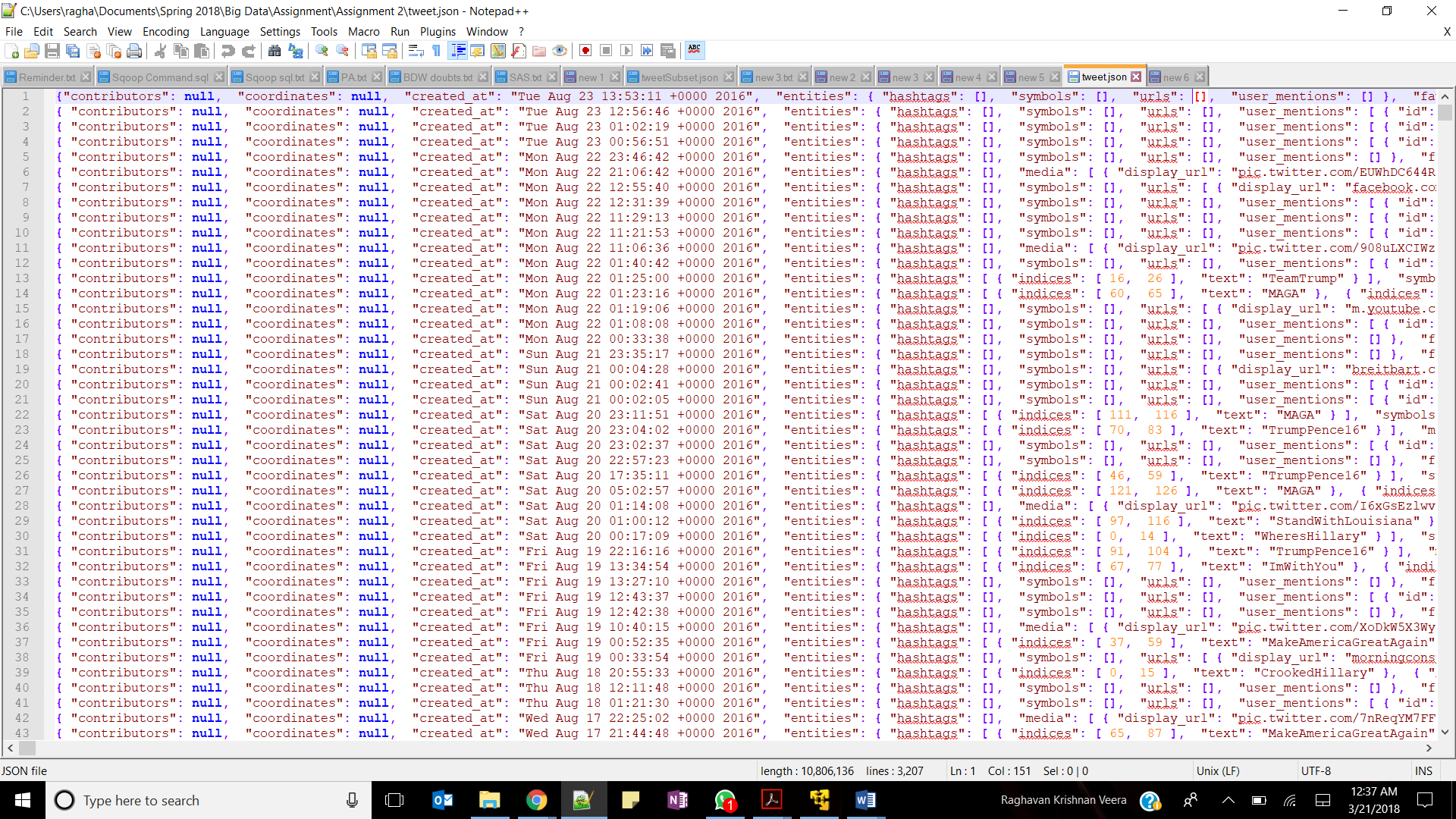
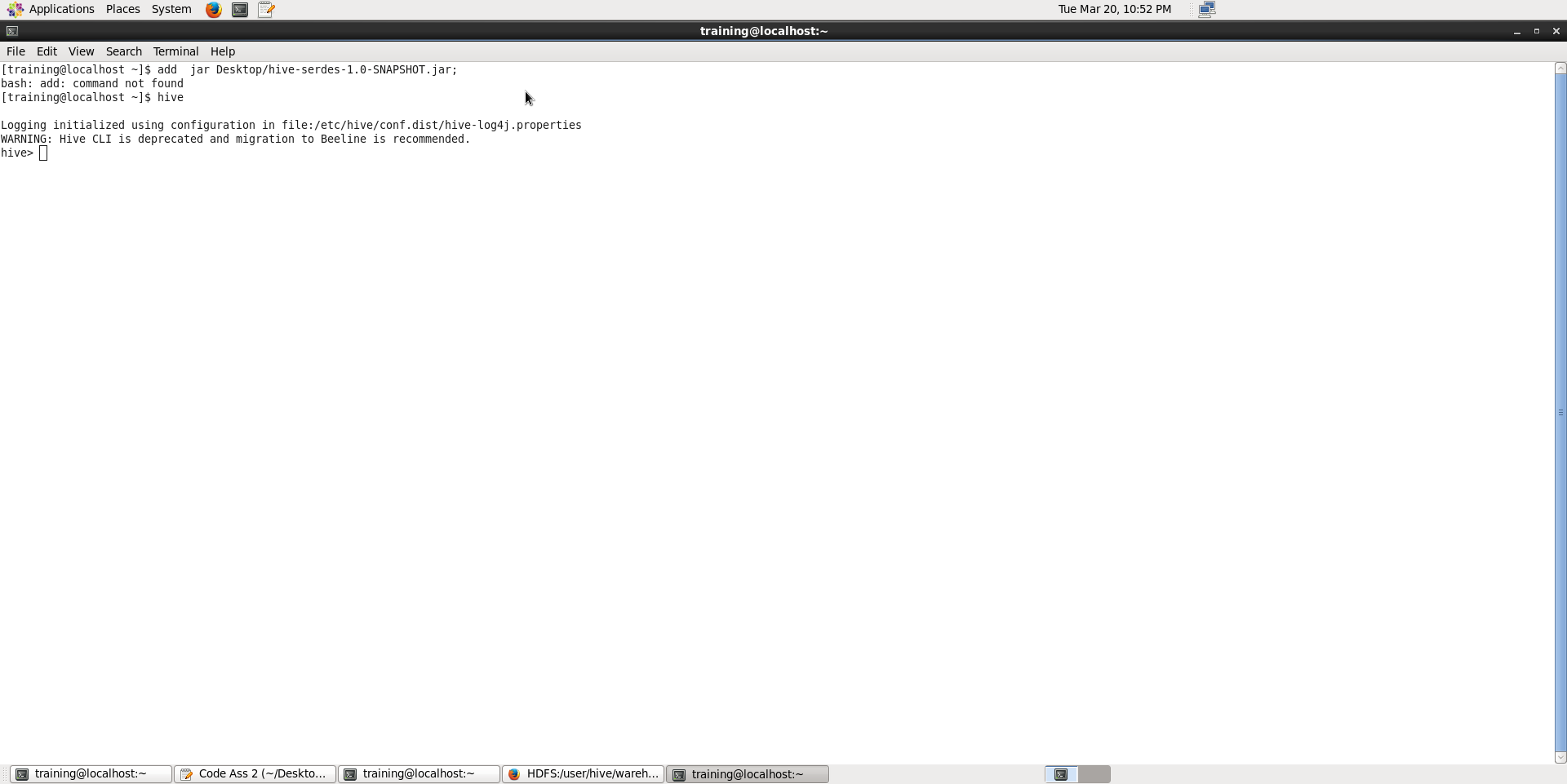
Cleaned the JSON file as rows separated by a new line character.



Added the SerDe into hive environment.



CREATE TABLE tweets\_json (

id BIGINT,

created\_at STRING,

source STRING,

favorited BOOLEAN,

retweeted\_status STRUCT<

text:STRING,

user:STRUCT<screen\_name:STRING,name:STRING>,

retweet\_count:INT>,

entities STRUCT<

urls:ARRAY<STRUCT<expanded\_url:STRING>>,

user\_mentions:ARRAY<STRUCT<screen\_name:STRING,name:STRING>>,

hashtags:ARRAY<STRUCT<text:STRING>>>,

place STRUCT<

country\_code:STRING,

full\_name:STRING>,

retweet\_count STRING,

text STRING,

user STRUCT<

screen\_name:STRING,

name:STRING,

id:int,

friends\_count:INT,

followers\_count:INT,

location:STRING,

statuses\_count:INT,

verified:BOOLEAN,

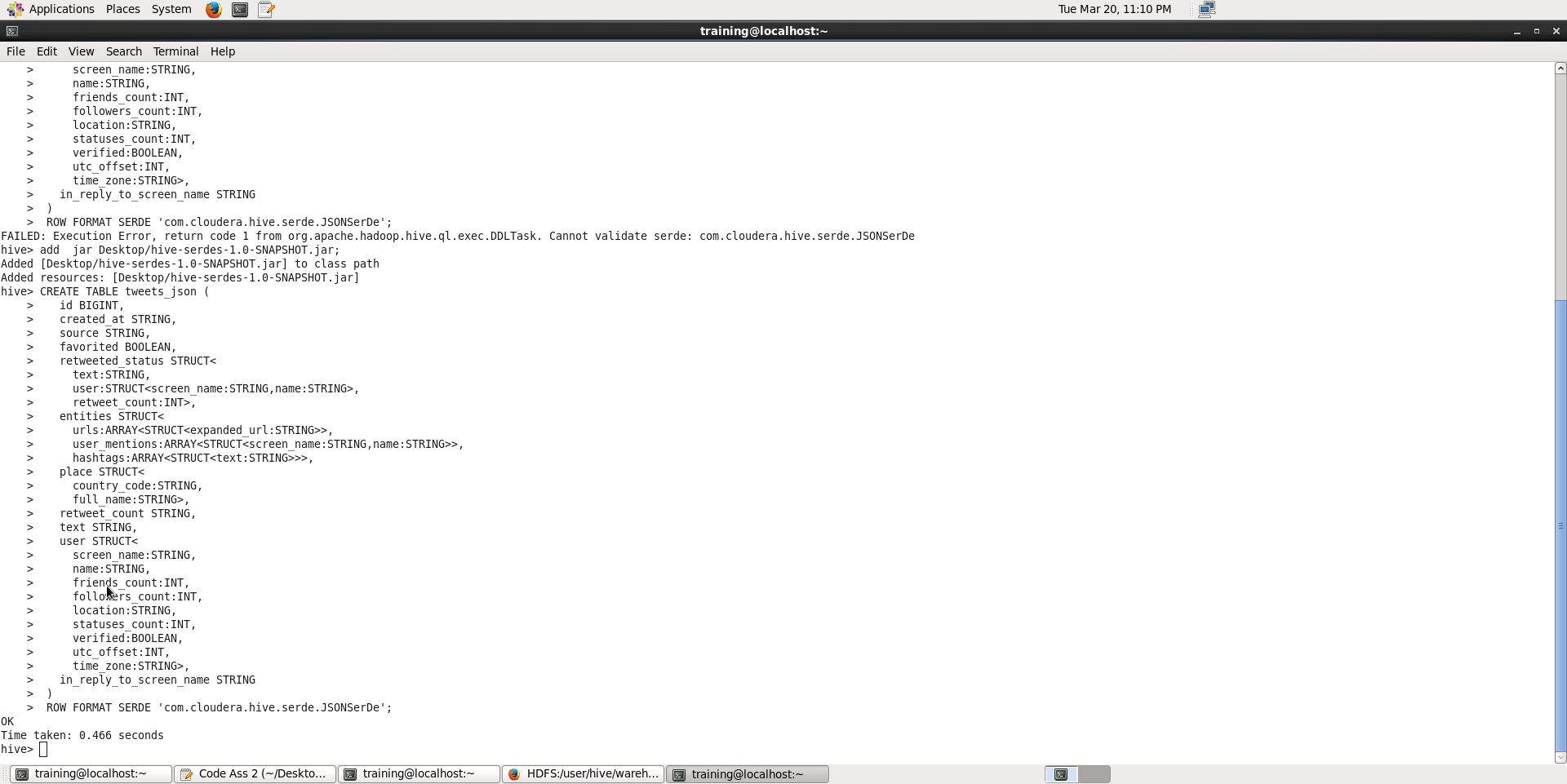
utc\_offset:INT,

time\_zone:STRING>,

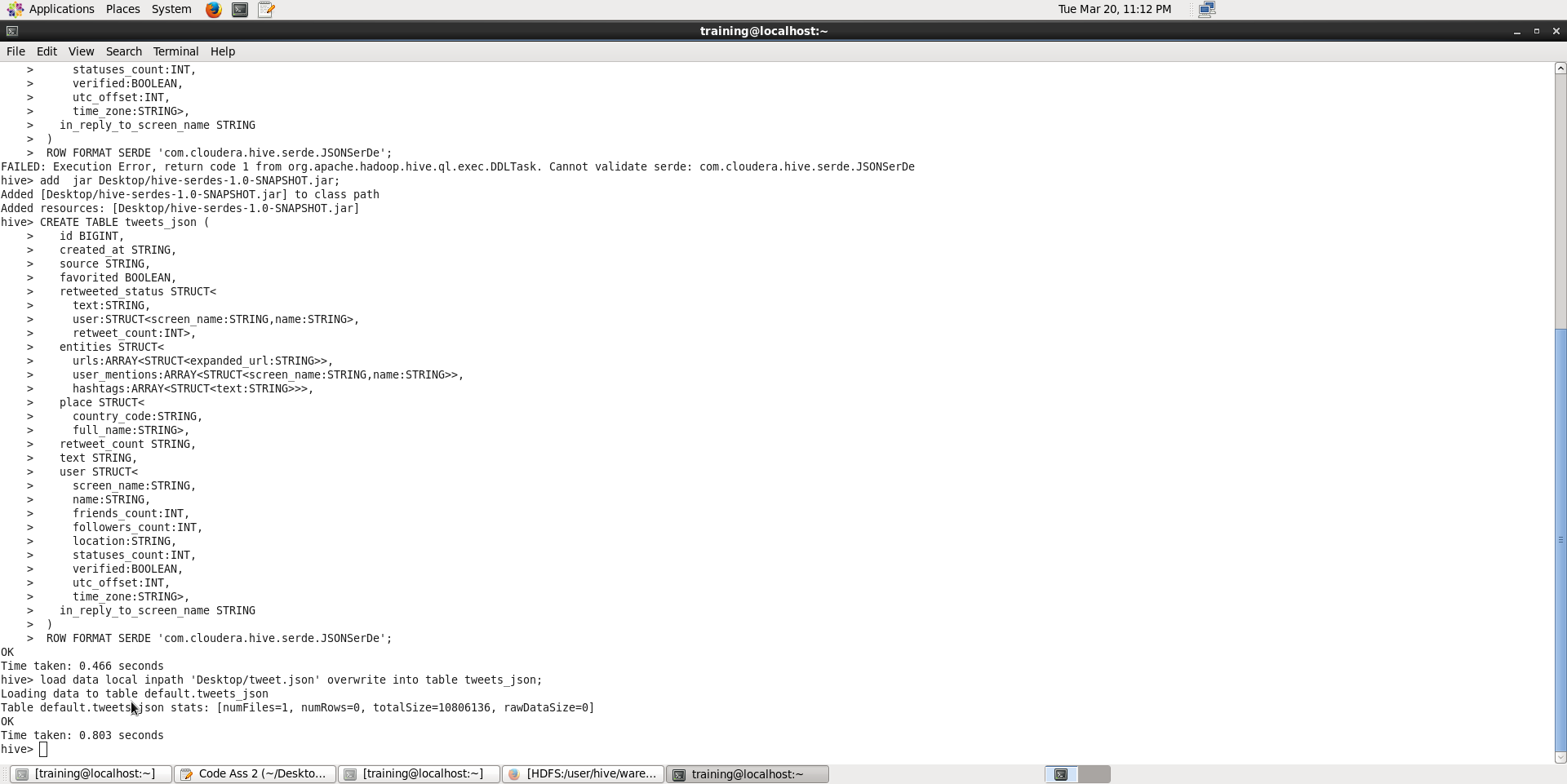
in\_reply\_to\_screen\_name STRING

)

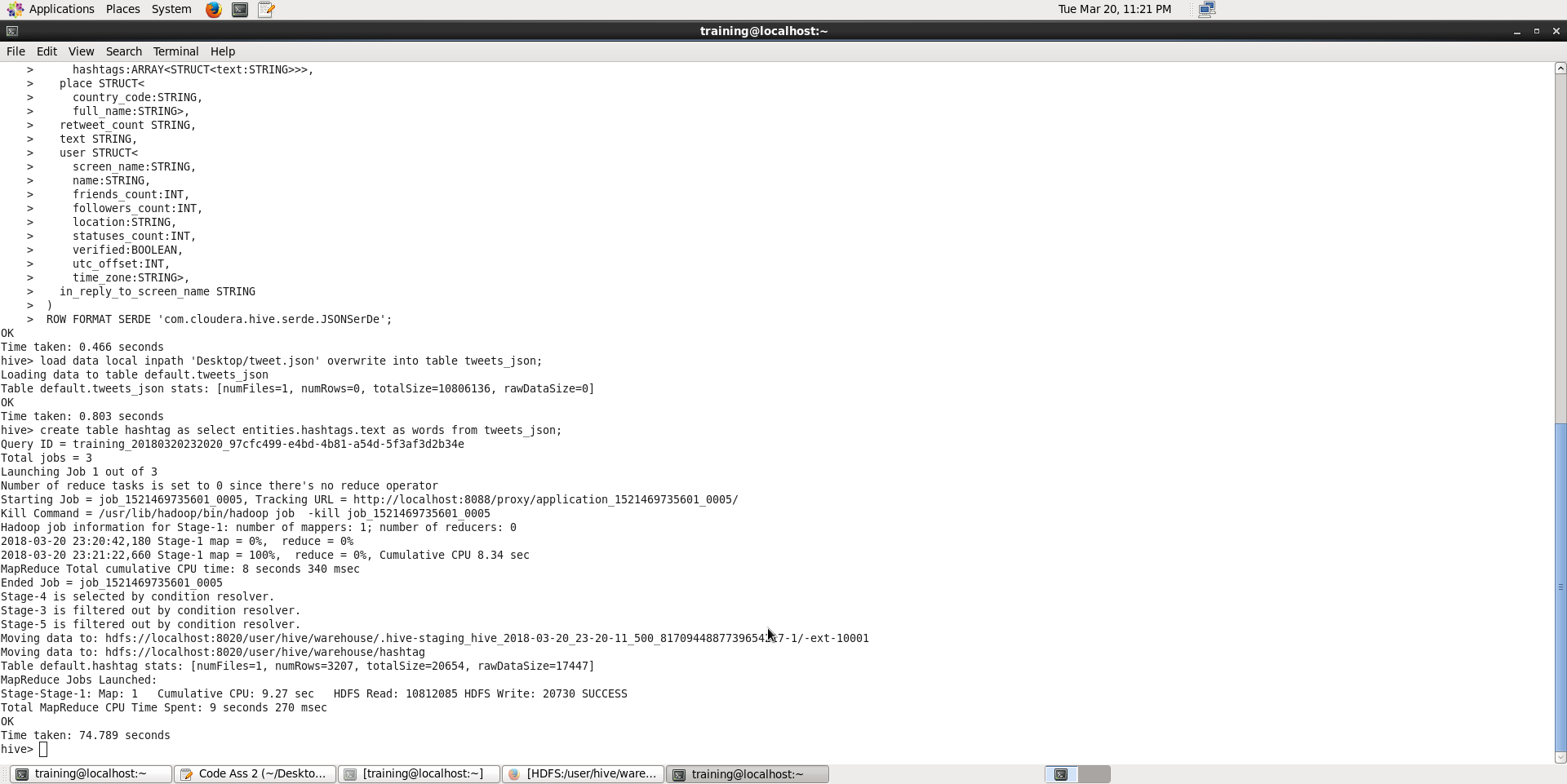
ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe';

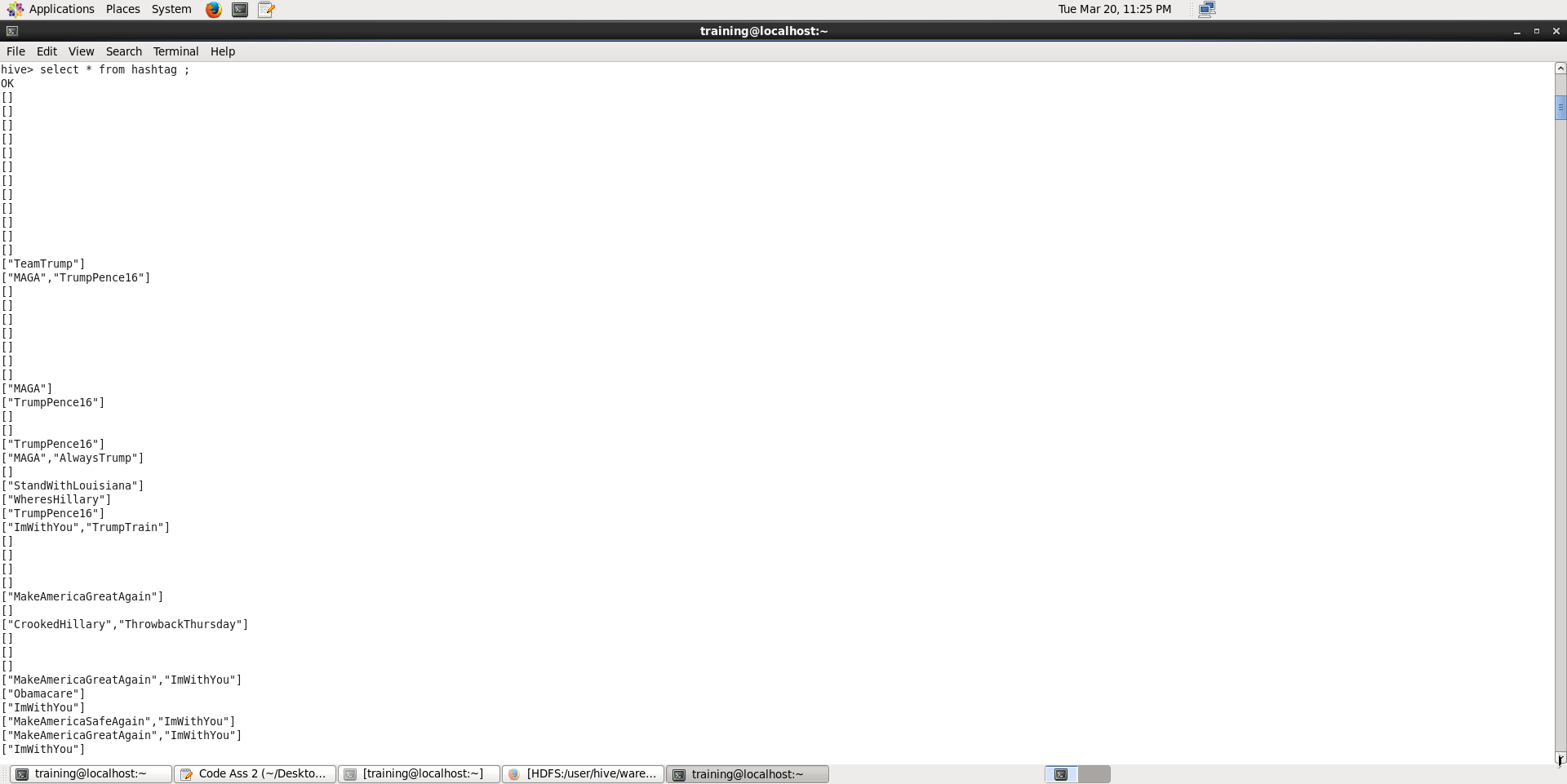


load data local inpath 'Desktop/tweet.json' overwrite into table tweets\_json;

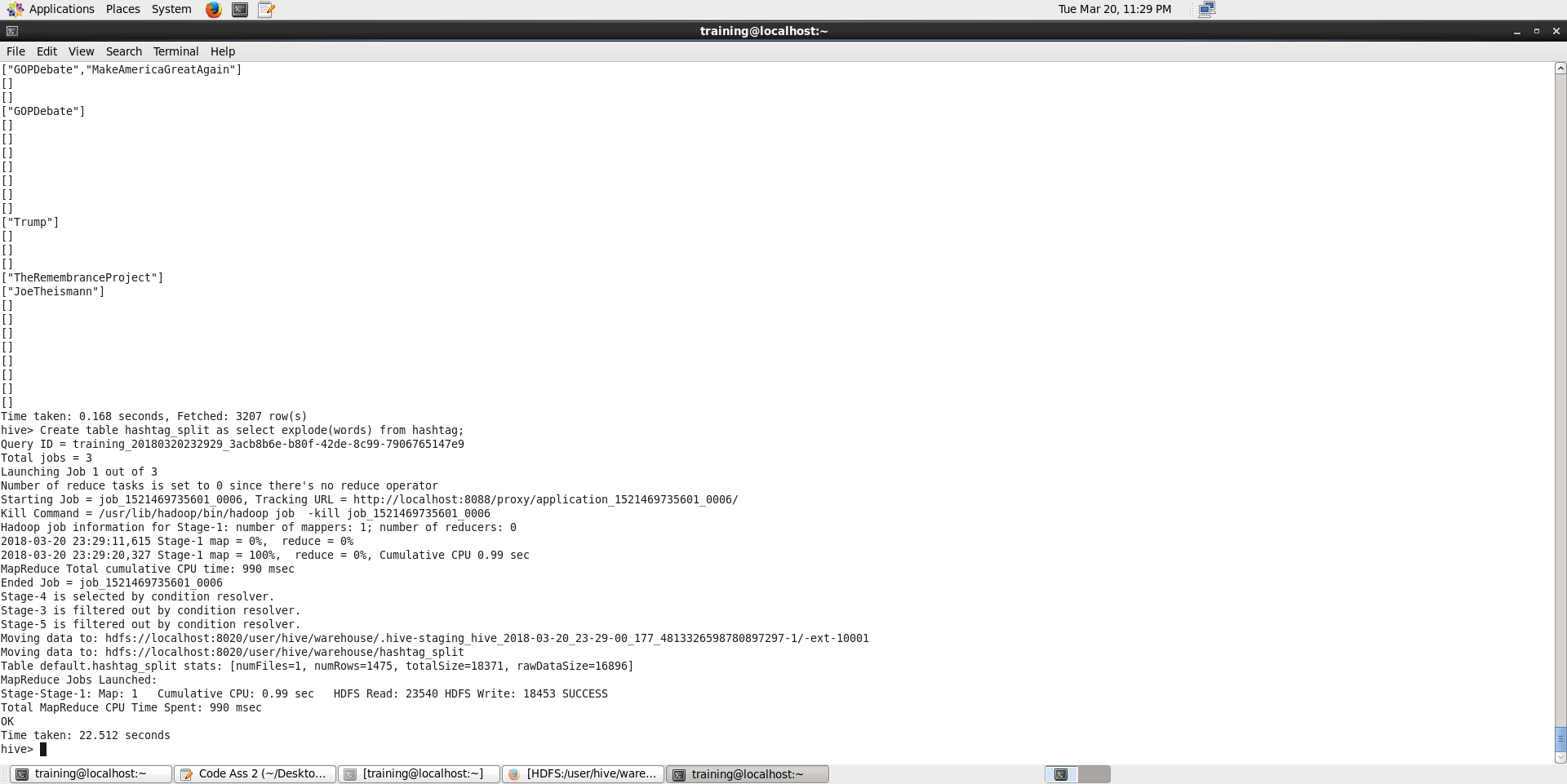


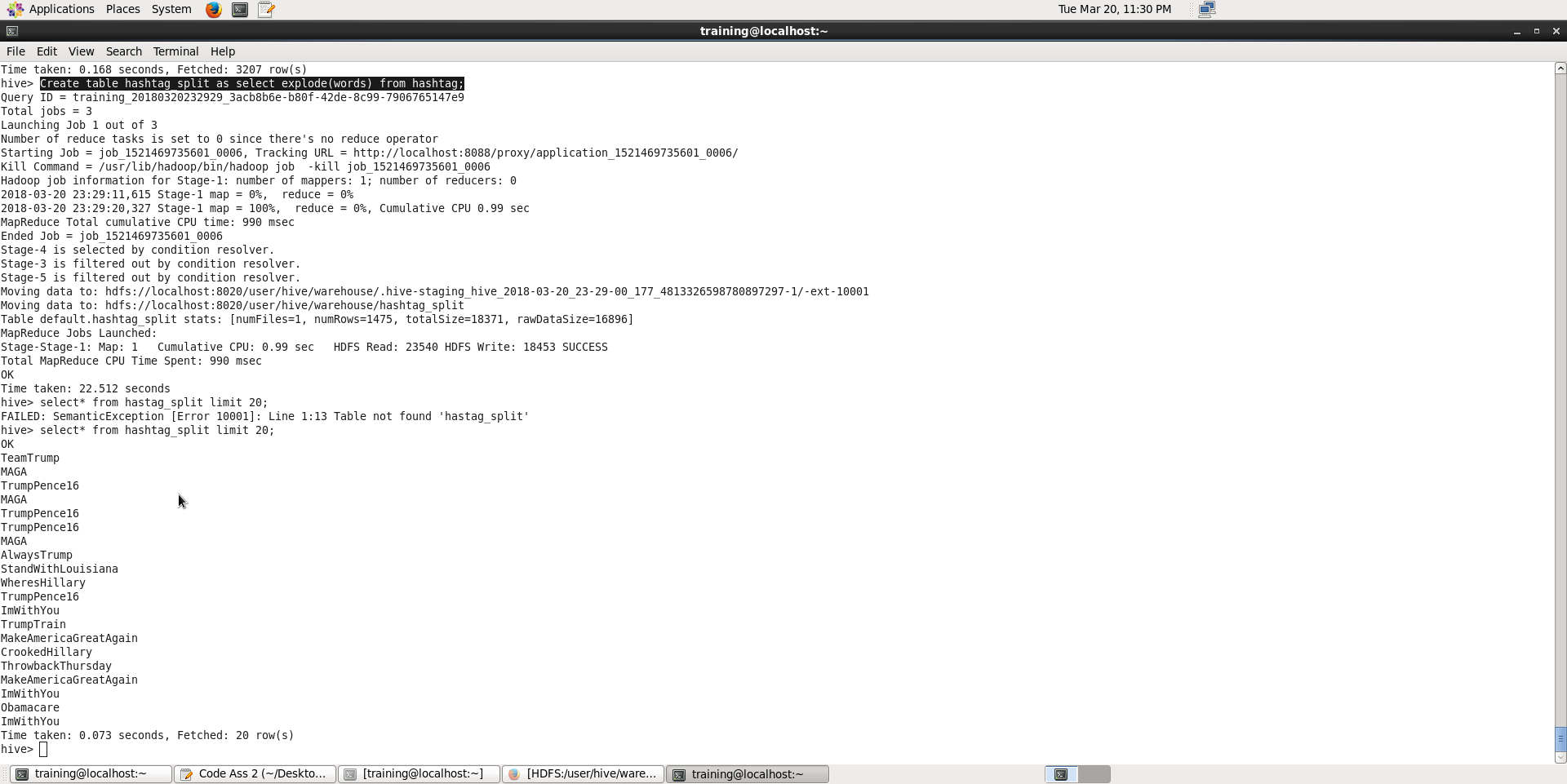
1. A. create table hashtag as select entities.hashtags.text as words from tweets\_json;



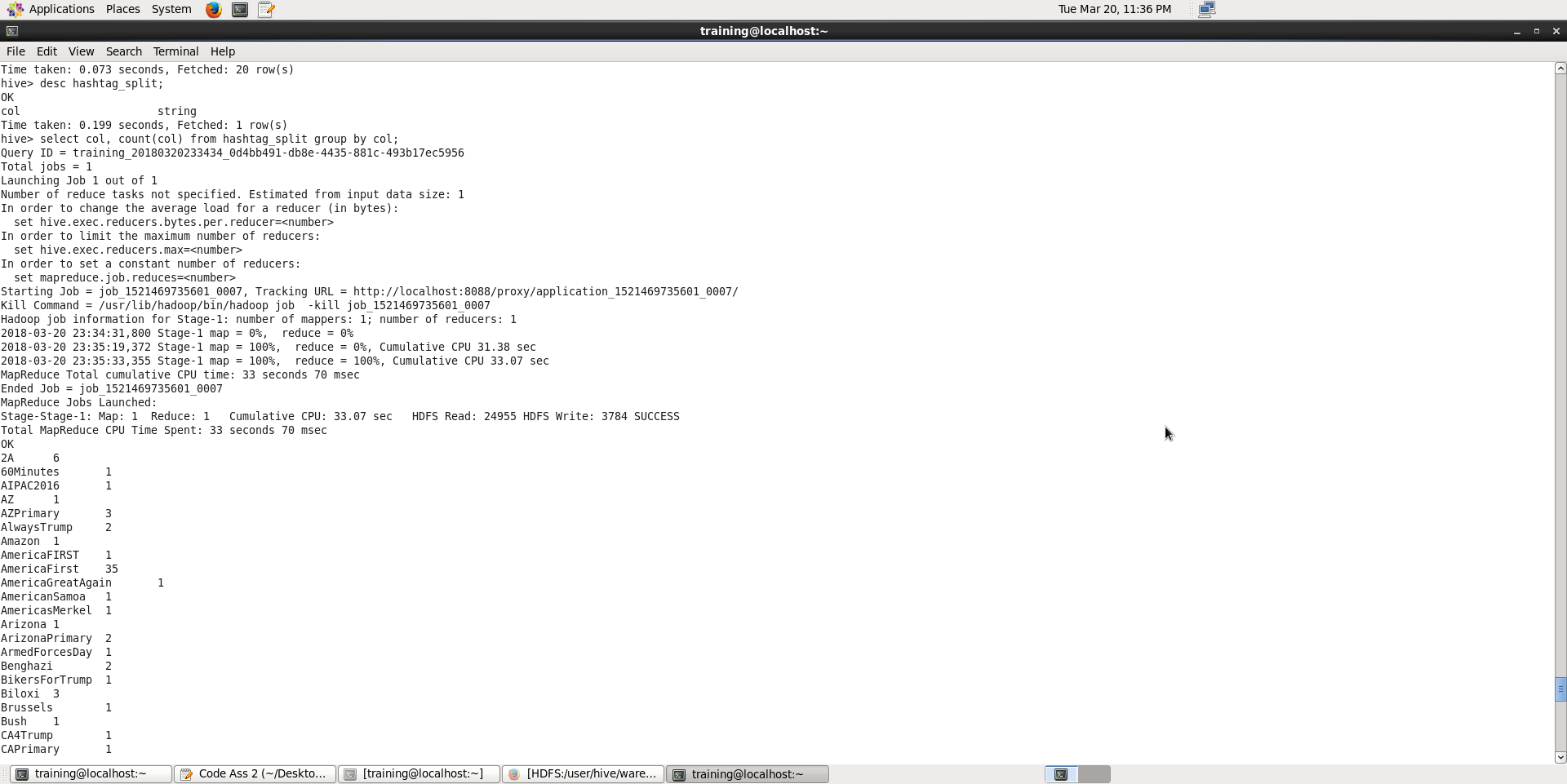


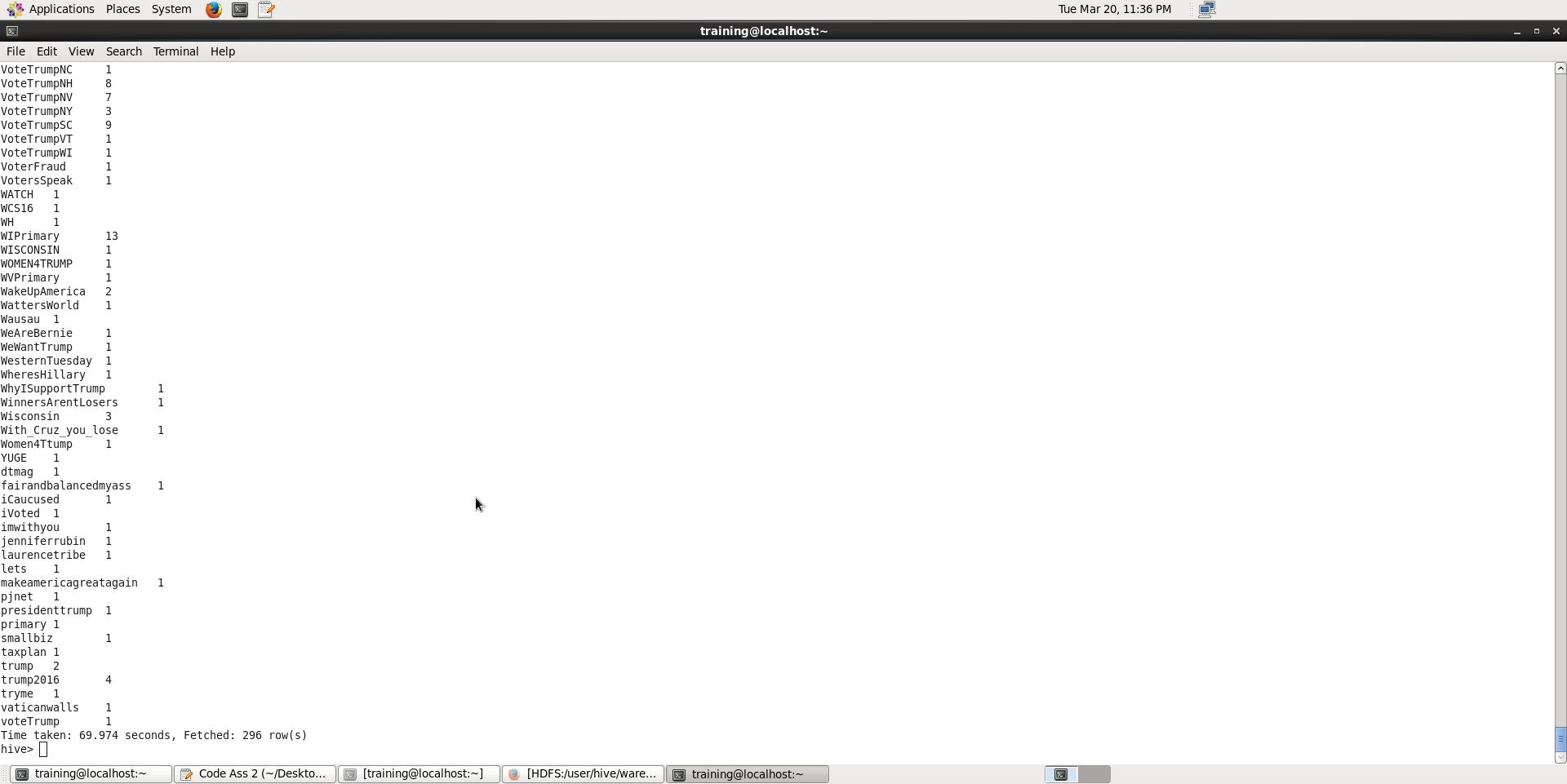
Create table hashtag\_split as select explode(words) from hashtag;





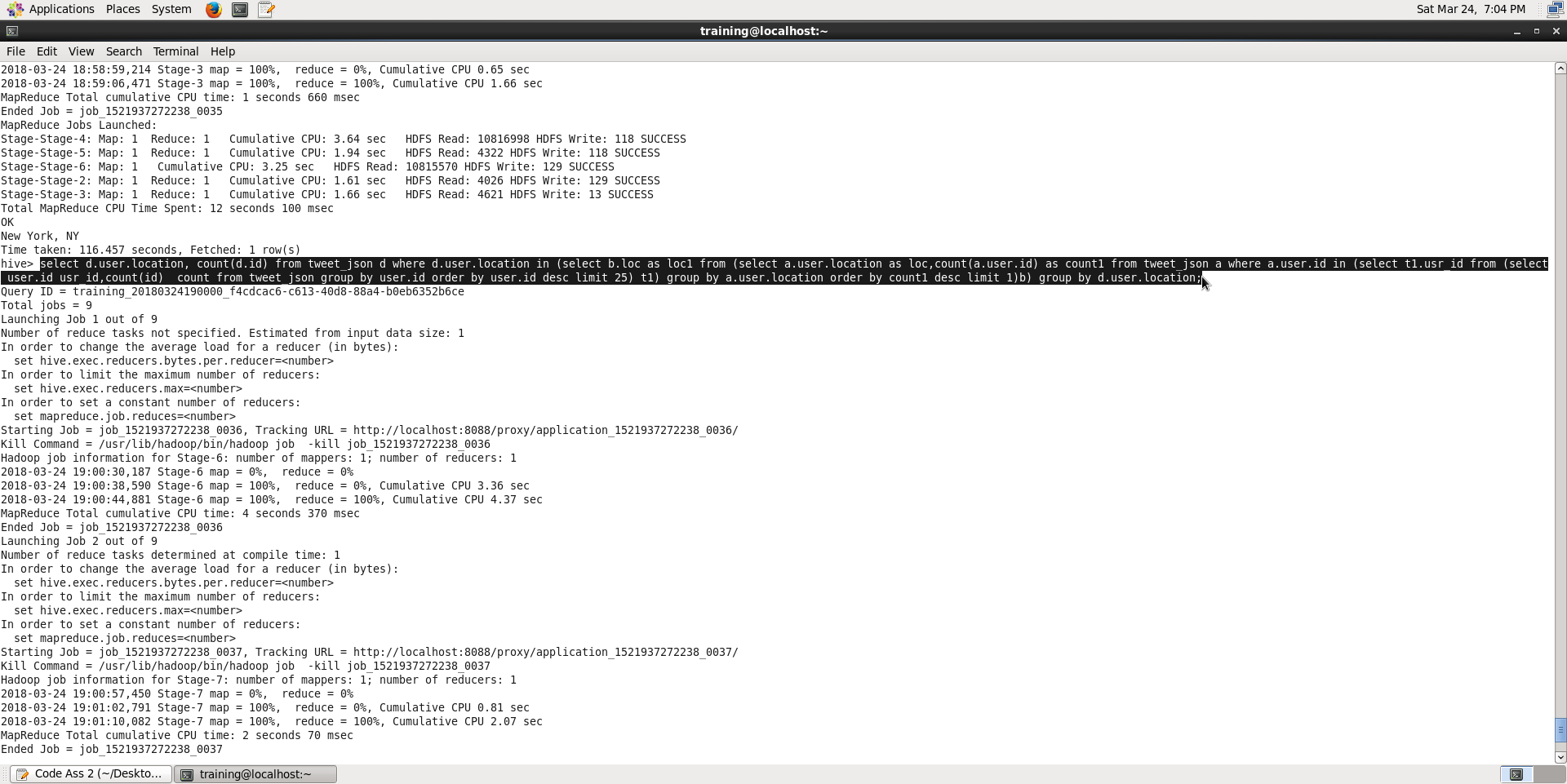
select col, count(col) from hashtag\_split group by col;

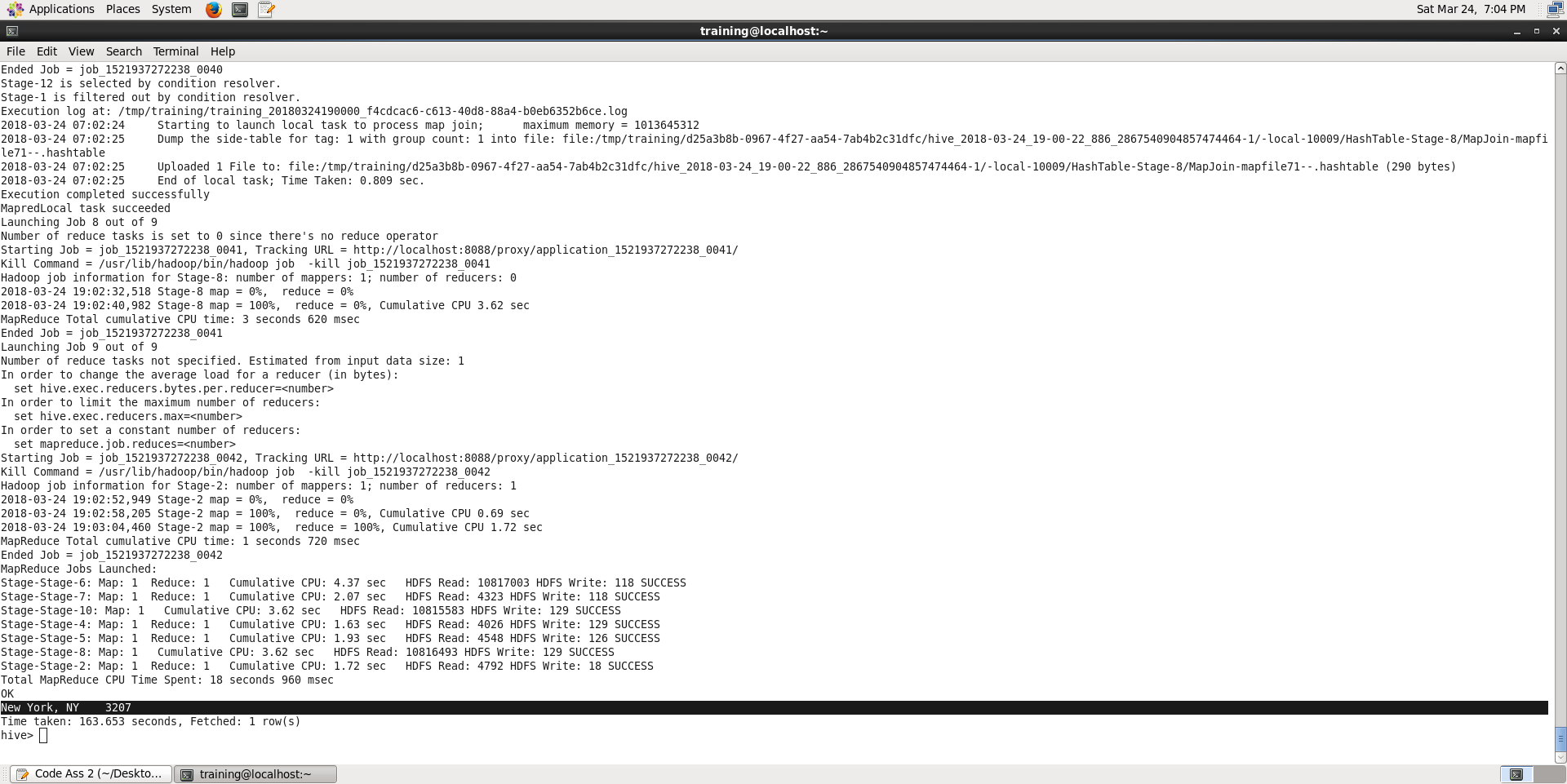




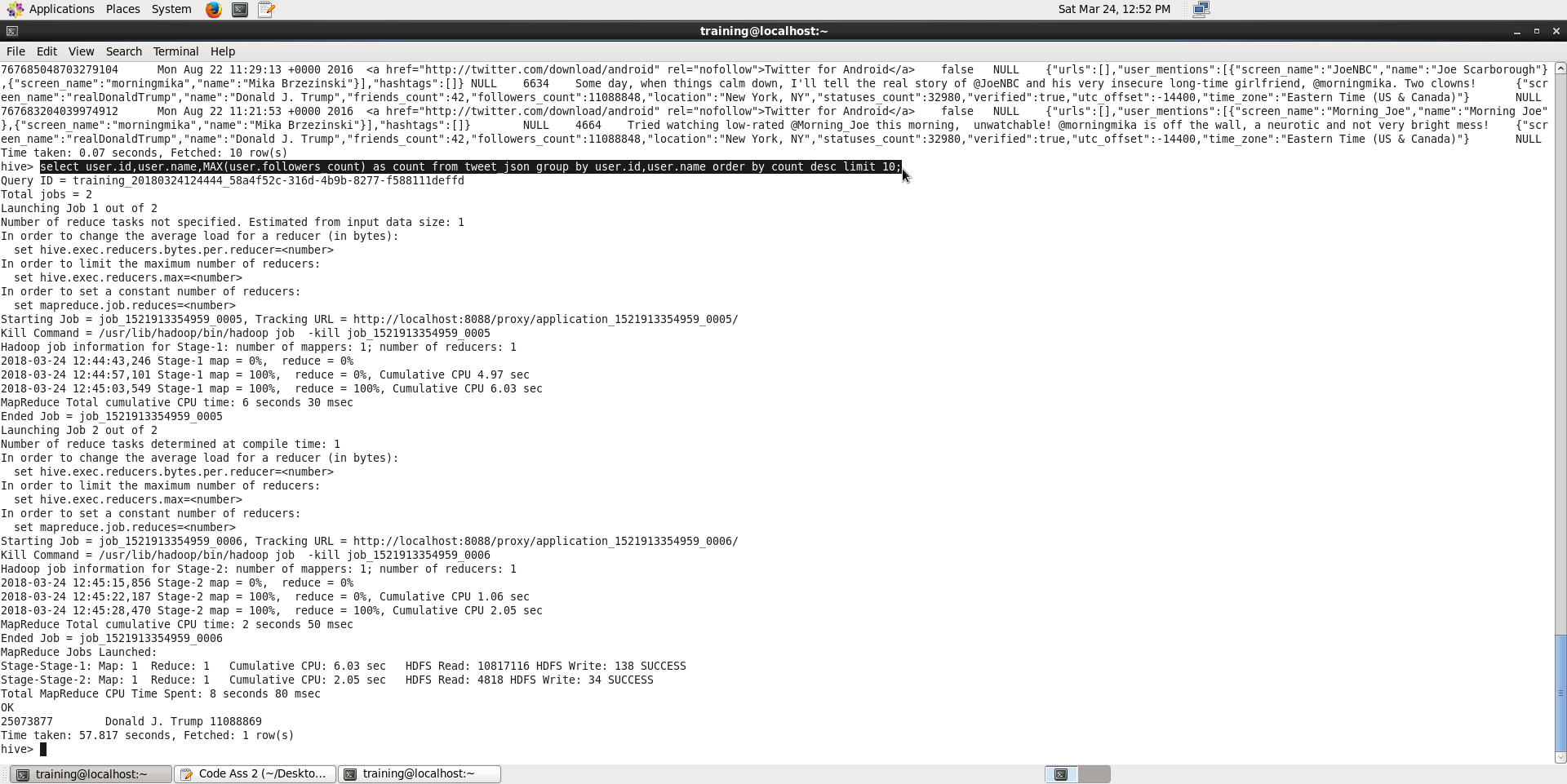
Ans: There are 296 unique hastags

B. select d.user.location, count(d.id) from tweet\_json d where d.user.location in (select b.loc as loc1 from (select a.user.location as loc,count(a.user.id) as count1 from tweet\_json a where a.user.id in (select t1.usr\_id from (select user.id usr\_id,count(id) count from tweet\_json group by user.id order by user.id desc limit 25) t1) group by a.user.location order by count1 desc limit 1)b) group by d.user.location;

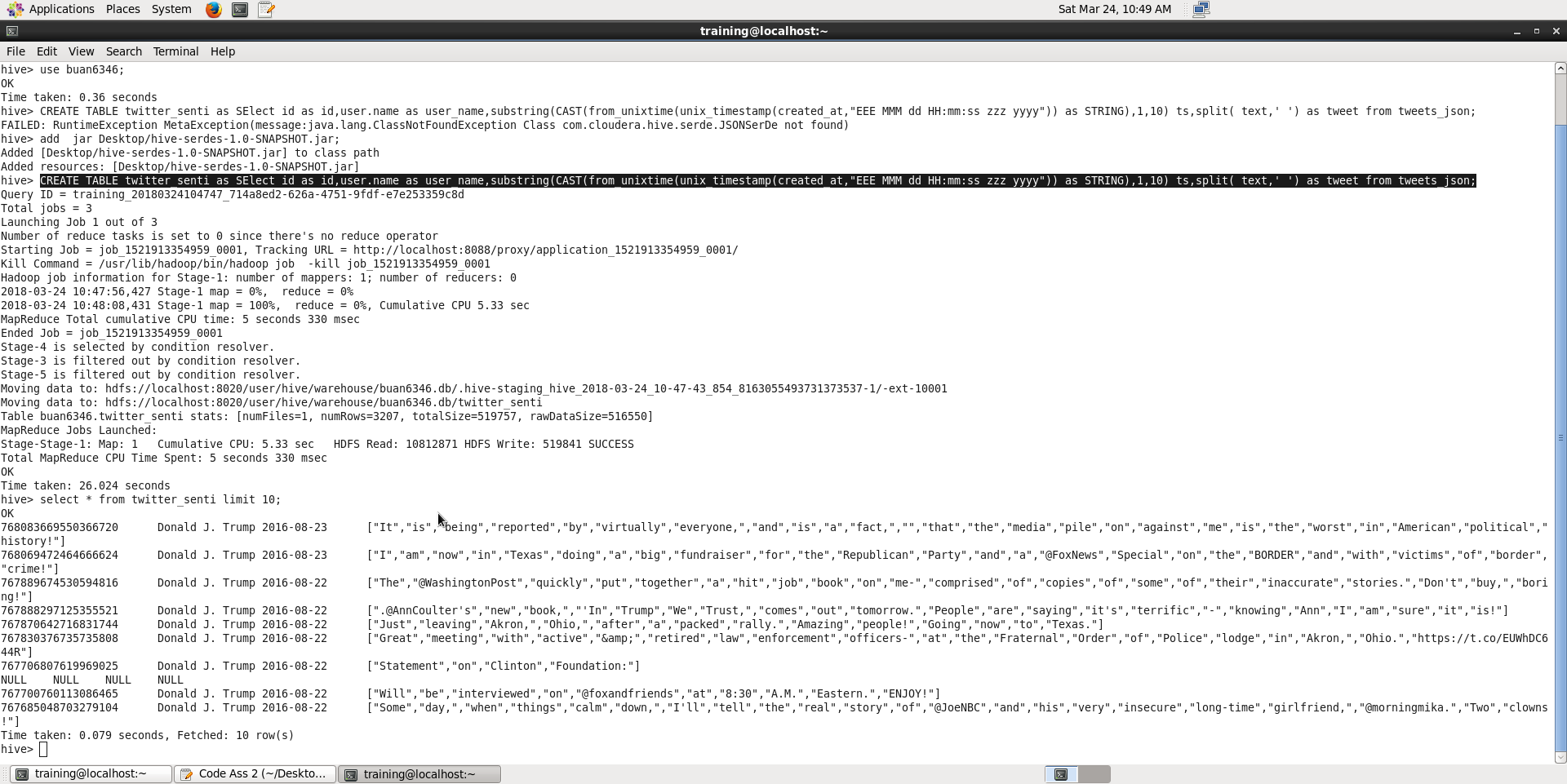




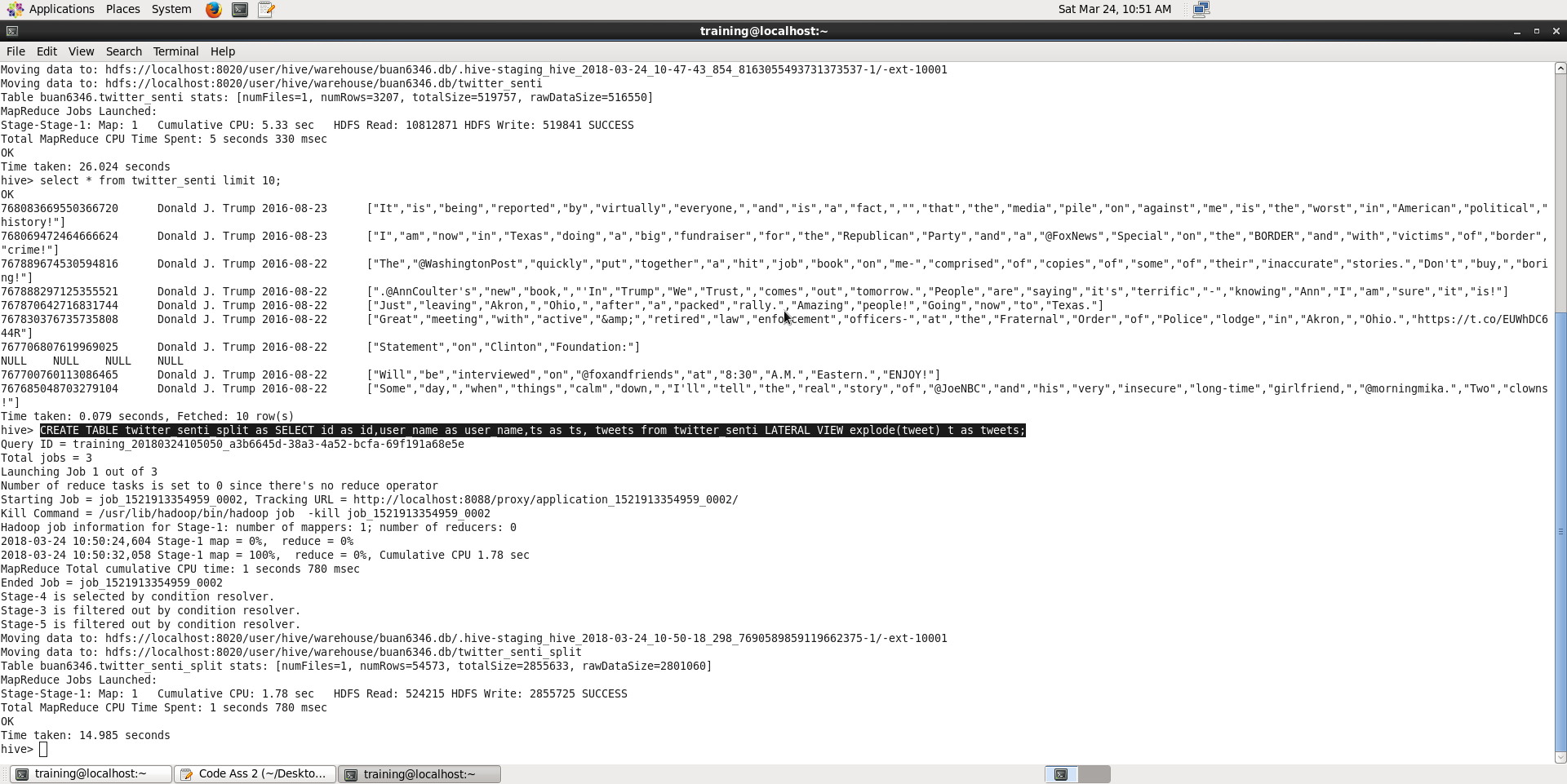
C. select user.id,user.name,MAX(user.followers\_count) as count from tweet\_json group by user.id,user.name order by count desc limit 10;

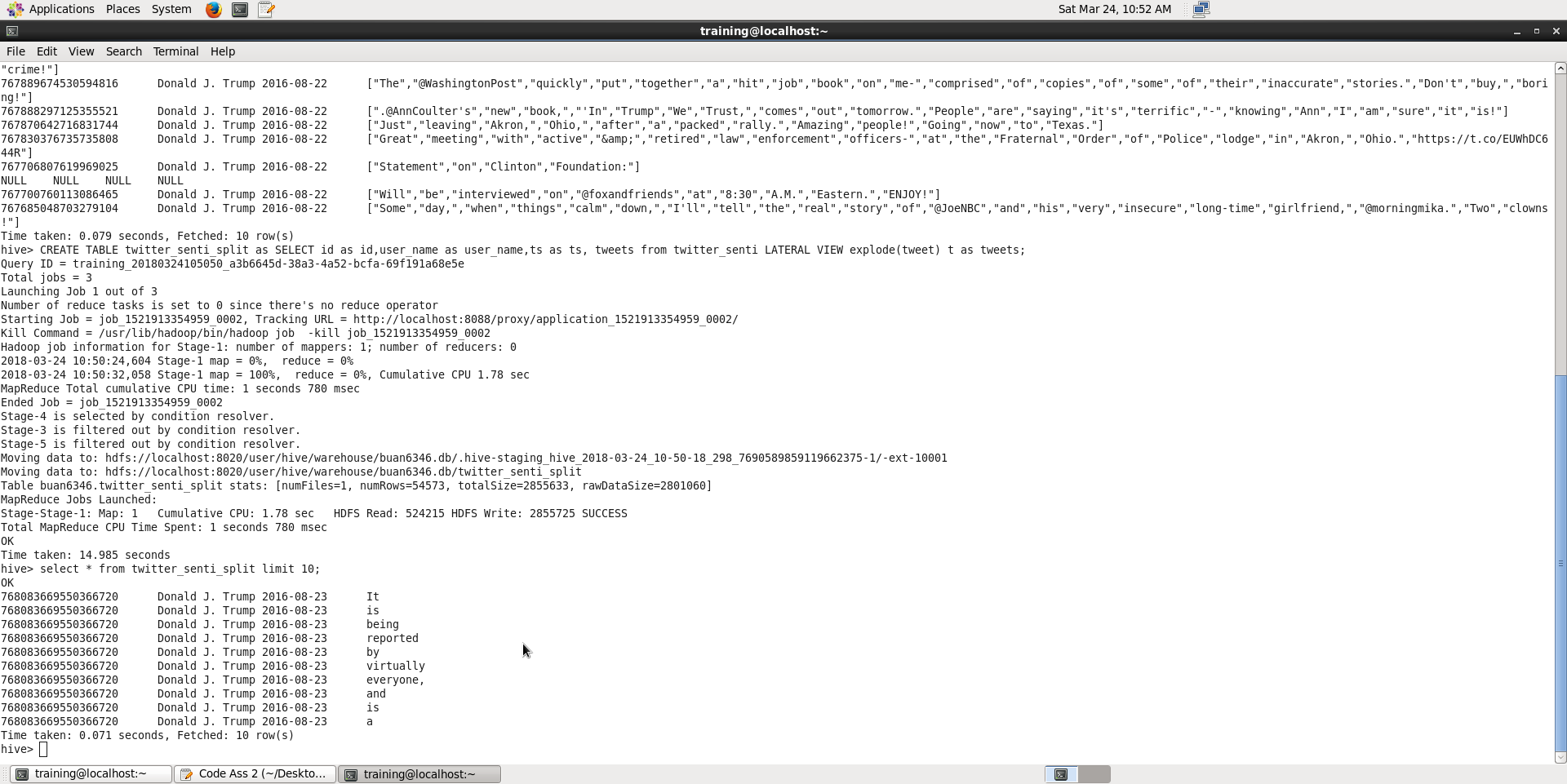


D. CREATE TABLE twitter\_senti as SElect id as id,user.name as user\_name,substring(CAST(from\_unixtime(unix\_timestamp(created\_at,"EEE MMM dd HH:mm:ss zzz yyyy")) as STRING),1,10) ts,split( text,' ') as tweet from tweets\_json;

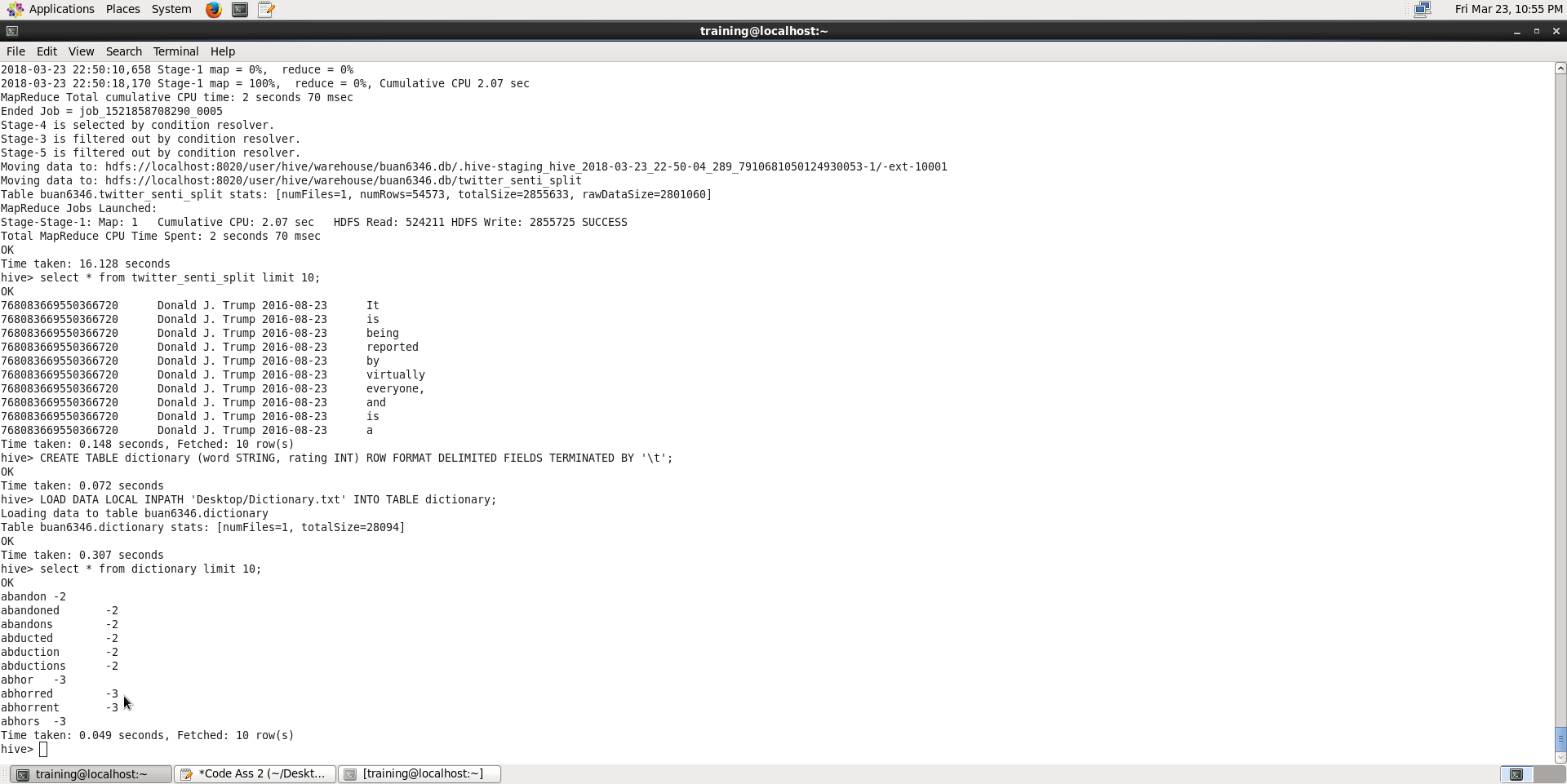


CREATE TABLE twitter\_senti\_split as SELECT id as id,user\_name as user\_name,ts as ts, tweets from twitter\_senti LATERAL VIEW explode(tweet) t as tweets;

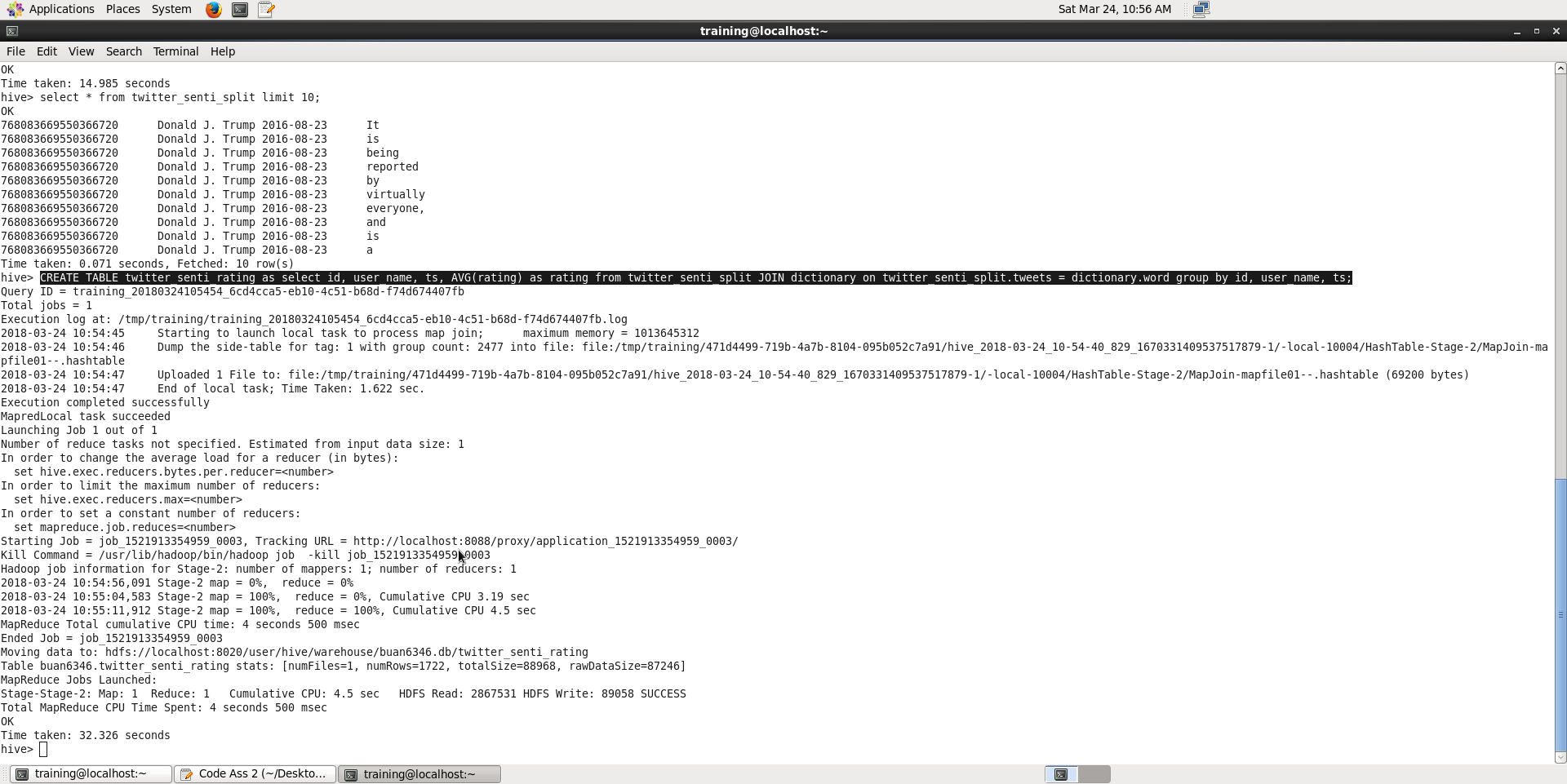


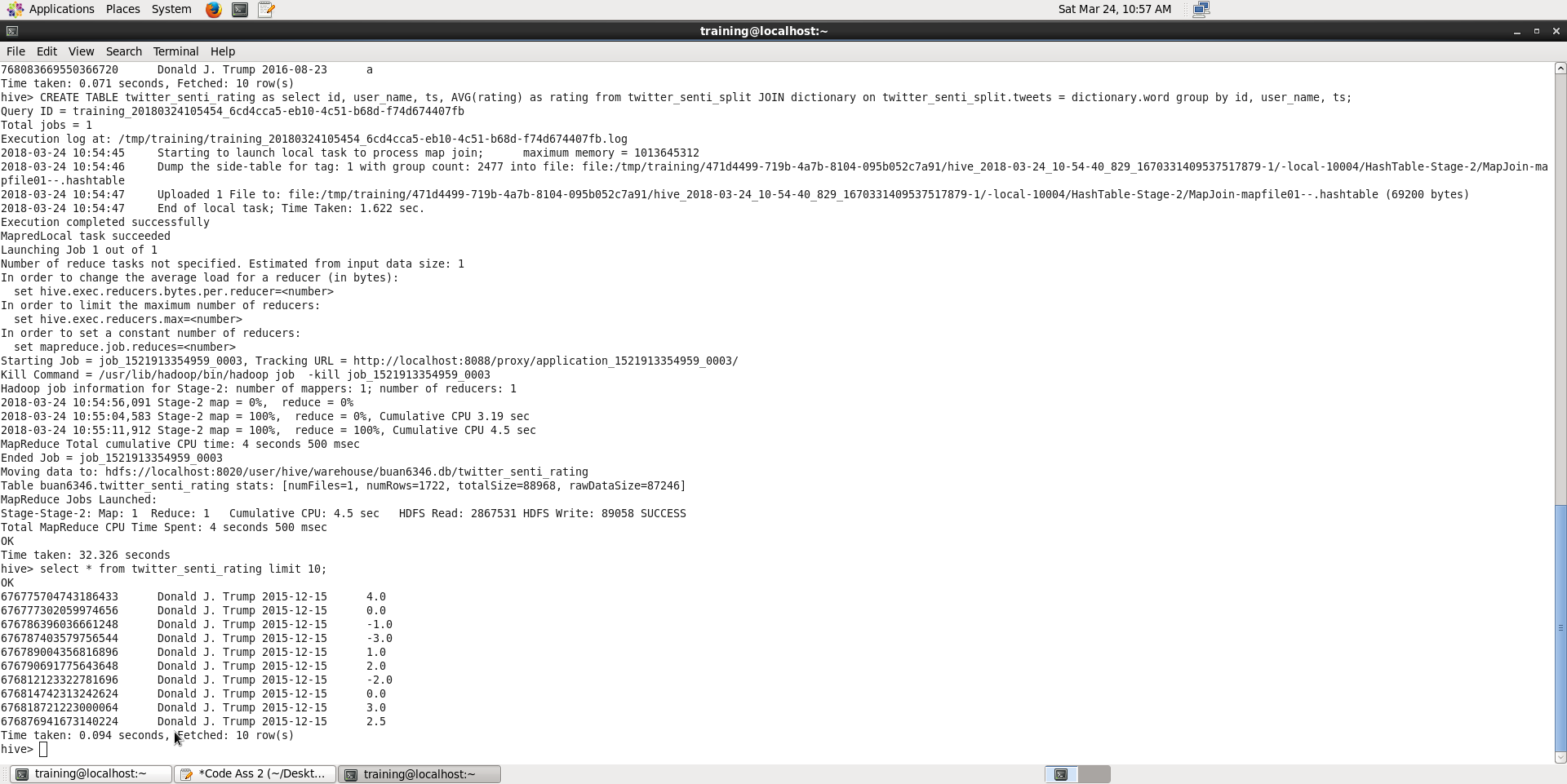


LOAD DATA LOCAL INPATH 'Desktop/Dictionary.txt' INTO TABLE dictionary;

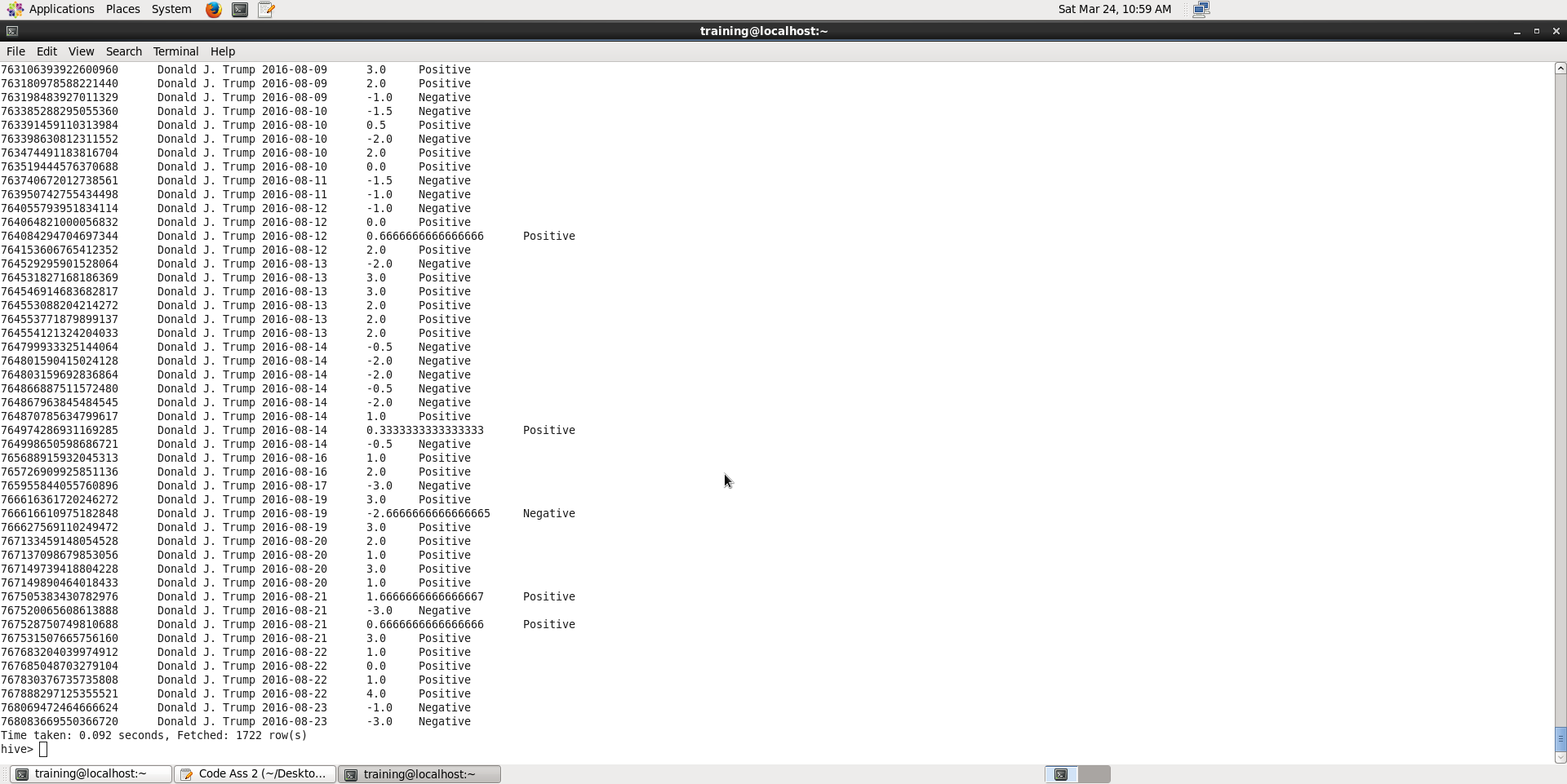


CREATE TABLE twitter\_senti\_rating as select id, user\_name, ts, AVG(rating) as rating from twitter\_senti\_split JOIN dictionary on twitter\_senti\_split.tweets = dictionary.word group by id, user\_name, ts;





select id, user\_name, ts, rating, CASE WHEN rating < 0 THEN 'Negative' ELSE 'Positive' END as sentiment from twitter\_senti\_rating;



1. The sentimental analysis that was done in the previous question was based on the polarity value assigned to each of the words, if the sum is negative then the sentiment is negative and if the sum is positive the sentiment is positive. But, if we analyze the tweet contextually we can end up getting more insights and the exact sentiment.

So, my suggestion would be sentimental analysis based on the context. Ex. The food was not terrible is more like weak positive but the negative words not and terrible might make the entire more negative.