LOADING DATA FILE

```
add jar file:///home/training/Desktop/json-serde-1.3.7-jar-with-dependencies.jar ;
```

```
create table tweet(id bigint,text string,created_at string,retweet_count int,user
struct<location:string,id_str:bigint,name:string,screen_name:string,followers_count
:int>)ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe';
```

```
load data local inpath ’Desktop/Twitter.json’ overwrite into table tweet;
```

1. What are the hashtags used and how many times each are used?

```
SELECT word, count(1) as wcount from tweet LATERAL VIEW
explode(split(regexp_replace(trim(text),"[^#A-Za-z0-9]"," "), ' ')) text_ex as word
WHERE word rlike "^#[a-zA-Z0-9]+$" GROUP BY word ORDER BY wcount;
```

2. What is the most trending hashtag in a day and how many times is it tweeted? [Note: day should be in the format ‘yyyy-mm-dd’]

```
create table tweet_date as select unix_timestamp(created_at, "EEE MMM d
HH:mm:ss Z yyyy") as created_date,text,id,user.name from tweet;
```

```
create table tweet_trending as select from_unixtime(created_date, 'yyyy-MM-dd')
as created_at_date,text,id,name from tweet_date;
```

```
create table tweet_hashtag as SELECT created_at_date, word, count(1) as wcount
from tweet_trending LATERAL VIEW
explode(split(regexp_replace(trim(text),"[^#A-Za-z0-9]"," "), ' ')) text_ex as word
WHERE word rlike "^#[a-zA-Z0-9]+$" GROUP BY word,created_at_date ORDER
BY wcount,created_at_date;
```

```
select a.created_at_date,a.word,a.wcount from(select
created_at_date,word,wcount,rank () over (PARTITION BY created_at_date order
by wcount desc) as rank from tweet_hashtag) as a where rank=1;
```

3. Which state users are most active and how many tweets are posted by State?

```
select user.location,count(*) as c from tweet group by user.location order by c desc
LIMIT 1;
```

```
select count(text) as cnt from tweet where user.location='Oregon';
```

4. Based on the user’s followers count, who are the top ten users who have tweeted?

```
select user.name, user.followers_count c from tweet order by c desc LIMIT 10;
```

5. **What is the score for each tweet that was posted? Does the tweet have a positive or negative sentiment?**

```
create table dictionary(word string,score int) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

load data local inpath ‘Desktop/Dictionary.txt’ into table dictionary;

create table tweet_dictionary as select created_at_date,name,id,text,word from tweet_trending LATERAL VIEW explode(split(regexp_replace(lower(text),"[^#A-Za-z0-9]"," "), ' ')) text_ex as word;

create table tweet_join as select tweet_dictionary.created_at_date,tweet_dictionary.name,tweet_dictionary.id,tweet_dictionary.word,dictionary.score from tweet_dictionary LEFT OUTER JOIN dictionary ON(tweet_dictionary.word=dictionary.word);


select created_at_date,name,id,SUM(score) as score,case when score>0 then'POSITIVE' when score<0 then'NEGATIVE'end as sentiment from tweet_join where score is not null GROUP BY tweet_join.created_at_date,tweet_join.name,tweet_join.id,score order by score DESC;
```