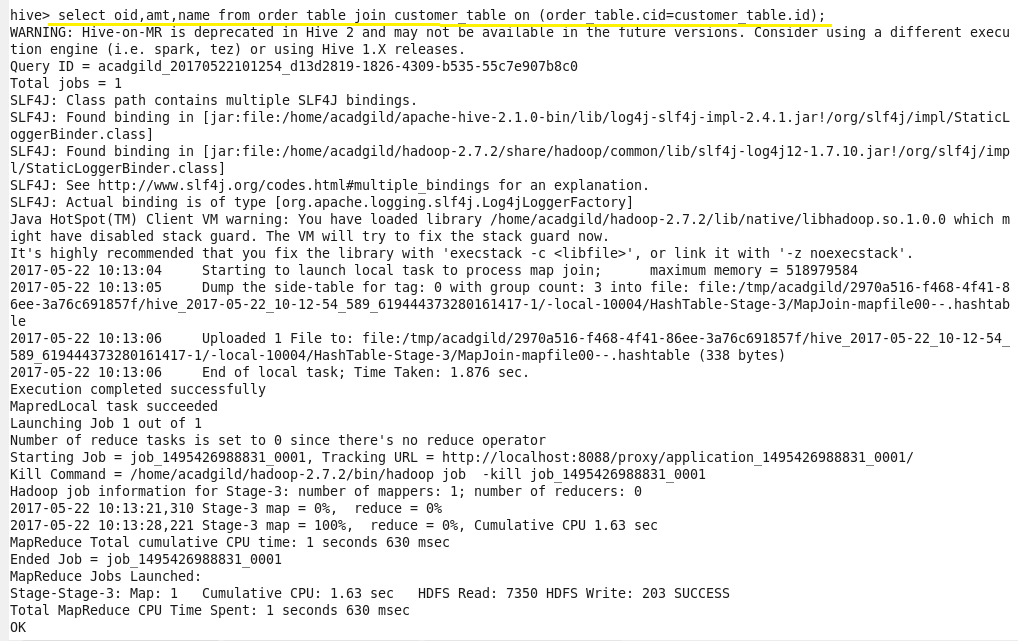


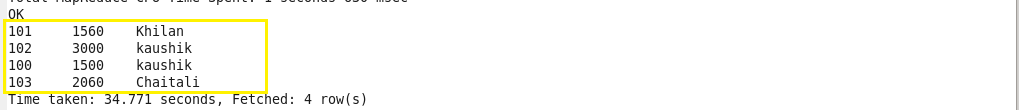
**Join table ordering (Largest table last):**

As with any type of tuning, it is important to understand the internal working of a system. 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.

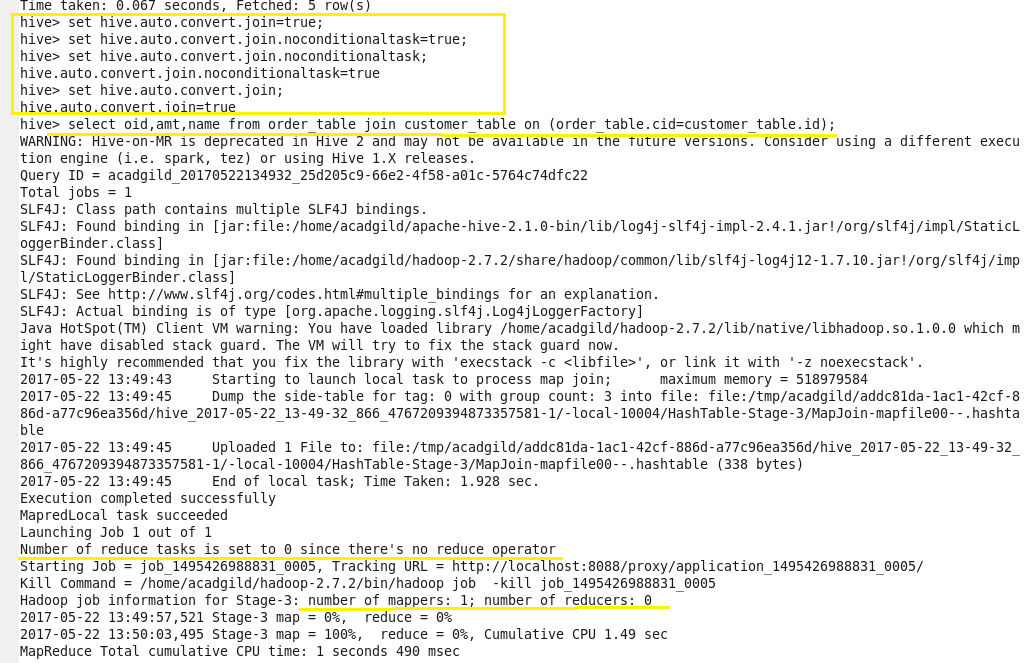
When two tables are joined it is important that the larger table comes last in the query.





**Map Side Join:**

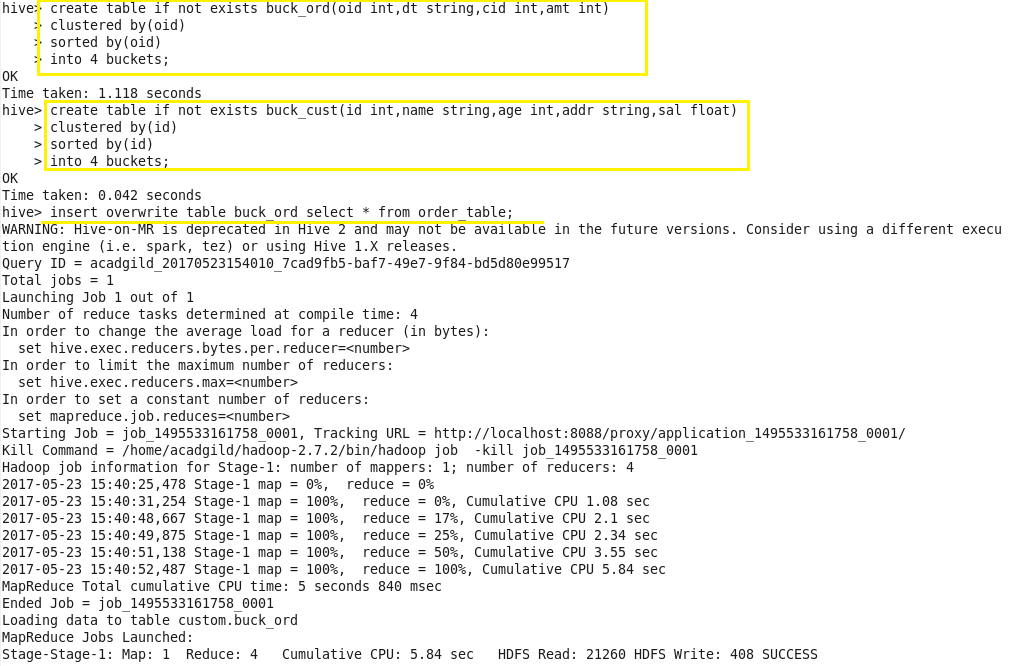
Also known as replicated join, a map-side join is a special type of join where a smaller table is loaded in memory and join is performed in map phase of MapReduce job. Since there is no reducer involved in the map-side join, it is much faster when compared to regular join.

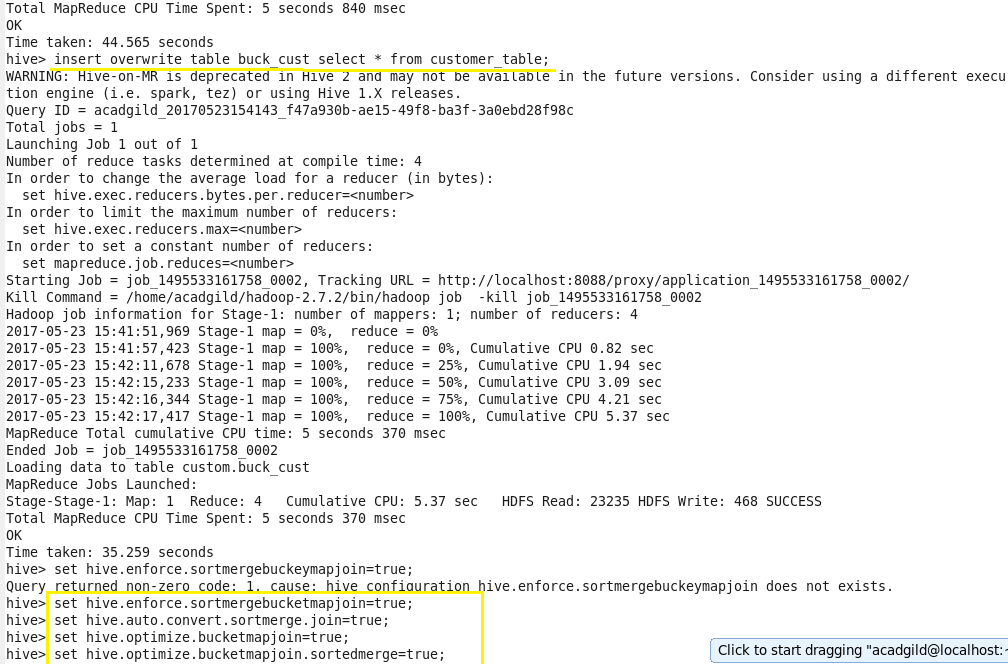


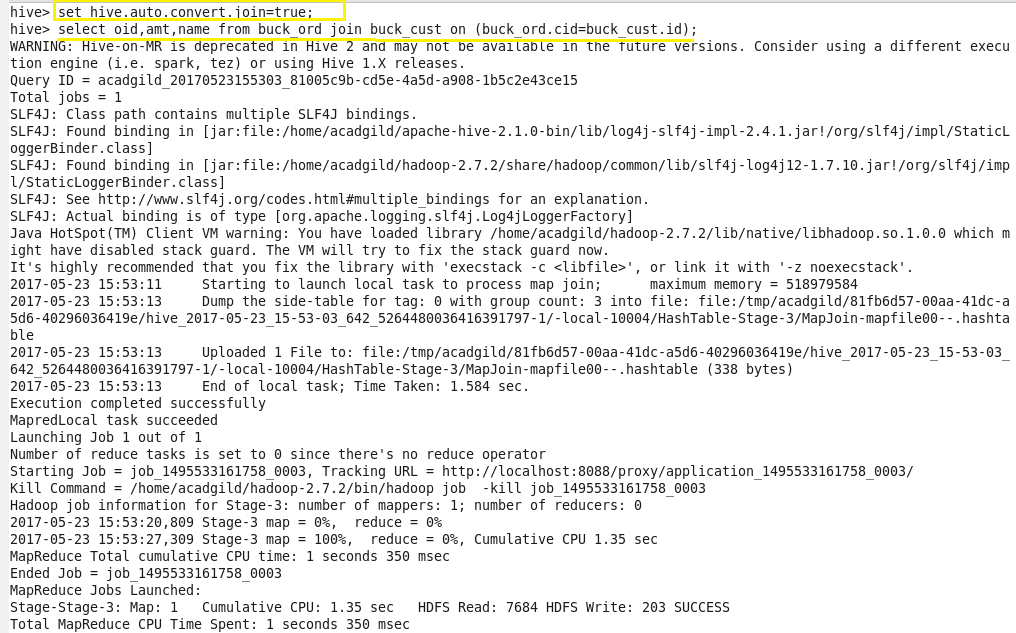


**Sort-Merge-Bucket (SMB) Map Join:**

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.









This helps in performing faster join operation when compared to regular joins.