**Assignment 26.1**

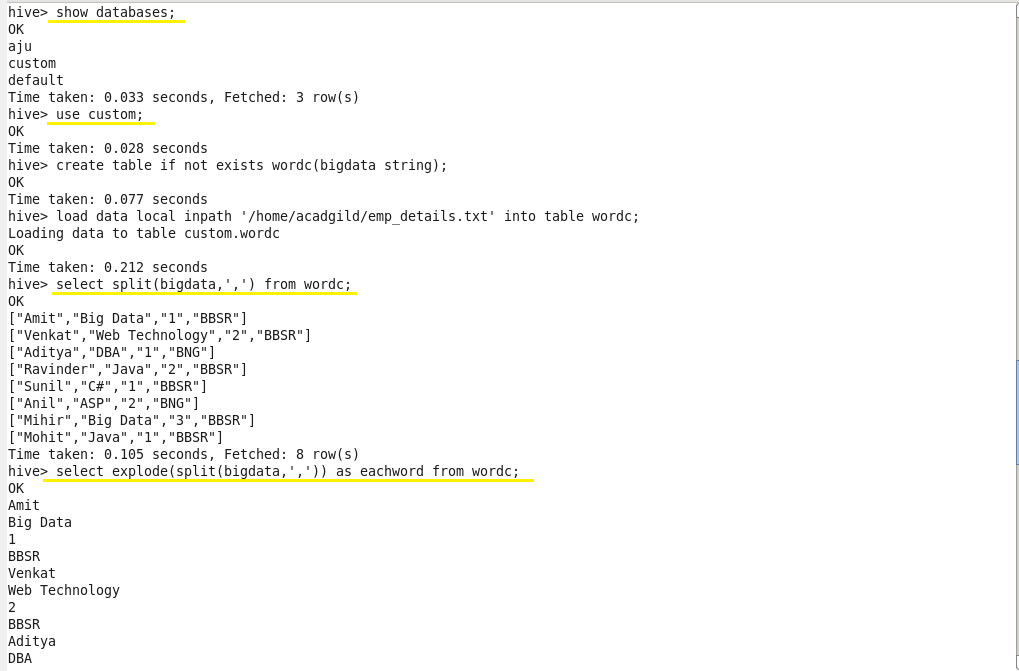
**Problem Statement 1:**

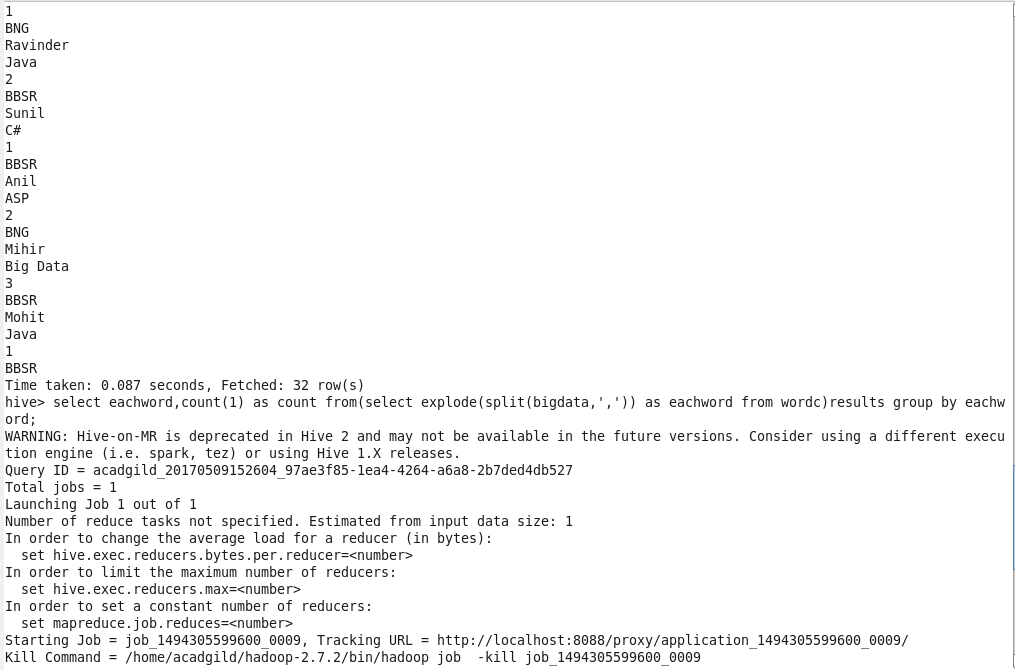
Link:

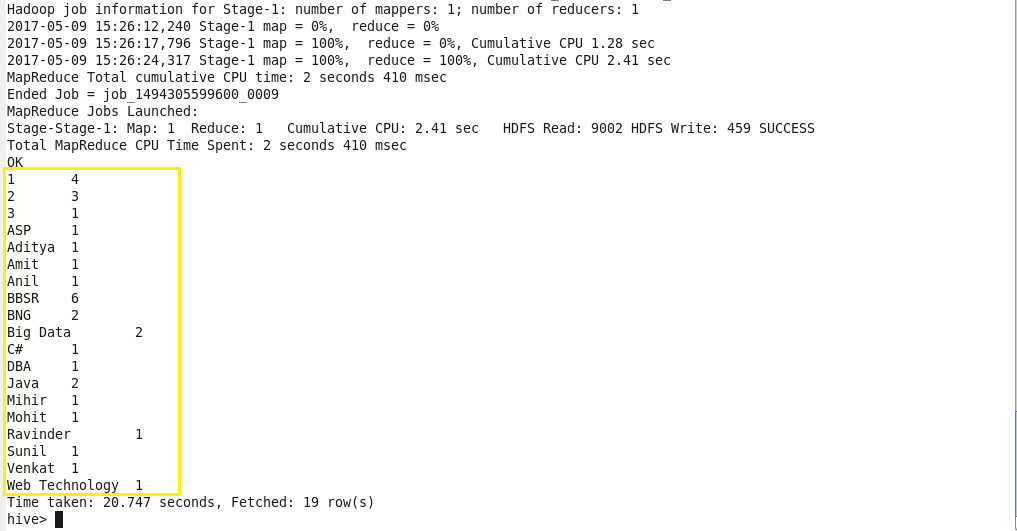
<https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/WC_emp_details.txt>

Perform word count in Hive for above given dataset.

**Output:**

****

****

****

**Problem Statement 2:**

● Explain the working of Partitioning in brief.

**Answer:**

* **Working of Partitioning:**

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.

Partitioning in hive is dividing table data into some parts based on the values of particular columns like date or country, segregate the input records into different files/directories based on date or country.

Hive organizes tables into partitions. Using partition, it is easy to query a portion of the data.

**For example,**

The first thing we need to do is create a partitioned table with the PARTITIONED BY command.  What we’re saying here is that we want all the rows in a day, separated out in a separate directory and file(s). So when you issue Hive, it doesn’t have to scan an entire data set.

CREATE TABLE temps\_partition\_date (statecode STRING, countrycode STRING, sitenum STRING, paramcode STRING, poc STRING, latitude STRING, longitude STRING, datum STRING, param STRING, timelocal STRING, dategmt STRING, timegmt STRING, degrees double, uom STRING, mdl STRING, uncert STRING, qual STRING, method STRING, methodname STRING, state STRING, county STRING, dateoflastchange STRING) PARTITIONED BY (datelocal STRING);

We are inserting data from the temps\_txt table that we loaded in the previous examples.  The big difference here is that we are PARTITION’ed on datelocal, which is a date represented as a string.

E.g. “2014-01-01”.  The one thing to note here is that see that we moved the “datelocal” column to being last in the SELECT.  For dynamic partitioning to work in Hive, this is a requirement.

INSERT INTO TABLE temps\_partition\_date

PARTITION (datelocal)

SELECT statecode, countrycode, sitenum, paramcode, poc, latitude, longitude, datum, param, timelocal, dategmt, timegmt, degrees, uom, mdl, uncert, qual, method, methodname, state, county, dateoflastchange, datelocal

FROM temps\_txt;

We can see results using SELECT command as below:

SELECT \* FROM temps\_partition\_date WHERE degrees = 25.0 AND datelocal = '2014-01-01' AND county = 'India' AND state = 'TN';

SELECT \* FROM temps WHERE degrees = 25.0 AND datelocal = '2014-01-01' AND county = 'India' AND state = ‘TN’;

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

**Answer:**

* **Difference between Static and Dynamic Partitioning:**

In Hive There is two types of Partitions:

1. Static Partition in Hive
2. Dynamic Partition in Hive

* **Static Partition in Hive:**

Insert input data files individually into a partition table is Static Partition

Usually when loading files (big files) into Hive tables 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.

Usually when loading files (big files) into Hive tables static partitions are preferred. That 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. Since the files are big they are usually generated in HDFS.

You can get the partition column value form the filename, day of date etc. without reading the whole big file.

**For example,**

CREATE TABLE cityreport(cityid string,creport string, ctover string)

partitioned BY (city string)

row format delimited

FIELDS terminated BY ‘|’

stored AS textfile;

**Load Data in table by using Static Partition:**

hive> LOAD DATA LOCAL inpath ‘/home/mahesh/hive-related/hyderabad.log’ INTO TABLE cityreport partition (city = ‘hyderabad’);

* **Dynamic Partition in Hive:**

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 no strict mode

Here is hive dynamic partition properties you should allow.

In case of dynamic partition whole big file i.e. 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. So usually dynamic partition are useful when you are doing sort of an ETL flow in your data pipeline.

E.g. you load a huge file through a move command into a Table X. then you run an inert query into a Table Y and partition data based on field in table X say day, country. You may want to further run an ETL step to partition the data in country partition in Table Y into a Table Z where data is partitioned based on cities for a particular country only.

**For example,**

SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;

**Load data into Table by using Dynamic Partition:**

hive> INSERT INTO TABLE t2 PARTITION(country) SELECT \* FROM T1;