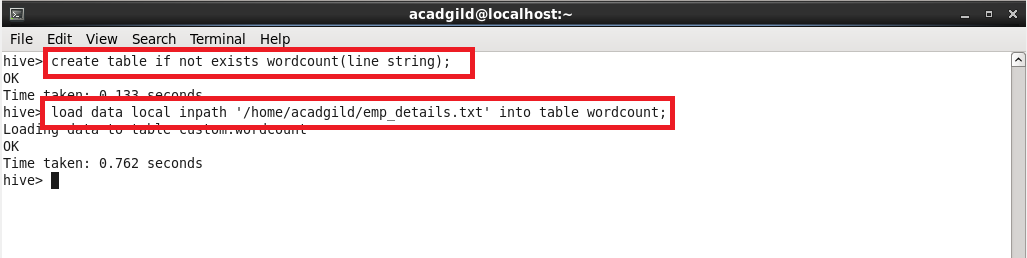
**ASSIGNMENT 26.1**

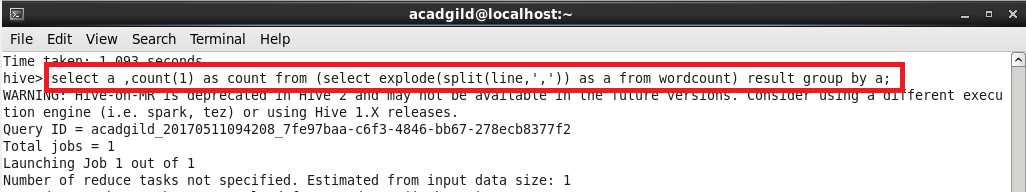
**link: WC\_emp\_details**

**● Perform word count in Hive for above given dataset.**

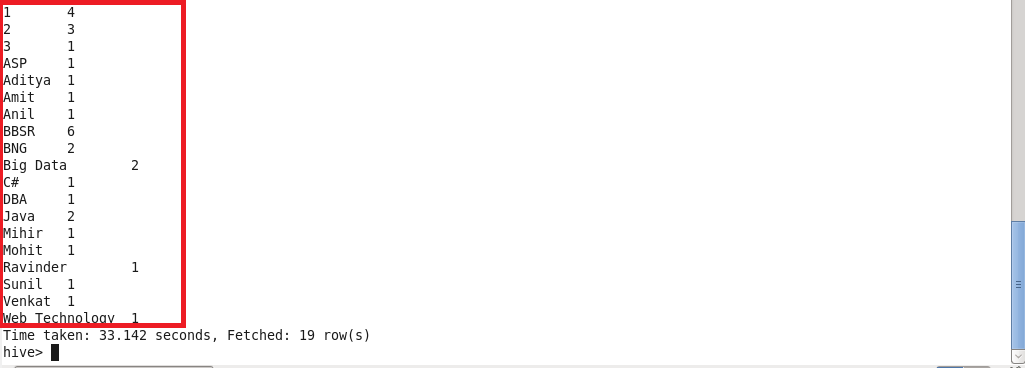
*Step 1 : creating a table ‘wordcount’ with single column and loading data into the table.*



*Step 2: split will split the data with comma delimiter and explode will arrange the words in multiple rows*



*Step 3: Output*



**Explain the working of Partitioning in brief.**

A simple query in Hive reads the entire dataset even if we have where clause filter. This becomes a bottleneck for running MapReduce jobs over a large table. We can overcome this issue by implementing partitions in Hive. Hive makes it very easy to implement partitions by using the automatic partition scheme when the table is created.

In Hive’s implementation of partitioning, data within a table is split across multiple partitions. Each partition corresponds to a particular value(s) of partition column(s) and is stored as a sub-directory within the table’s directory on HDFS. When the table is queried, where applicable, only the required partitions of the table are queried, thereby reducing the I/O and time required by the query.

**When to use Hive Partitioning:**

When any user wants data contained within a table to be split across multiple sections in hive table, use of partition is suggested.

The entries for the various columns of dataset are segregated and stored in their respective partition. When we write the query to fetch the values from table , only the required partitions of the table are queried, which reduces the time taken by query to yield the result.

**ADAVANTAGES:**

* Partitioning is used for distributing execution load horizontally.
* As the data is stored as slices/parts , query response time will be faster to process the small part of data instead of looking for a search in the entire data set.

**DISADVANTAGES:**

* Having too many partitions in the table creates large number of files and directories in hdfs, which is an overhead to namenode since it must keep all metadata for the filesystem in memory only
* Partitions may optimize some queries based on where clauses , but may be less responsive for other important queries on grouping clauses.

There are two types of partitioning in hive .They are

* Static partitioning.
* Dynamic partitioning.

Static partitioning needs to be applied when we know data(supposed to be inserted) belongs to which partition.

Dynamic partition can create as many number of partitions with single hive statement.

**Explain the difference between Static and Dynamic Partitioning in Hive with an example.**

**STATIC PARTITION:**

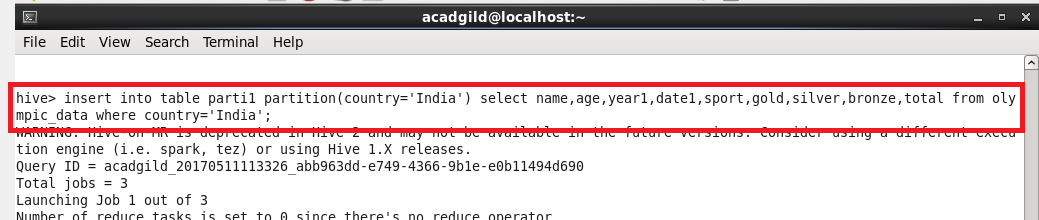
Static partitioning needs to be applied when we know data(supposed to be inserted) belongs to which partition.

In static partitioning we need to specify the partition column value in each and every LOAD statement.

If we know the data in the columns very well, then we can go for static partitioning.

* Insert input data files individually into a partition table is Static Partition
* Usually when loading files (big files) into [Hive tables](http://www.hadooptpoint.com/hive-create-table-examples/) static partitions are preferred
* Static Partition saves your time in loading data compared to dynamic partition
* You “statically” add a partition in table and move the file into the partition of the table.
* We can alter the partition in static partition.
* You can get the partition column value form the filename, day of date etc., without reading the whole big file.
* If you want to use Static partition in hive you should set property **set hive.mapred.mode = strict** This property set by default in hive-site.xml.
* Static partition is in Strict Mode.
* You should use where clause to use limit in static partition.
* You can perform Static partition on Hive Manage table or external table.
* In static partitioning every partitioning needs to be backed with individual hive statement which is not feasible for large number of partitions as it will require writing of lot of hive statements.

During inserting we should specify the value of the column based on which we have partitioned.



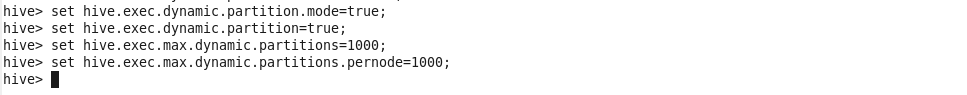
**DYNAMIC PARTITION:**

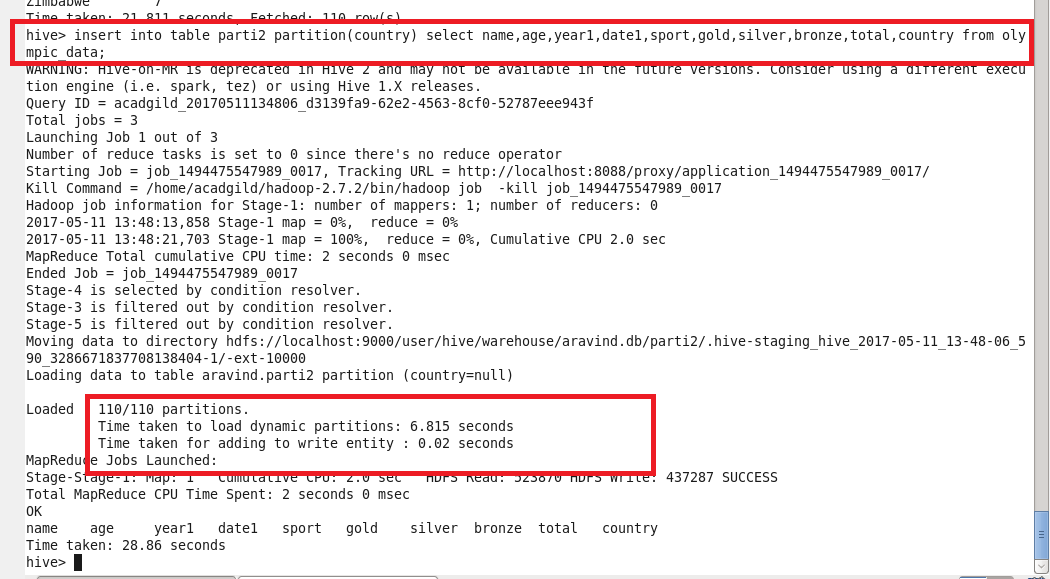
**In Dynamic partitioning** every row of the data is read and data is partitioned through a MR job into the destination tables depending on certain field in file.

* single insert to partition table is known as dynamic partition
* Usually dynamic partition load the data from non partitioned table
* Dynamic Partition takes more time in loading data compared to static partition
* When you have large data stored in a table then Dynamic partition is suitable.
* If you want to partition number of column but you don’t know how many columns then also dynamic partition is suitable
* Dynamic partition there is no required where clause to use limit.
* we can’t perform alter on Dynamic partition
* You can perform dynamic partition on hive external table and managed table
* If you want to use Dynamic partition in hive then mode is in nonstrict mode

For dynamic partition we need to set some commands in shell

They are,





During inserting in the partitioned table we need not to specify the value of the column based on which we have partitioned .