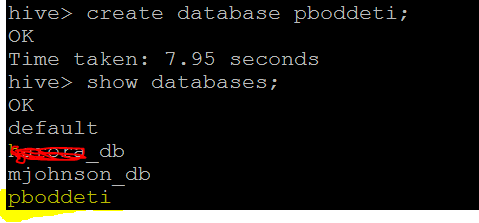
Hive Partitions and Bucketing:

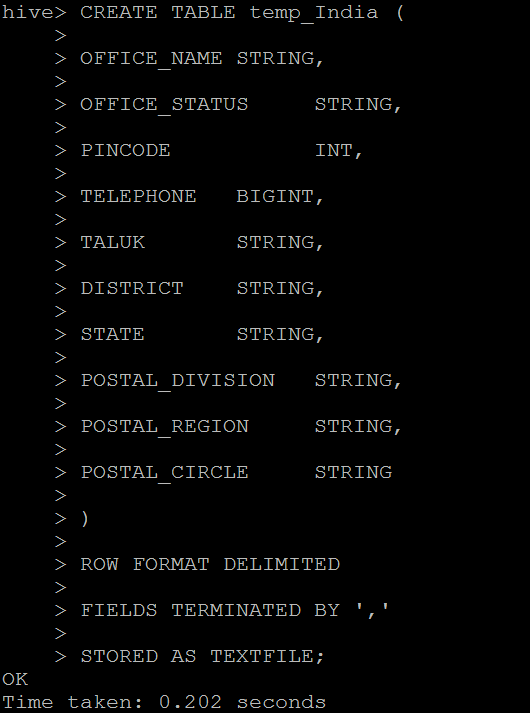
## Partitioning

We can use partitioning feature of Hive to divide a table into different partitions. Each partition of a table is associated with a particular value(s) of partition column(s). Partitioning allows Hive to run queries on a specific set of data in the table based on the value of partition column used in the query.

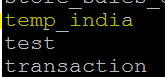
Without partitioning, any query on the table in Hive will read the entire data in the table.  
If we have a large table then queries may take long time to execute on the whole table. We can make Hive to run query only on a specific partition by partitioning the table and running queries on specific partitions. A table can be partitioned on columns like – city, department, year, device etc.

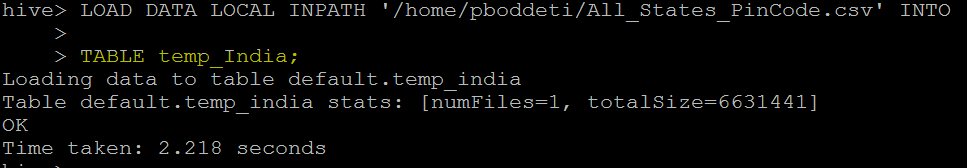


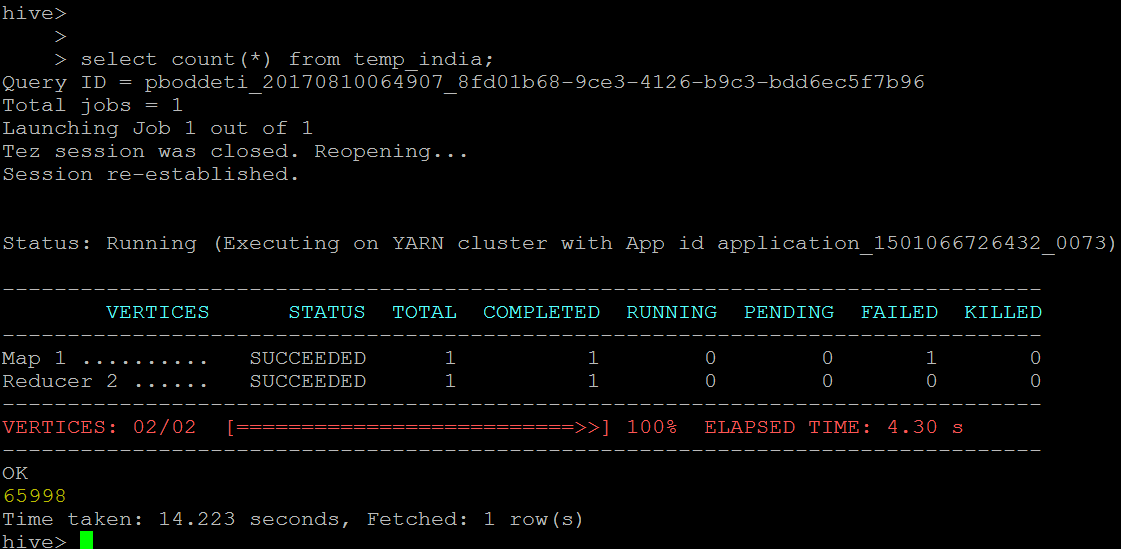
Dataset :





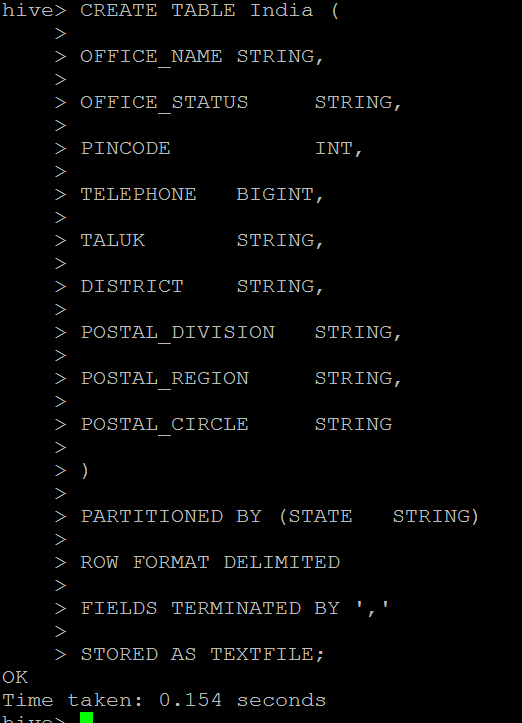


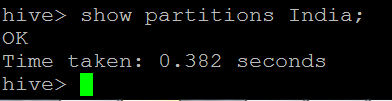




Next important thing to understand about hive is partitioned table. Let’s assume we are a global company and we have millions of customers across the globe. We have large data available about our customers from all over the world and we want to load it into a data warehouse. Combined data file is of many terabytes or may a couple of petabytes. You can imagine hod your quries will perform in such situations if you have not partitioned your table smartly and your queries does not include appropriate partition filters. If you have any experience in writing database queries on very large databases, probably you are already aware with such problems. This makes partitioning an basic feature for any database table no matter its apache hive or any other database. However, hive implementation of table partitioning is completely different than most of other databases. We will understnd it using an example. I will use very small data size for this dimonstration but to explain concept behind it, size does not matter.

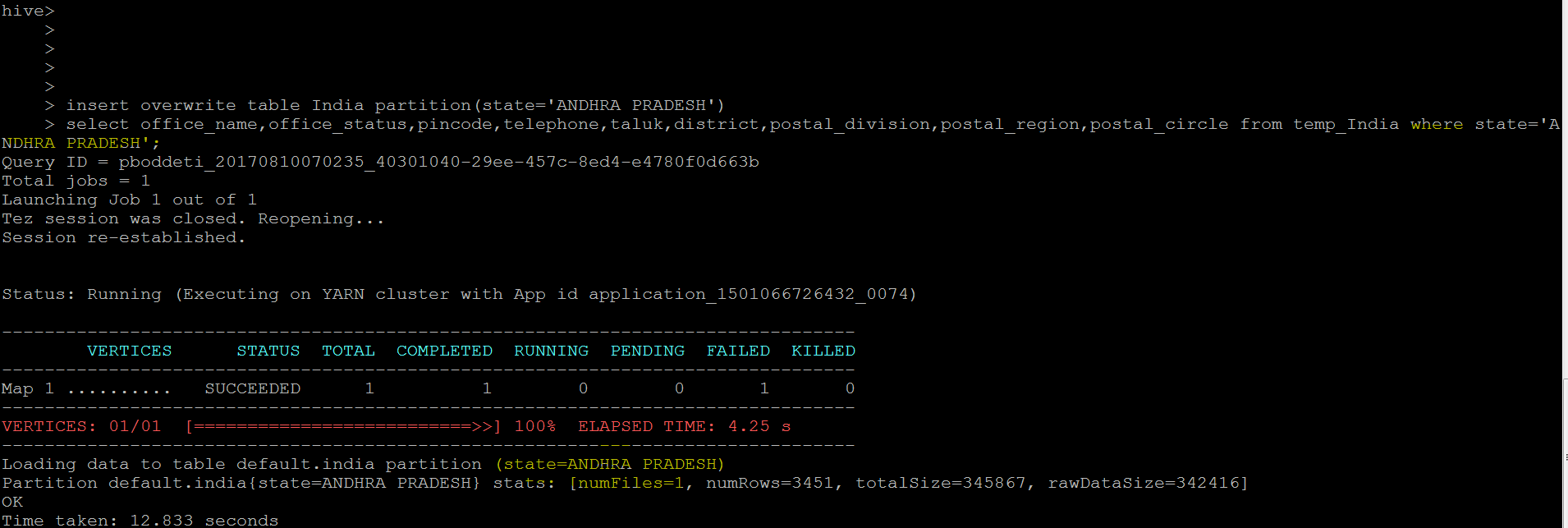
Let’s create a customers table which is partitioned on two keys, country and state. Syntax for creating partioned table is shown in screen below. You will notice that I haven’t created any column for country and state. There is a reason for this. Since country and state information is encoded into my partition, why do I need to repeat that data into table? That’s redundent data so we do not create country and state columns in our table. Any partition itself will be able to provide that information and all data within specific partion belongs to same value.  
After creating a partitioned table, I loaded data into this table for one partition only i.e. STATE=ANDHRA PRADESH . There are two things to note here. First, I need to tell hive which partition I am loading using **partition(state=’ANDHRA PRADEESH’)**clause. Second, my table India will only have data for State/AP. Data is loaded and now you are readu to execute your queries on this table. Partitioned table does not place any restriction on querying data, you can still execute your queries for entire world, but if you query for specific partition, hive need not to read all partitions making your query faster.

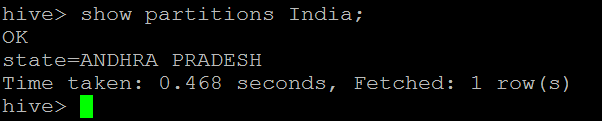


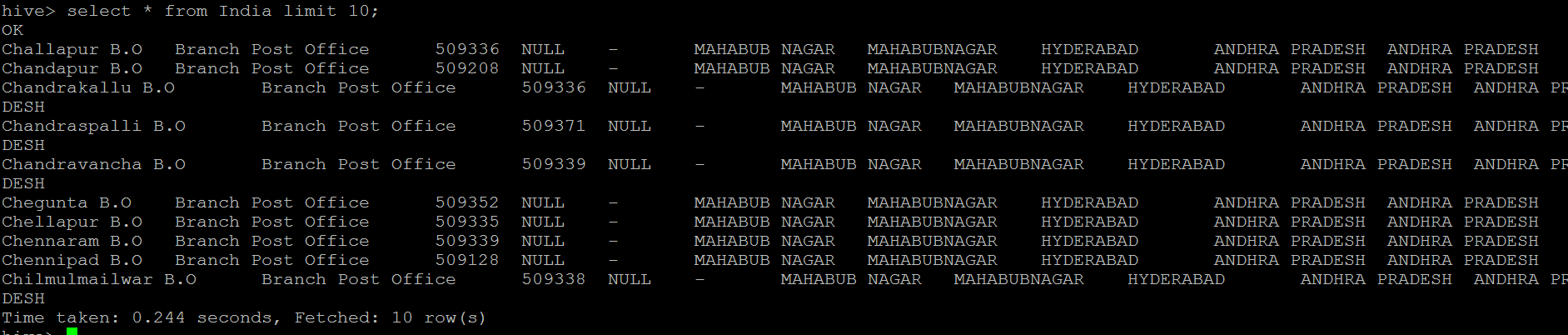


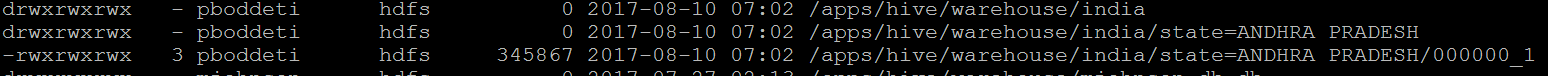
insert overwrite table India partition(state='ANDHRA PRADESH')

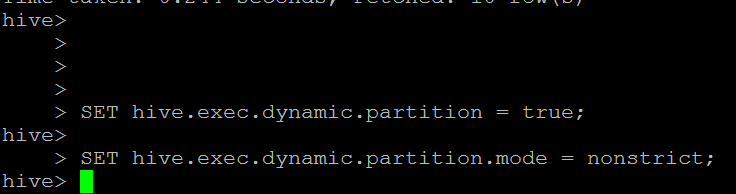
select office\_name,office\_status,pincode,telephone,taluk,district,postal\_division,postal\_region,postal\_circle from temp\_India where state='ANDHRA PRADESH';











Let’s move further and learn dynamic partitioning. In my external table, I had six combinations of state. If I want to load all of the data into their respective partitions of my hive table, I have to write a long statement. This situation could be more daunting if I have a file with 100 such combinations resulting into several partitions. That’s where dynamic partitioning comes to rescue.

**Dynamic Partition:**

1.columns whose values are only known at EXECUTION TIME.

2.We use dynamic partition while loading from an existing table that is not partitioned.

3.We use dynamic partition while unknown values for partition columns.

4.Usually dynamic partition load the data from non partitioned table.

5.Dynamic Partition takes more time in loading data compared to static partition.

6.There is no required where clause to use limit.

7.We can’t perform alter on Dynamic partition.

8.Perform dynamic partition on hive external table and managed table.

***In dynamic partition the partitioned column of the partitioned hive table is must present in the last column of the existing hive table.***

Dynamic partitioning in hive is controlled by configuration variables. I have used hive **SET** command to display current values of these variables.

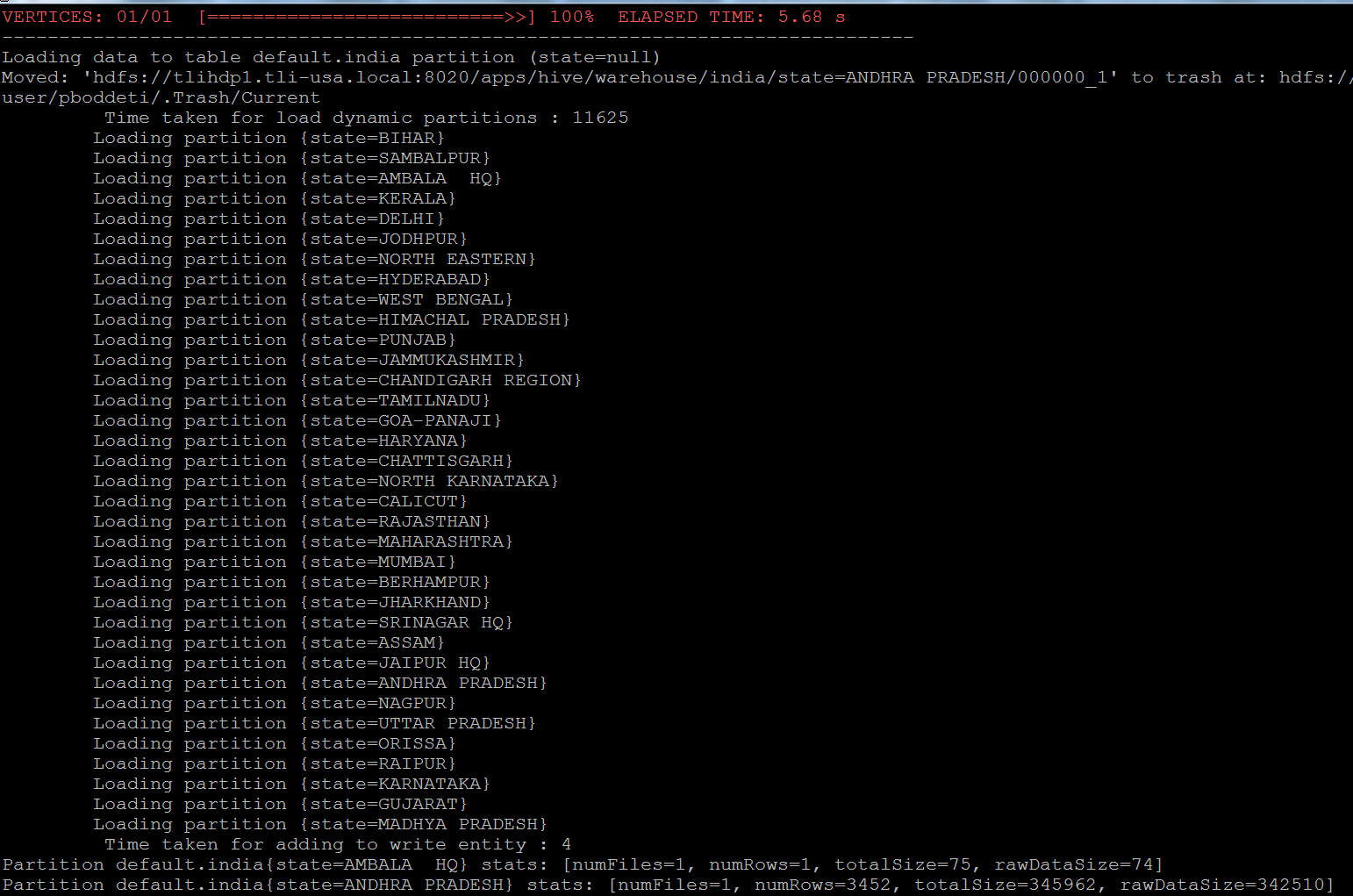
All of above variables are briefly explained into table below. Their default values are alredy displayed in above screen.

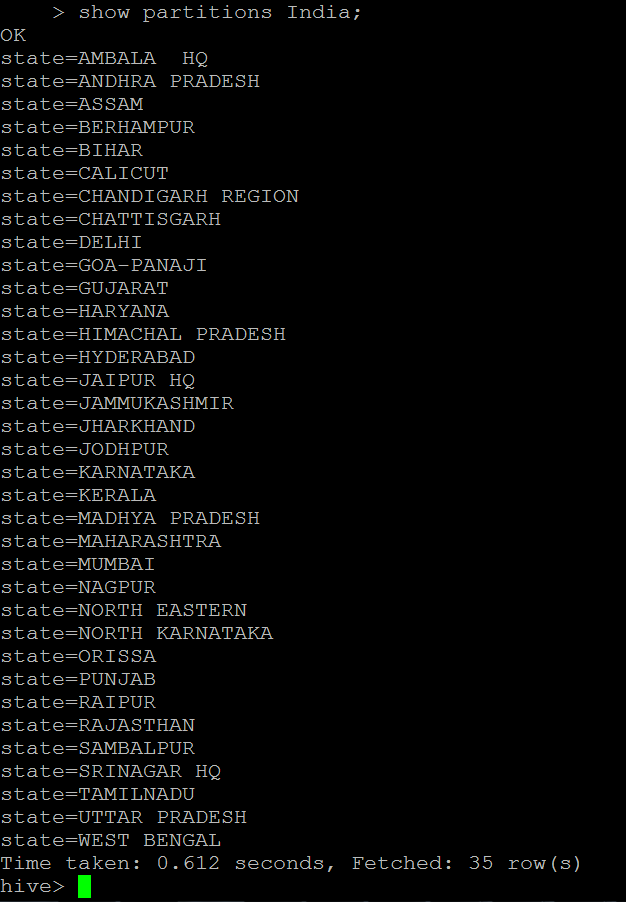
|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| hive.exec.dynamic.partition | true/false for enable/disable dynamic partitioning |
| hive.exec.dynamic.partition.mode | strict/nonstrict as explained below. |
| hive.exec.max.dynamic.partitions.pernode | Max number of partitions each mapper/reducer can create. |
| hive.exec.max.dynamic.partitions | Max number of partitions allowed by single statement. |
| hive.exec.max.created.files | Max number of files that can be created globally. |

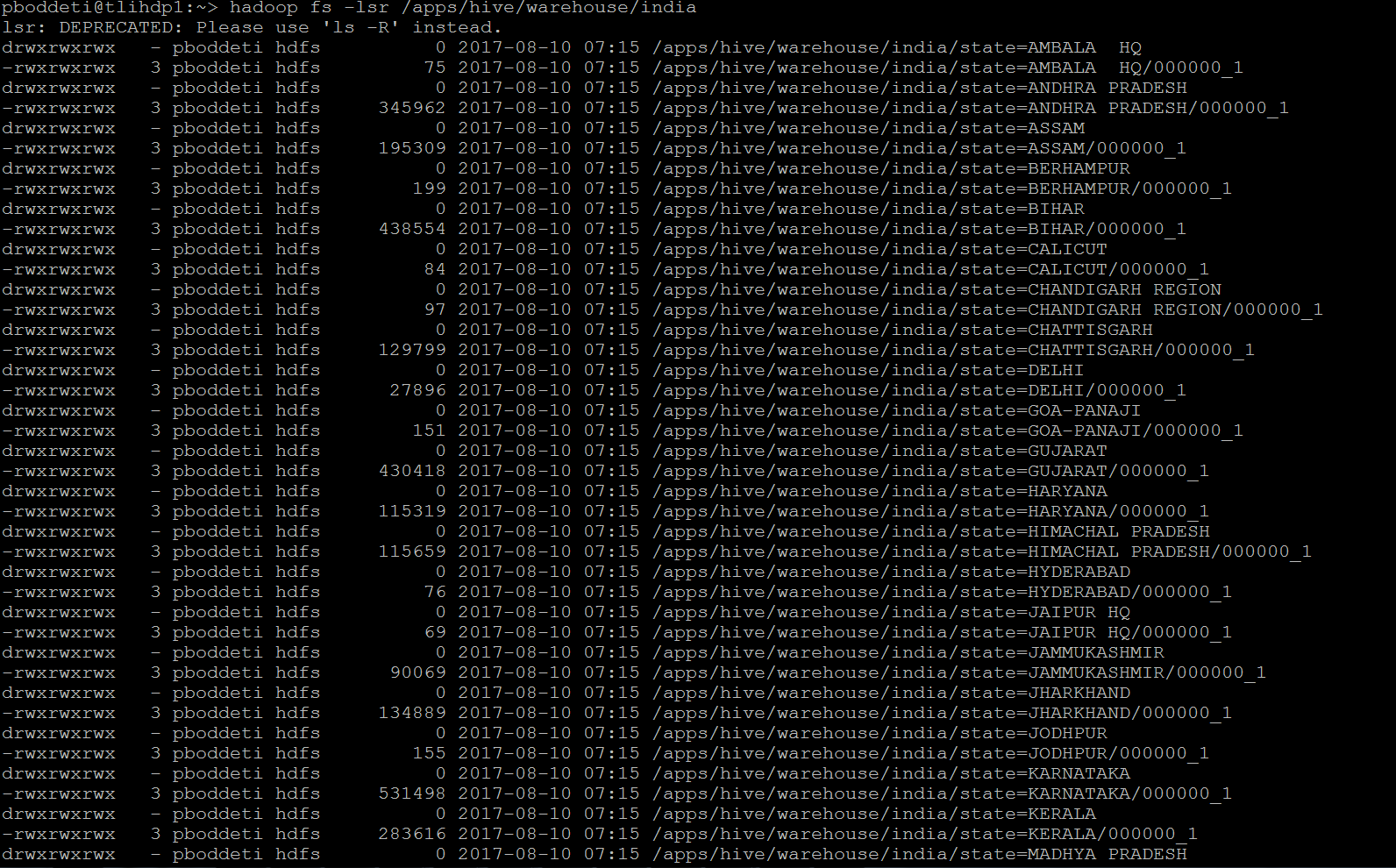
insert overwrite table India partition(state)

select \* from temp\_India;

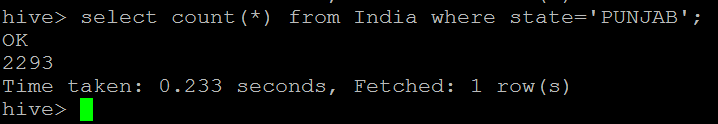








select count(\*) from india where state=’Punjab’;



[Hive Bucketed Tables and Sampling](http://thriveschool.blogspot.in/2013/11/hive-bucketed-tables-and-sampling.html)

Bucketing is a simple idea if you are already aware. You create multiple buckets. You read each record and place it into one of the buckets based on some logic mostly some kind of hashing algorithm. This allows you to organize your data by decomposing it into multiple parts. You might think that we do achieve same thing using partitioning then why bother about bucketing. There is one difference. When we do partitioning, we create a partition for each unique value of the column. This may burst into a situation where you might need to create thousands of tiny partitions. But if you use bucketing, you can limit it to a number which you choose and decompose your data into those buckets. In hive a partition is a directory but a bucket is a file. We will see it in action.

In hive, bucketing does not work by default. You will have to set following variable to enable bucketing. **set hive.enforce.bucketing=true;**Once you ensured that above variable is set, lets create a bucketed table. In my previous post, we had an external table named employees. We will create a bucketed hive table named emphive and load data from employees table. Once data is loaded into bucketed table, we will go back to our warehouse to check if we have multiple data files created which should be equal to the number of buckets in our table.  
Here is command to create a bucketed file. I have bucketed data on STATE column into 3 buckets.

**set hive.enforce.bucketing=true;**

CREATE TABLE Buck\_India (

OFFICE\_NAME STRING,

OFFICE\_STATUS STRING,

PINCODE INT,

TELEPHONE BIGINT,

TALUK STRING,

DISTRICT STRING,

STATE STRING,

POSTAL\_DIVISION STRING,

POSTAL\_REGION STRING,

POSTAL\_CIRCLE STRING

)

CLUSTERED BY (STATE) INTO 3 BUCKETS

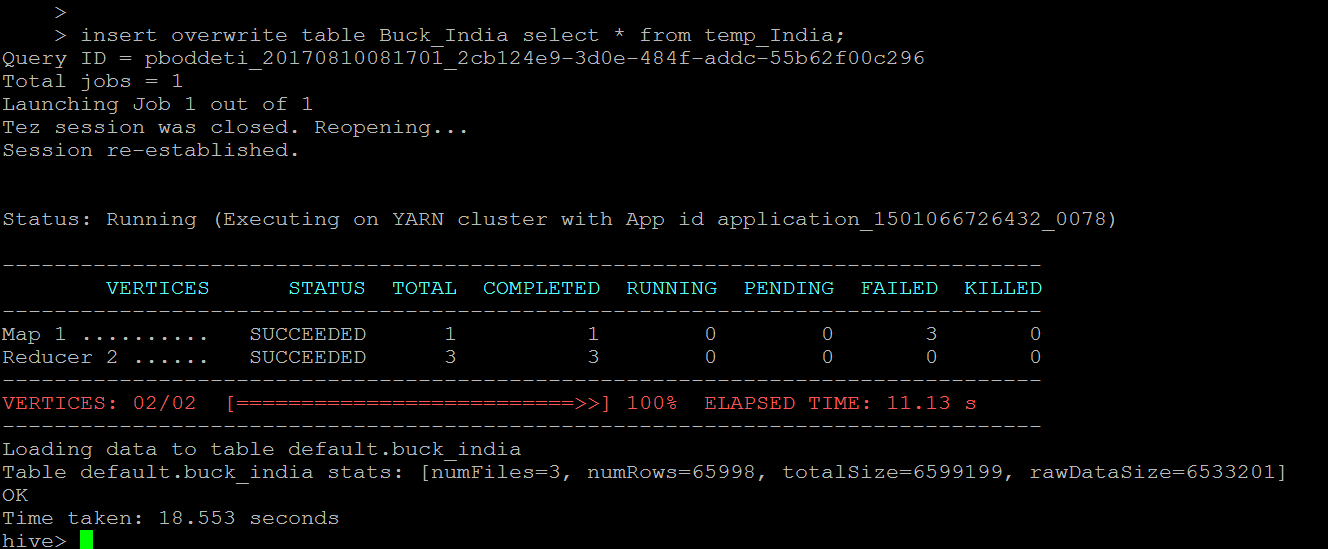
ROW FORMAT DELIMITED

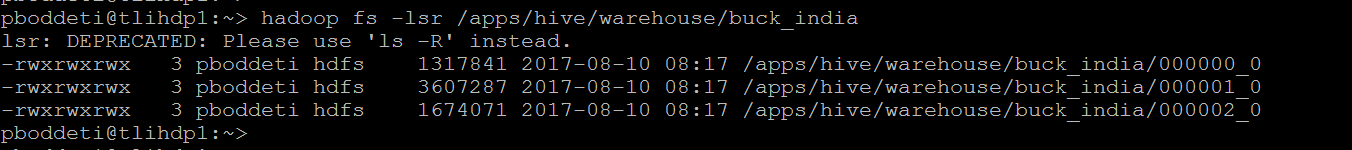
FIELDS TERMINATED BY ','

STORED AS TEXTFILE;



**insert overwrite table Buck\_India select \* from temp\_India;**

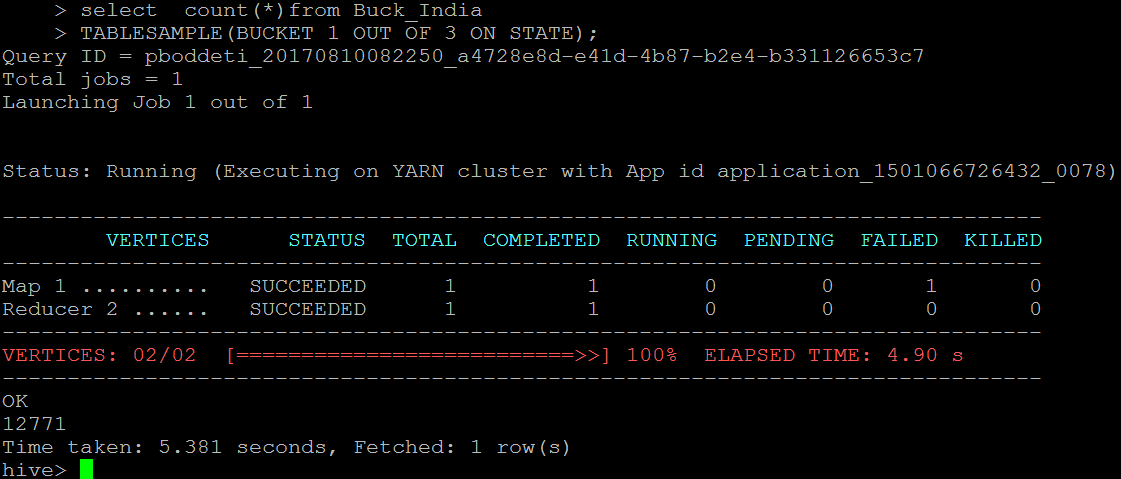




Let’s count how many records we have in our different buckets. We will use hive sampling query for this.

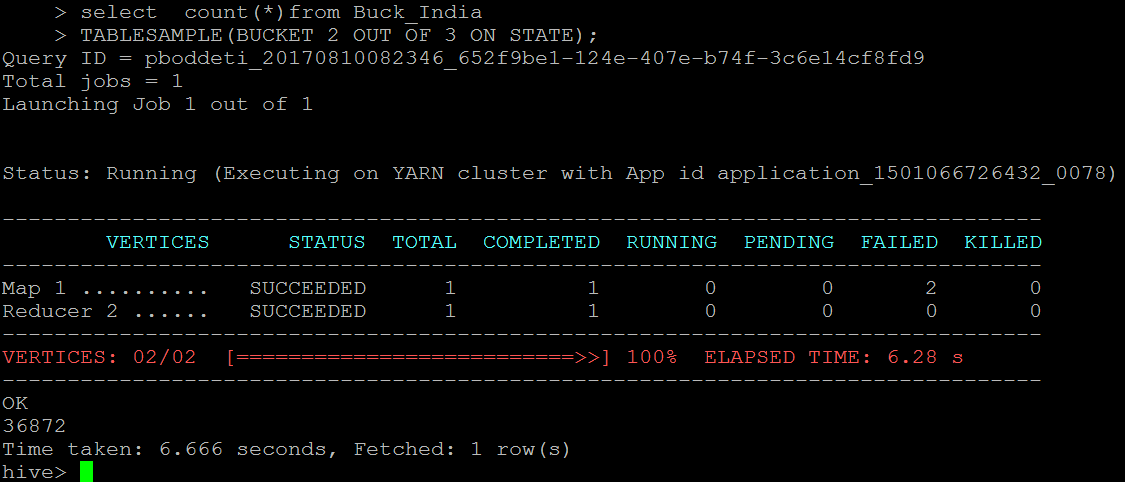
**select count(\*)from Buck\_India**

**TABLESAMPLE(BUCKET 1 OUT OF 3 ON STATE);**



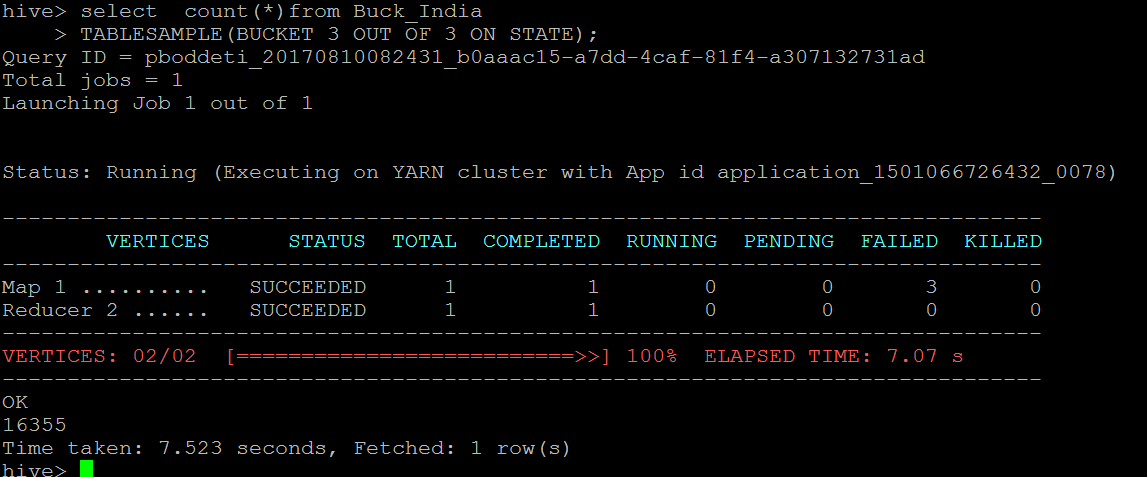
**select count(\*)from Buck\_India**

**TABLESAMPLE(BUCKET 2 OUT OF 3 ON STATE);**



**select count(\*)from Buck\_India**

**TABLESAMPLE(BUCKET 3 OUT OF 3 ON STATE);**



hadoop fs -cat /apps/hive/warehouse/buck\_india/000000\_0 |grep ',ANDHRA PRADESH,'

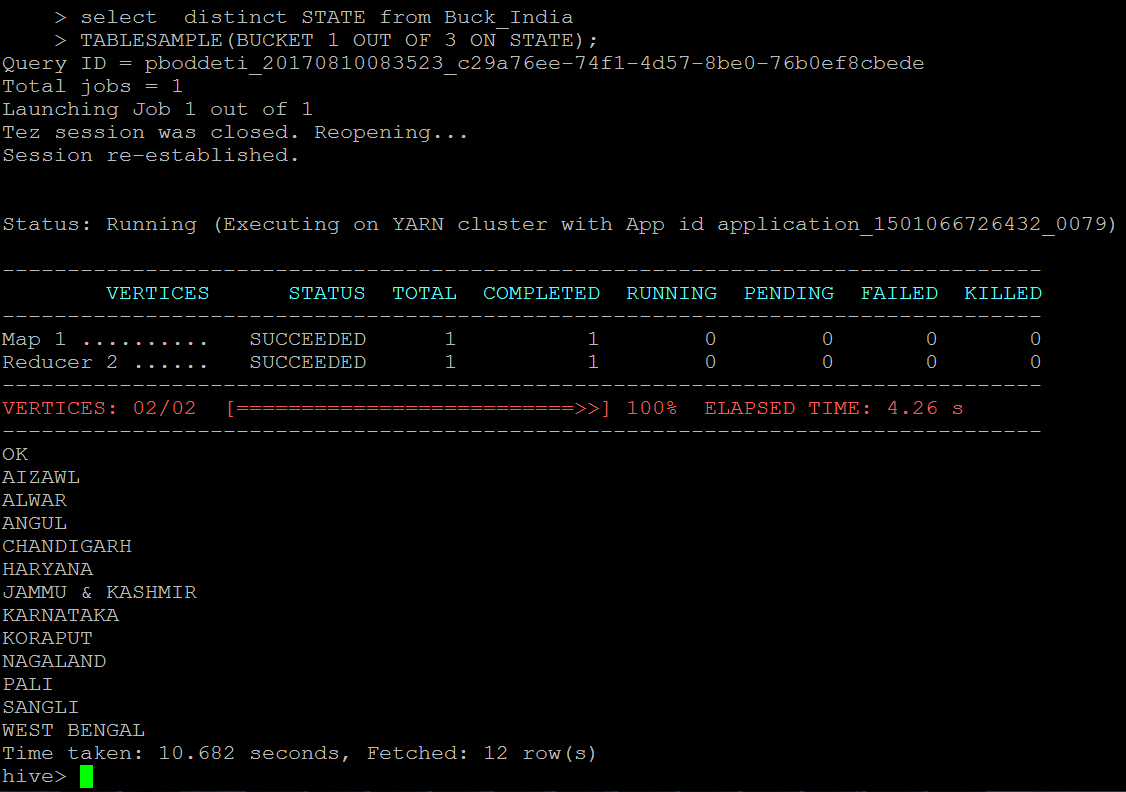
hadoop fs -cat /apps/hive/warehouse/buck\_india/000001\_0 |grep ',ANDHRA PRADESH,'



What does this mean, a specific state does not go into two buckets, that’s what we expect. Let’s check which states bucket 1 and 2 contains?

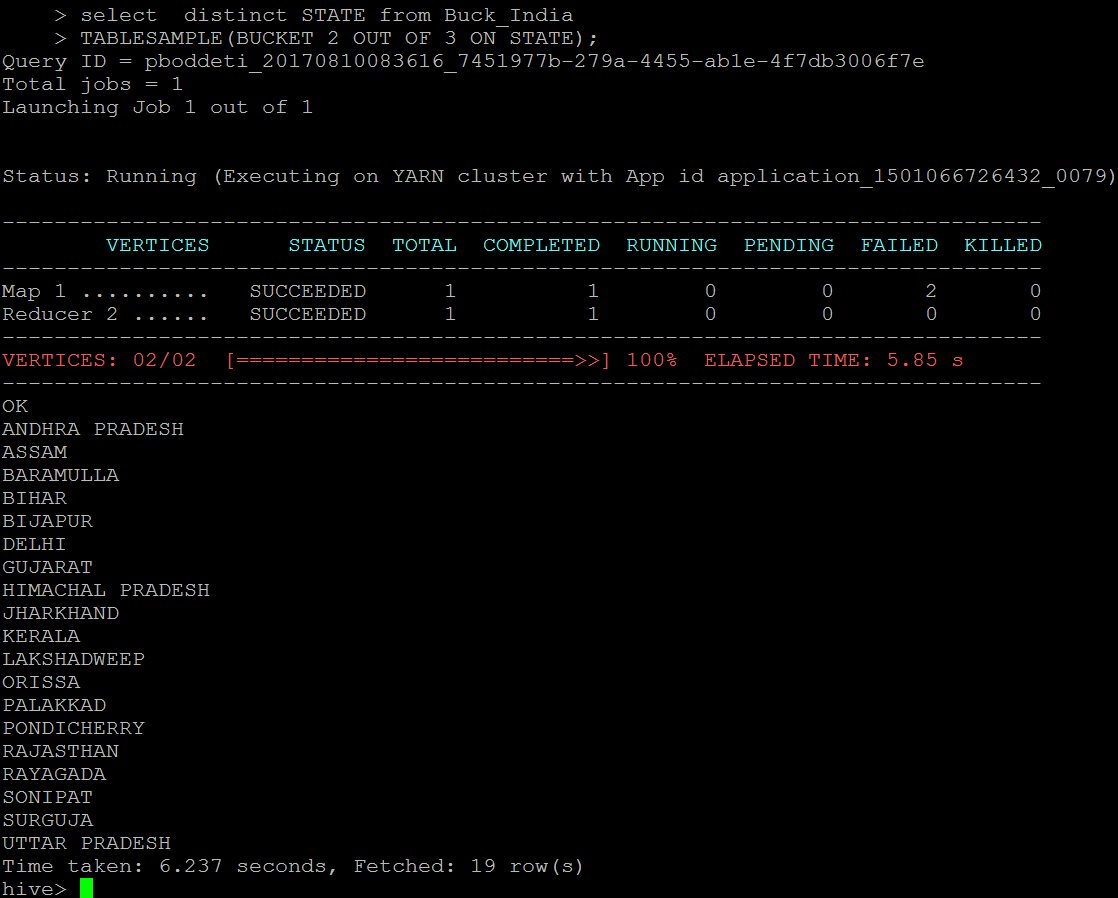
**select distinct STATE from Buck\_India**

**TABLESAMPLE(BUCKET 1 OUT OF 3 ON STATE);**



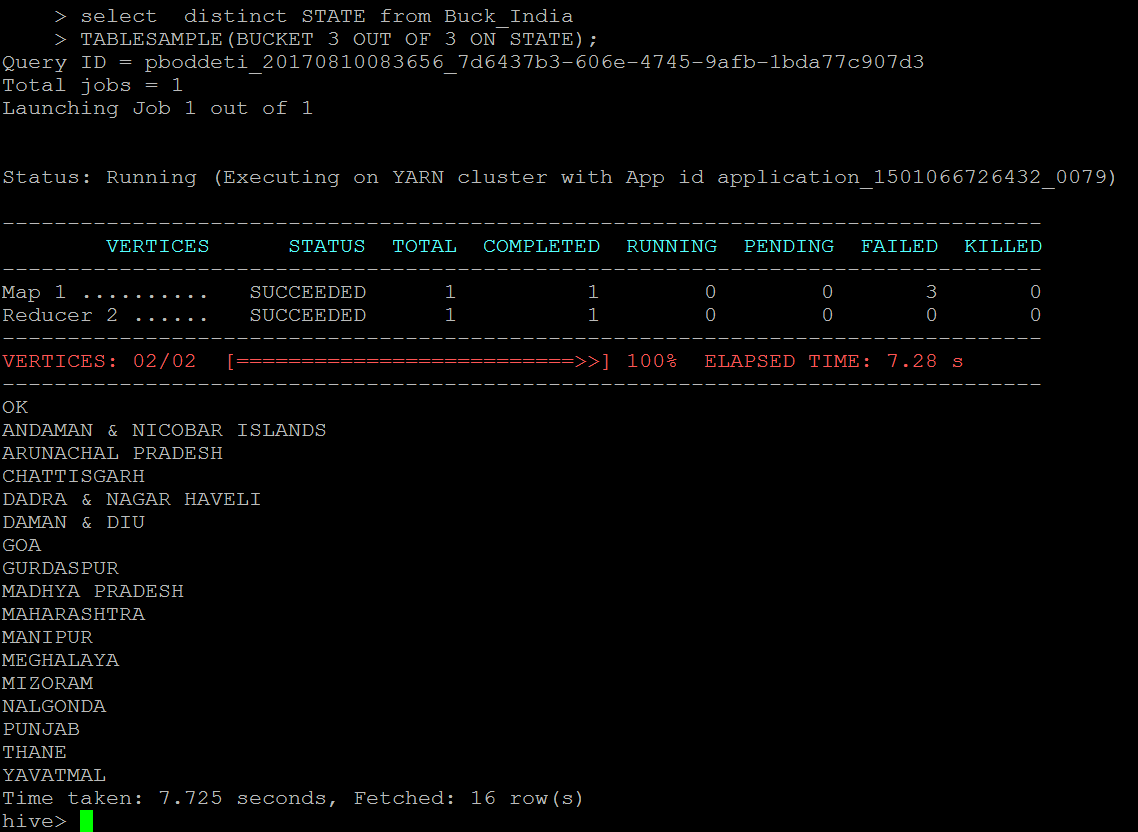
**select distinct STATE from Buck\_India**

**TABLESAMPLE(BUCKET 2 OUT OF 3 ON STATE);**



**select distinct STATE from Buck\_India**

**TABLESAMPLE(BUCKET 3 OUT OF 3 ON STATE);**



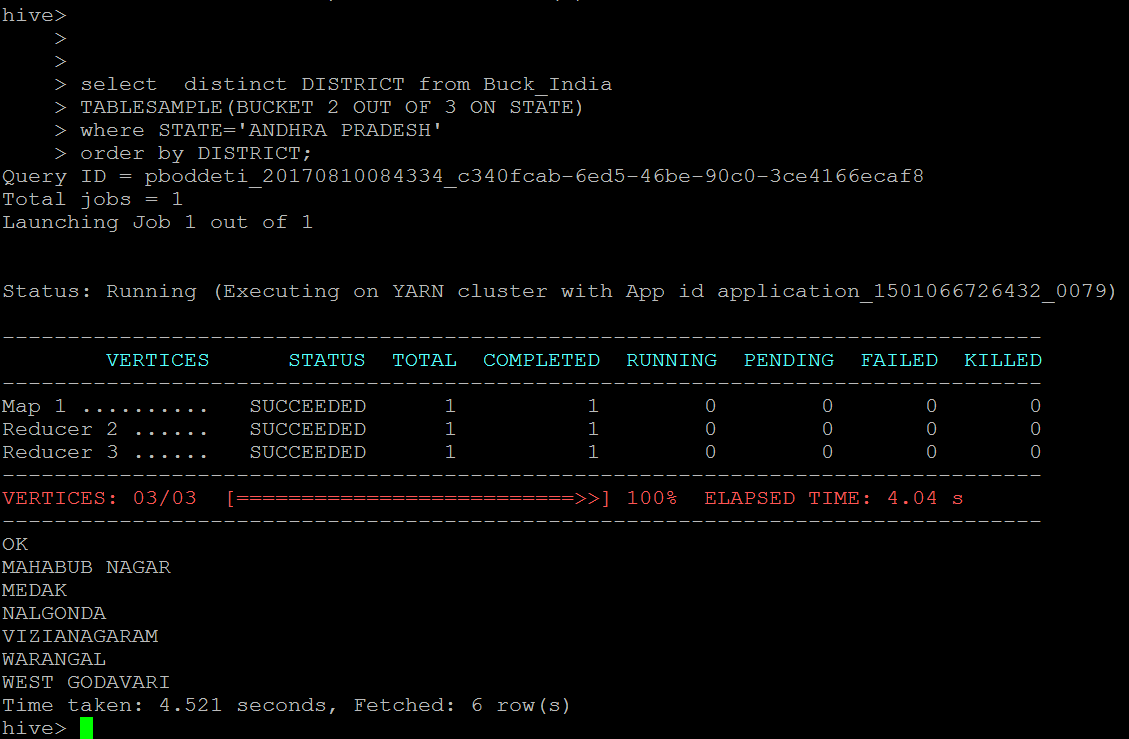
So, bucket 3 contains data for GOA and PUNJAB whereas bucket 2 contains data for ANDHRA PRADESH. I hope you understood what hive has done when we inserted data into this bucketed table. Let me summarize. Hive created three buckets as I instructed it to do so in create table statement. When I loaded data into this table, hive has used some hashing technique for each country to generate a number in range of 1 to 3. Based on the outcome of hashing, hive has placed data row into appropriate bucked. This hashing technique is deterministic so that when hive has to read it back or write more data, it goes in correct bucket.  
One thing about tablesample clause is worth mentioning, it’s part of from clause so you can have where clause or other clauses like order by etc as usual. Check below example if you have any doubt.

**select distinct DISTRICT from Buck\_India**

**TABLESAMPLE(BUCKET 2 OUT OF 3 ON STATE)**

**Where STATE=’ANDHRA PRADESH’**

**Order by DISTRICT;**

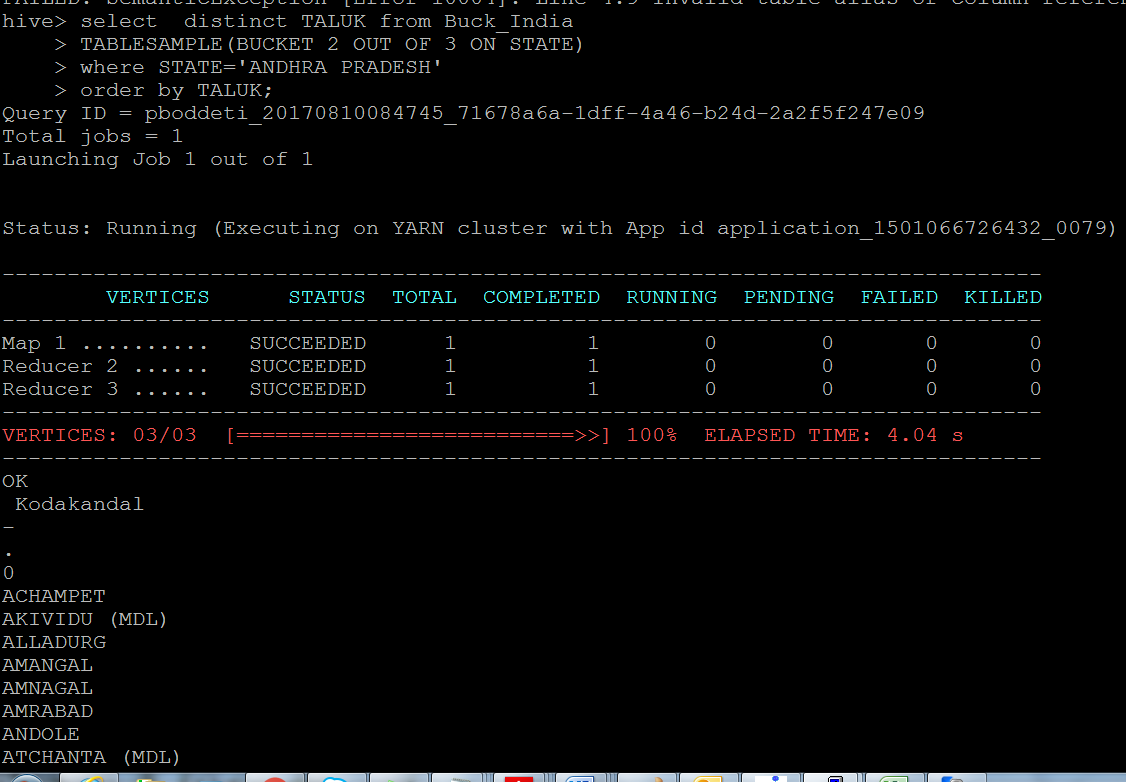


**select distinct TALUK from Buck\_India**

**TABLESAMPLE(BUCKET 2 OUT OF 3 ON STATE)**

**Where STATE=’ANDHRA PRADESH’**

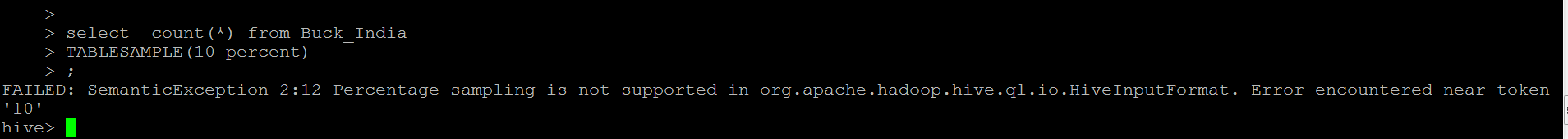
**Order by TALUK;**



Hive supports another type of sampling. That’s called block sampling. In this sampling we specify percentage of data to be sampled. Let’s see an example.

**select count(\*) from Buck\_India**

**TABLESAMPLE(10 percent)**



Hive Imp Questions:

**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.

**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 nonstrict mode Here is hive dynamic partition properties you should allow

SET hive.exec.dynamic.partition = true;

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