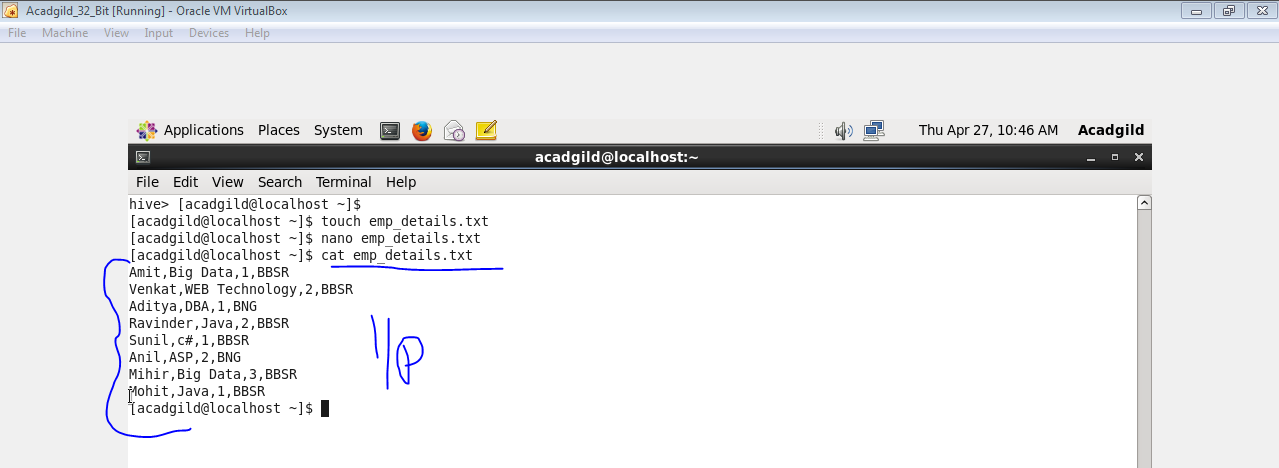
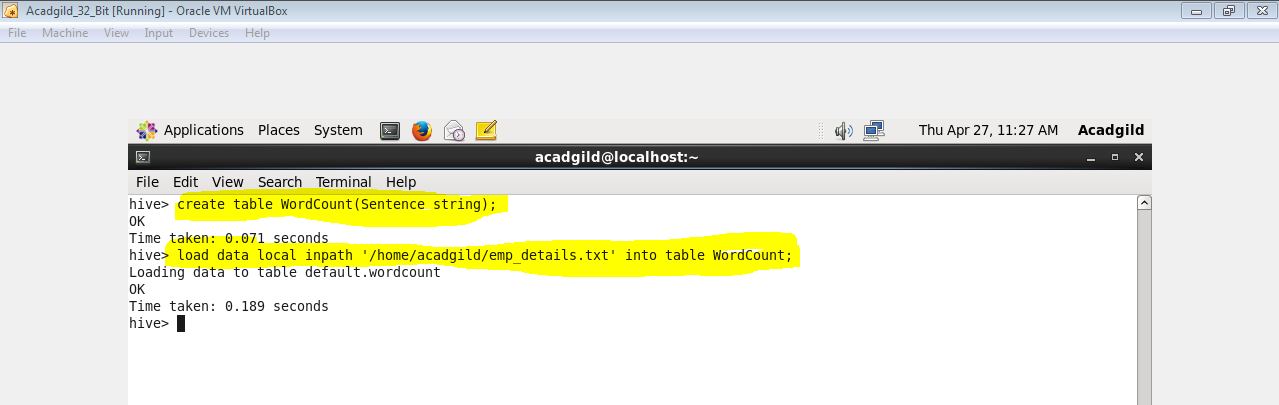
**Problem Statement 1:**

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

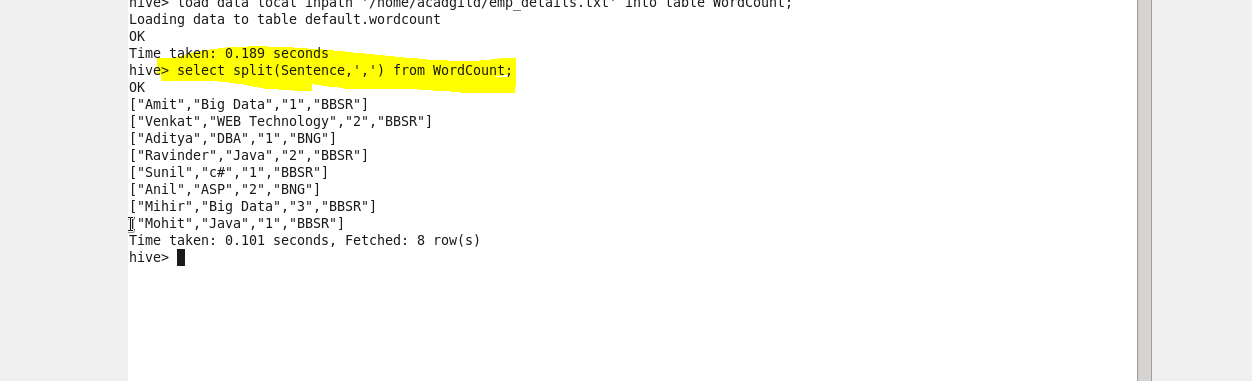
**INPUT FILE**



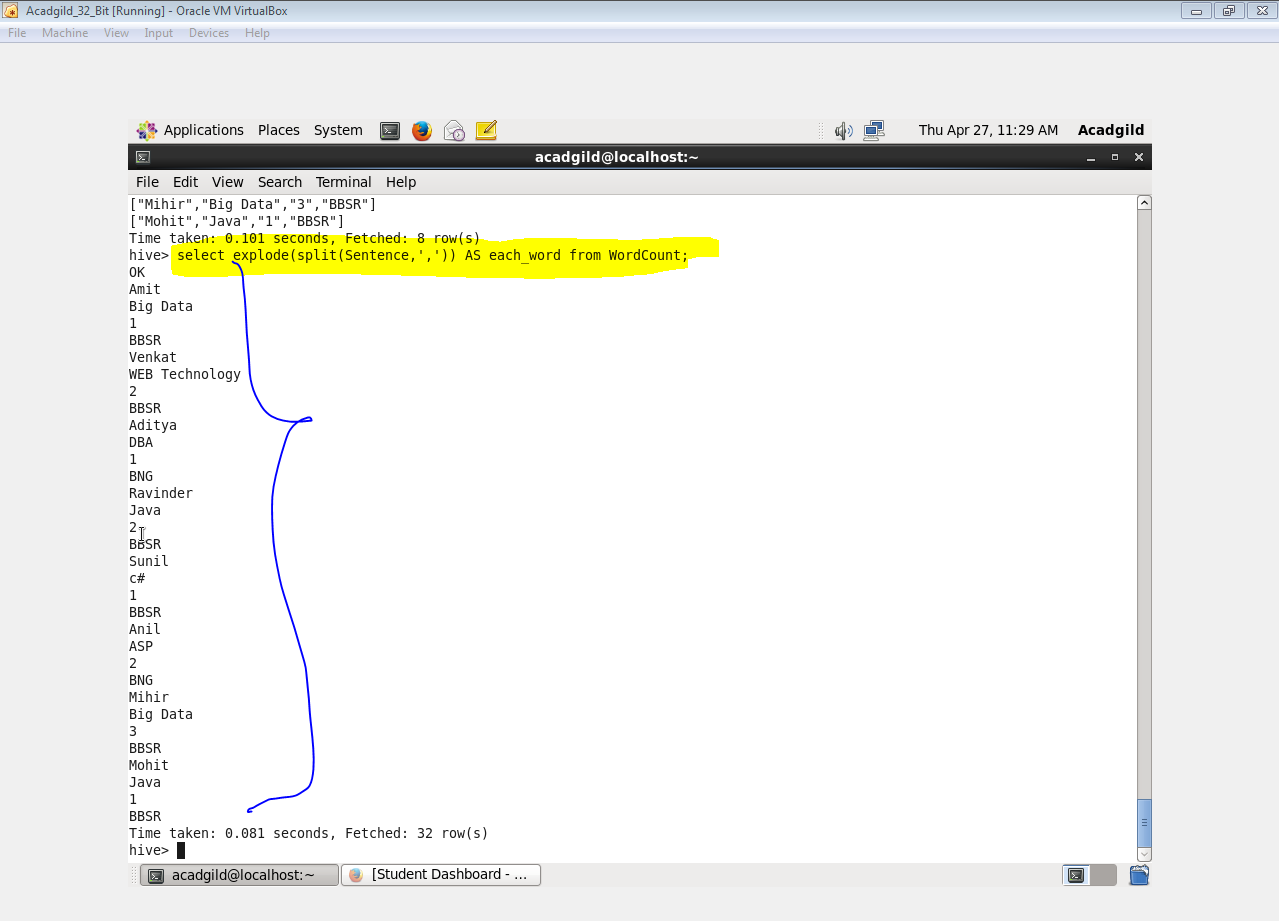
**CREATING A TABLE AND LOADING THE FILE**



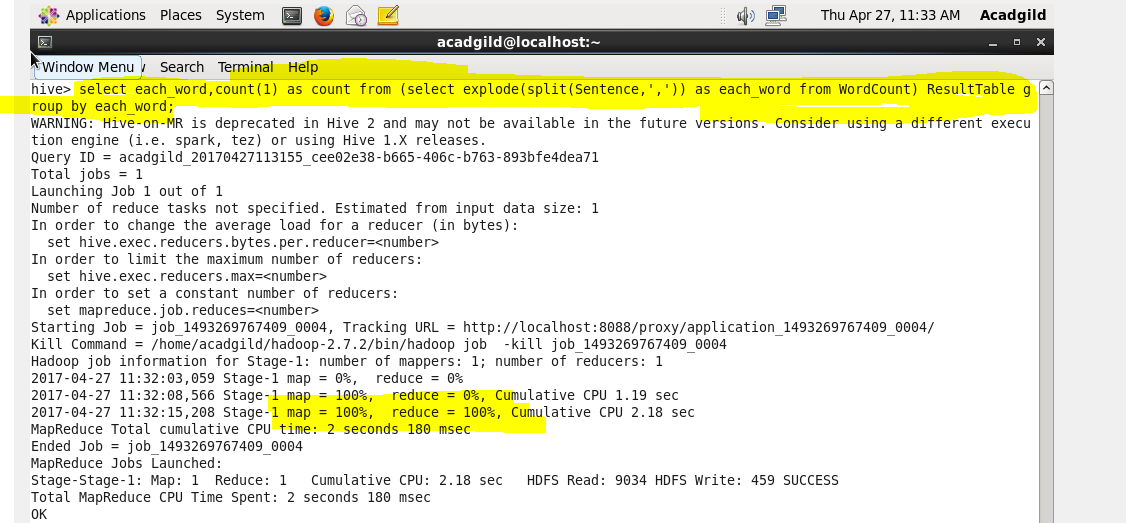
**SPLIT COMMAND**



**EXPLODE COMMAND**



**COUNTING EACH WORD**



**OUTPUT**



**Problem Statement 2:**

● Explain the working of Partitioning in brief.

**Hive Partitioning**

Hive organizes tables into partitions. It is a way of dividing a table into related parts based on the values of partitioned columns such as date, city, and department. Using partition, it is easy to query a portion of the data.

Tables or partitions are sub-divided into **buckets,** to provide extra structure to the data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.

For example, a table named **Tab1** contains employee data such as id, name, dept, and yoj (i.e., year of joining). Suppose you need to retrieve the details of all employees who joined in 2012. A query searches the whole table for the required information. However, if you partition the employee data with the year and store it in a separate file, it reduces the query processing time.

* **Load data into HDFS**
* **Create Partitioned hive table**

#### Insert data into Partitioned table, by using select clause

## Adding a Partition

We can add partitions to a table by altering the table. Let us assume we have a table called **employee** with fields such as Id, Name, Salary, Designation, Dept, and yoj.

**Syntax:**

ALTER TABLE table\_name ADD [IF NOT EXISTS] PARTITION partition\_spec

[LOCATION 'location1'] partition\_spec [LOCATION 'location2'] ...;

partition\_spec:

:(p\_column = p\_col\_value, p\_column = p\_col\_value, ...)

The following query is used to add a partition to the employee table.

hive> ALTER TABLE employee

> ADD PARTITION (year=’2013’)

> location '/2012/part2012';

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

**When to use static partitioning**

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

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

hive> LOAD DATA LOCAL INPATH '${env:HOME}/staticinput.txt'

INTO TABLE partitioned\_user

PARTITION (country = 'US', state = 'CA');

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.

**When to use dynamic partitioning**

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.

In that scenario dynamic partitioning is suggested as we can create as many number of partitions with single hive statement.

hive>INSERT INTO TABLE partitioned\_user

PARTITION (country, state)

SELECT firstname ,lastname ,address , city ,post ,phone1 ,phone2 ,email ,web ,country, state

FROM temp\_user;