Hive

Data warehouse structure tool
Resides on top of hadoop
Makes data churning easy
Provides sql like queries

MapReduce code takes lot of time

Login to Hive through CLI

NSTRUCTIONS

- Login to the web console using your cloudxlab username and password
- Launch Hive by typing `hive` in the web console

To see the list of all databases type command:

```
show databases;
```

-

To see the list of all tables type command:

```
show tables;
```

-

To create your own database type command:

```
create database <YOUR_USER_NAME>;
```

-

  Kindly remember to substitute `<YOUR_USER_NAME>` with your cloudxlab username

To see the metadata of your database type command:

```
describe database <YOUR_USER_NAME>;
```

-

To use your database type command:

```
use <YOUR_USER_NAME>;
```

-

To create a table `x` in your database type command:

```
create table x (a int);
```

-

To view the data of table `x` type command:

```
select * from x;
```

-

To view the metadata(structure) of the table type command:

```
describe x;
```

-
- To see the metadata and low level details type command:

  ```
  describe formatted x;
  ```

Types of Tables in Hive

Managed tables

- These are internal tables
- Lifecycle managed by hive
- Data is stored in the warehouse directory.
- Dropping the table deletes data from warehouse.

External Tables

- Lifecycle is not managed by hive
- Hive assumes that it does not own the data
- Dropping the table does not delete the underlying data

Managed Tables

- Login to web console

Copy NYSE data from HDFS to your local

```
hadoop fs -copyToLocal /data/NYSE_daily
```

-
- Launch Hive with typing in `hive` on console

Use your own database by using the below command. Replace `YOUR_USER_NAME` with your cloudxlab username

```
use YOUR_USER_NAME;
```

-

Create table `nyse` using below command

```
CREATE TABLE nyse(
exchange1 STRING,
symbol1 STRING,
ymd STRING,
price_open FLOAT,
price_high FLOAT,
price_low FLOAT,
price_close FLOAT,
volume INT,
price_adj_close FLOAT
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

-

See metadata of table using below command

```
DESCRIBE nyse;
```

-

To see more low-level details, type below command

```
DESCRIBE FORMATTED nyse;
```

-

Load data to your `nyse` table

```
use YOUR_USER_NAME;
load data local inpath 'NYSE_daily' overwrite into table nyse;
```

-
- Check the warehouse directory in Hue (in Hue File Store)

Select rows from table (in HIVE console)

```
select * from nyse;
```

-

Loading data from HDFS

```
CREATE TABLE nyse_hdfs(
exchange1 STRING,
symbol1 STRING,
ymd STRING,
price_open FLOAT,
price_high FLOAT,
price_low FLOAT,
price_close FLOAT,
volume INT,
price_adj_close FLOAT
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

-

Run the following command in HIVE console.

```
load data inpath 'hdfs:///user/abhinav9884/NYSE_daily' overwrite into table nyse_hdfs;
```

-
- Above command moves the data from specified location to warehouse.

External Tables

- Login to web console

Copy NYSE data from HDFS to your local

```
hadoop fs -copyToLocal /data/NYSE_daily
```

-
- Launch Hive with typing in `hive` on console

use your own database by using the below command. Replace `YOUR_USER_NAME` with your cloudxlab usernmae

```
use YOUR_USER_NAME;
```

-

Create an external table `nyse_external` using below command

```
    CREATE TABLE nyse_external(
    exchange1 STRING,
    symbol1 STRING,
    ymd STRING,
    price_open FLOAT,
    price_high FLOAT,
    price_low FLOAT,
    price_close FLOAT,
    volume INT,
    price_adj_close FLOAT
    )
    ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    LOCATION '/user/YOUR_USER_NAME/NYSE_daily';
```

-

To see more low-level details, type below command

```
DESCRIBE FORMATTED nyse_external;
```

-
- To drop the external table type

  ```
  DROP TABLE nyse_external;
  ```

Select and Aggregation Queries

To select all columns

```
SELECT * FROM nyse;
```

-

To select only required columns

```
SELECT exchange1, symbol1 FROM nyse;
```

-

Find average opening price for each stock

```
SELECT symbol1, AVG(price_open) AS avg_price FROM nyse
```

- `GROUP BY symbol1;`

Saving Data

- Login to web console
- Launch Hive with typing in `hive` on console

use your own database by using the below command. Replace `YOUR_USER_NAME` with your cloudxlab username

```
use YOUR_USER_NAME;
```

-

To save the data in local file system

```
insert overwrite local directory '/home/abhinav9884/onlycmc'
select * from nyse where symbol1 = 'CMC';
```

-

Hive

To view this data type in the following commands (In the web console)

```
 tail onlycmc/000000_0
```

-

To save data in HDFS

```
insert overwrite directory 'onlycmc' select * from nyse where
```

- `symbol1 = 'CMC';`

DDL - Alter Table

- Login to web console
- Launch Hive with typing in `hive` on console

Use your own database by using the below command. Replace `YOUR_USER_NAME` with your cloudxlab username

```
use YOUR_USER_NAME;
```

-

To rename a table from x to x1

```
ALTER TABLE x RENAME TO x1;
```

-

To change the datatype of a column

```
ALTER TABLE x1 CHANGE a a FLOAT;
```

-
- To add columns to an existing table
  ```
  ALTER TABLE x1 ADD COLUMNS (b FLOAT, c INT);
  ```

Partitions

- Data is located at /data/bdhs/employees/ on HDFS

Copy data to your home directory in HDFS

```
hadoop fs -cp /data/bdhs/employees
```

-

Create table

```
CREATE TABLE employees(
name STRING,
department STRING,
somedate DATE
)
PARTITIONED BY(year STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

-

Load dataset `2012.csv`

```
load data inpath 'hdfs:///user/sandeepgiri9034/employees/2012.csv' into table
employees partition (year=2012);
```

-

Load dataset `2015.csv`

```
load data inpath 'hdfs:///user/sandeepgiri9034/employees/2015.csv' into table
employees partition (year=2015);
```

-
- To view the partitions
  ```
  SHOW PARTITIONS employees;
  ```

Views

To create a view run the commands written below:

```
CREATE VIEW employee_engineering as
SELECT * FROM employees where department = 'Engineering' ;
```

-
- To query from the view run:
  ```
  SELECT * FROM employee_engineering
  ```

Load JSON Data

- Login to the web console
- Launch Hive by typing `hive` in the web console

Add JSON-SERDE JAR using below command:

```
ADD JAR hdfs:///data/serde/json-serde-1.3.6-SNAPSHOT-jar-with-dependencies.jar;
```

-

To create the table use the following command, keep in mind that you have to change `<YOUR_USER_NAME>` to your cloudxlab username:

```
CREATE EXTERNAL TABLE tweets_raw( )
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
```

- `LOCATION '/user/<YOUR_USER_NAME>/senti/upload/data/tweets_raw';`

ORC File Format

Optimized row columnar file format.
Highly efficient to store hive data
Improves performance when reading, writing, processing

- Launch Hive by typing `hive` in the web console

Use your own database by using the below command. Replace `YOUR_USER_NAME` with your cloudxlab username

```
use YOUR_USER_NAME;
```

-

Hive

To create an ORC file format:

```
CREATE TABLE orc_table (first_name STRING, last_name STRING) STORED AS ORC;
```

-

To insert values in the table:

```
INSERT INTO orc_table VALUES ('John','Gill');
```

-
- To retrieve all the values in the table:
  ```
  SELECT * FROM orc_table;
  ```

Recap

- My default the table is in the directory /apps/hive/warehouse/noahsheldon063907
- We can override the location by specifying 'location' in the create table clause.
- Load data copies from Local
- In, Relational database metadata is stored - Hive Metastore
- Dropping external table does not delete the data.

Hive - MovieLens Assignment

`MovieLens` data sets were collected by the GroupLens Research Project at the University of Minnesota.

This data set consists of

1. 100,000 ratings (1-5) from 943 users upon 1682 movies.
2. Each user has rated at least 20 movies.
3. Simple demographic info for the users (age, gender, occupation, zip)

Movielens dataset is located at `/data/ml-100k` in HDFS. Read the `README.md` file to understand the dataset.

We will load the `u.data` file in Hive managed table. `u.data` contains dataset where each row represents userid, movieid, rating, and timestamp fields. Fields are terminated by "`\t`"

INSTRUCTIONS

Hive

1. Copy the data to your home directory in HDFS. Run below commands. Replace **your-username** with your CloudxLab username

Copy the data from `/data` directory in HDFS to your home directory in HDFS. Run below command in Linux console

```
hadoop fs -cp /data/ml-100k/u.data /user/your-username/
```

○

2. Launch hive from the console or launch the Hive editor in Hue. Create a managed table `u_data` in your database in Hive. Run the below commands in. Replace **your-username** and **your-database-name** with your CloudxLab username

Create a database with your CloudxLab username

```
CREATE DATABASE If NOT EXISTS your-username;
```

○

Select your database

```
USE your-database-name;
```

○

Create a table

```
CREATE TABLE IF NOT EXISTS u_data( userid INT, movieid INT, rating INT, unixtime
TIMESTAMP)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE;
```

○

On hive prompt, load the data from your home directory in HDFS. Run below command in Hive query editor in Hue

```
LOAD DATA INPATH 'hdfs:///user/your-username/u.data' overwrite into table u_data;
```

○

3. Check if data is loaded. Go to the warehouse directory at `/apps/hive/warehouse` in the Hue file browser. Select your database name and go inside it. You will see the `u_data` directory. Go inside it and see if data exists.

Project - Sentiment Analysis

Objective

Hive

The objective of the exercise is to do the sentiment analysis based on the tweets data downloaded from Twitter.

We'll do sentiment analysis of movie "Iron Man 3" using Hive and visualize the sentiment data using Tableau.

The dataset containing tweets of "Iron Man 3" movie is located at below location in HDFS

```
/data/SentimentFiles/SentimentFiles/upload/data
```

We'll calculate sentiment using a rudimentary technique. We've polarity of common words in below dictionary file in HDFS

```
/data/SentimentFiles/SentimentFiles/upload/data/dictionary/dictionary.tsv
```

Based on the polarity of words, we will calculate the sentiment of each tweet. You can choose exactly the same steps or use different strategy altogether to calculate the sentiment.

There are various deviations possible, for example:

1. Use pig or spark instead of hive
2. Use a completely different algorithm to compute the sentiment based on NLP
3. Use your own Flume pipeline to download the data (~/sentiment/flume/) and start afresh with a different movie
4. Create your own program to download data from Twitter
5. Use some other mechanism of displaying the data such as D3.js or BIRT

Objective of this step is to copy the **Iron Man 3** movie tweets in your home directory (/user/) in HDFS

After login into your web console, copy the "Iron Man 3" movie tweets to your home directory in HDFS by running the below command. Replace YOUR_USER_NAME with your CloudxLab username.

```
   hadoop fs -cp /data/SentimentFiles /user/YOUR_USER_NAME
```

Objective of this step is to create an external table which contains tweets of **Iron Man 3** movie

Steps-

launch hive console using "hive" command and run below commands on hive

```
ADD JAR hdfs:///data/hive/json-serde-1.1.9.9-Hive13-jar-with-dependencies.jar;
SET hive.support.sql11.reserved.keywords=false;
```

1.

Hive

Select your database. Replace YOUR_USER_NAME with your CloudxLab username. Run below commands

```
CREATE DATABASE IF NOT EXISTS YOUR_USER_NAME;
USE YOUR_USER_NAME;
```

2.

Create _tweets_raw_ external table. It contains details of each tweet. Replace YOUR_USER_NAME with your CloudxLab username

```
CREATE EXTERNAL TABLE tweets_raw (
    id BIGINT,
    created_at STRING,
    source STRING,
    favorited BOOLEAN,
    retweet_count INT,
    retweeted_status STRUCT<
    text:STRING,
    users:STRUCT<screen_name:STRING,name:STRING>>,
    entities STRUCT<
    urls:ARRAY<STRUCT<expanded_url:STRING>>,
    user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,
    hashtags:ARRAY<STRUCT<text:STRING>>>,
    text STRING,
    user STRUCT<
    screen_name:STRING,
    name:STRING,
    friends_count:INT,
    followers_count:INT,
    statuses_count:INT,
    verified:BOOLEAN,
    utc_offset:STRING, -- was INT but nulls are strings
    time_zone:STRING>,
    in_reply_to_screen_name STRING,
    year int,
    month int,
    day int,
    hour int
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES ("ignore.malformed.json" = "true")
LOCATION '/user/YOUR_USER_NAME/SentimentFiles/SentimentFiles/upload/data/tweets_raw';
```

3.

Question-

How many records are in _tweets_raw_ table?

Hint-

Run below query on hive console:

```
SELECT count(id) FROM tweets_raw;
```

Hive

- 89843
- 91786
- 87384

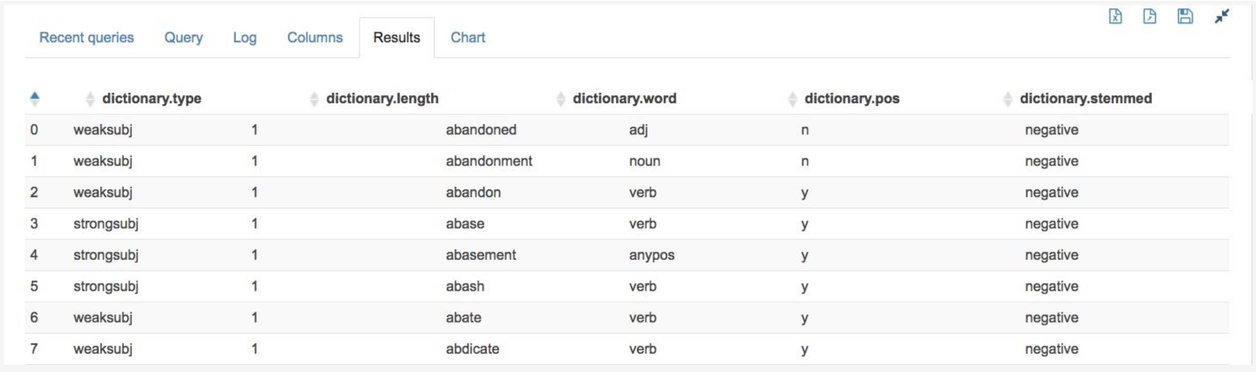
The objective of this step is to create an external table _dictionary_. This table contains English words and their polarity. Polarity means if the word has positive, negative, or neutral sentiment. Since a tweet consists of words, this table will help us in calculating the sentiment of the entire tweet.

Steps-

Create an external table _dictionary_. Run below command in Hive query editor in Hue. Replace YOUR_USER_NAME with your CloudxLab username. _dictionary_ table contains words and their polarity.

```
CREATE EXTERNAL TABLE dictionary (
type string,
length int,
word string,
pos string,
stemmed string,
polarity string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/user/YOUR_USER_NAME/SentimentFiles/SentimentFiles/upload/data/dictionary';
```

1. Sample rows of _dictionary_ table are



	dictionary.type	dictionary.length	dictionary.word	dictionary.pos	dictionary.stemmed	
0	weaksubj	1	abandoned	adj	n	negative
1	weaksubj	1	abandonment	noun	n	negative
2	weaksubj	1	abandon	verb	y	negative
3	strongsubj	1	abase	verb	y	negative
4	strongsubj	1	abasement	anypos	y	negative
5	strongsubj	1	abash	verb	y	negative
6	weaksubj	1	abate	verb	y	negative
7	weaksubj	1	abdicate	verb	y	negative

Question-

How many rows are there in _dictionary_ table?

- 5000
- 7985

Hive

- 8221

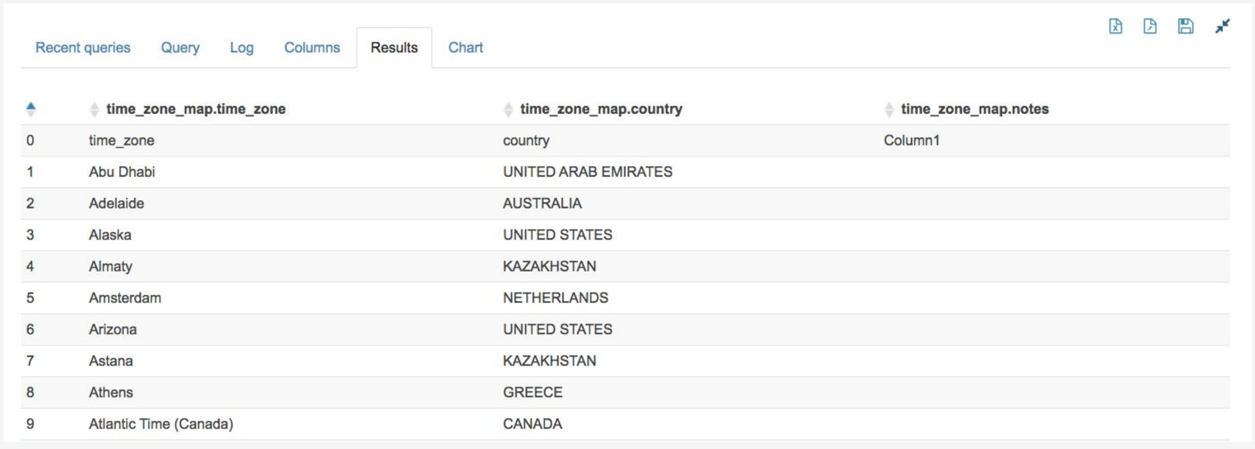
The objective of this step is to create an external table _time_zone_map_. _time_zone_map_ table is a temporary table which is used to map user's timezone in the tweet to the country in the next steps

Steps-

Create an external table _time_zone_map_. Run below command in Hive query editor in Hue. Replace YOUR_USER_NAME with your CloudxLab username.

```
   CREATE EXTERNAL TABLE time_zone_map (
    time_zone string,
    country string,
    notes string
    )
    ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    STORED AS TEXTFILE
    LOCATION
'/user/YOUR_USER_NAME/SentimentFiles/SentimentFiles/upload/data/time_zone_map';
```

Sample rows of _time_zone_map_ table are



	time_zone_map.time_zone	time_zone_map.country	time_zone_map.notes
0	time_zone	country	Column1
1	Abu Dhabi	UNITED ARAB EMIRATES	
2	Adelaide	AUSTRALIA	
3	Alaska	UNITED STATES	
4	Almaty	KAZAKHSTAN	
5	Amsterdam	NETHERLANDS	
6	Arizona	UNITED STATES	
7	Astana	KAZAKHSTAN	
8	Athens	GREECE	
9	Atlantic Time (Canada)	CANADA	

Question-

What is the time zone of country FINLAND?

- Hobart
- Helsinki
- Guadalajara
- Novosibirsk

In this step, we will create _tweets_simple_ and _tweets_clean_ views.

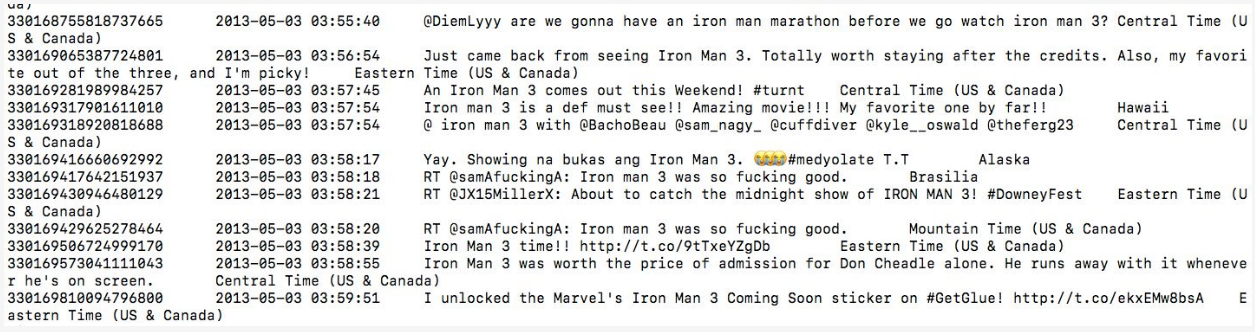
Steps-

Hive

Create view _tweets_simple_. _tweets_simple_ view contains tweet id, the timestamp of the tweet, tweet text and user's time zone. Run below command in the Hive query editor in Hue

```
CREATE VIEW tweets_simple AS
SELECT
id,
cast ( from_unixtime( unix_timestamp(concat( '2013 ', substring(created_at,5,15)),
'yyyy MMM dd hh:mm:ss')) as timestamp) ts,
text,
user.time_zone
FROM tweets_raw;
```

1. Sample rows of _tweets_simple_ views are

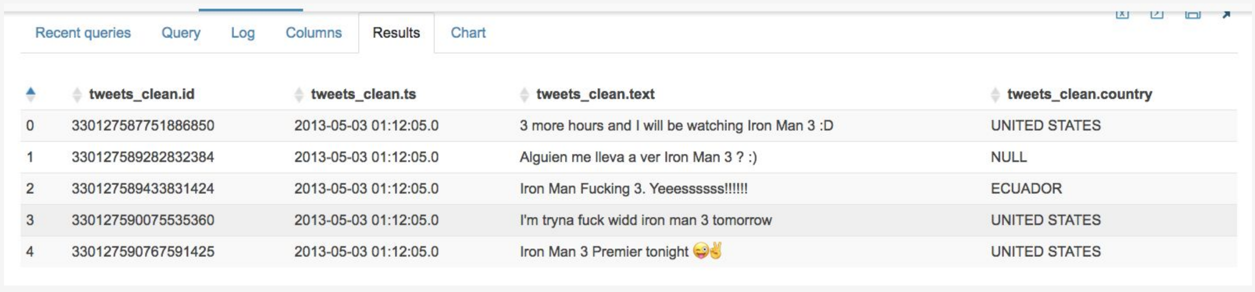


```
da)
330168755818737665      2013-05-03 03:55:40     @DiemLyyy are we gonna have an iron man marathon before we go watch iron man 3? Central Time (U
S & Canada)
330169065387724801      2013-05-03 03:56:54     Just came back from seeing Iron Man 3. Totally worth staying after the credits. Also, my favori
te out of the three, and I'm picky!     Eastern Time (US & Canada)
330169281989984257      2013-05-03 03:57:45     An Iron Man 3 comes out this Weekend! #turnt    Central Time (US & Canada)
330169317901611010      2013-05-03 03:57:54     Iron man 3 is a def must see!! Amazing movie!!! My favorite one by far!!        Hawaii
330169318920818688      2013-05-03 03:57:54     @ iron man 3 with @BachoBeau @sam_nagy_ @cuffdiver @kyle__oswald @theferg23     Central Time (U
S & Canada)
330169416660692992      2013-05-03 03:58:17     Yay. Showing na bukas ang Iron Man 3. 😭😭😭#medyolate T.T        Alaska
330169417642151937      2013-05-03 03:58:18     RT @samAfuckingA: Iron man 3 was so fucking good.       Brasilia
330169430946480129      2013-05-03 03:58:21     RT @JX15MillerX: About to catch the midnight show of IRON MAN 3! #DowneyFest    Eastern Time (U
S & Canada)
330169429625278464      2013-05-03 03:58:20     RT @samAfuckingA: Iron man 3 was so fucking good.       Mountain Time (US & Canada)
330169506724999170      2013-05-03 03:58:39     Iron Man 3 time!! http://t.co/9tTxeYZgDb        Eastern Time (US & Canada)
330169573041111043      2013-05-03 03:58:55     Iron Man 3 was worth the price of admission for Don Cheadle alone. He runs away with it wheneve
r he's on screen.       Central Time (US & Canada)
330169810094796800      2013-05-03 03:59:51     I unlocked the Marvel's Iron Man 3 Coming Soon sticker on #GetGlue! http://t.co/ekxEMw8bsA    E
astern Time (US & Canada)
```

Create view _tweets_clean_. _tweets_clean_ view maps user's timezone to the country.Each row of the _tweets_clean_ view contains tweet_id, the timestamp of the tweet, tweet text and user's country (which is derived from time zone). Run below command in the Hive query editor in Hue

```
CREATE VIEW tweets_clean AS
SELECT
id,
ts,
text,
m.country
FROM tweets_simple t LEFT OUTER JOIN time_zone_map m ON t.time_zone = m.time_zone;
```

2. Sample rows of _tweets_simple_ views are



	tweets_clean.id	tweets_clean.ts	tweets_clean.text	tweets_clean.country
0	330127587751886850	2013-05-03 01:12:05.0	3 more hours and I will be watching Iron Man 3 :D	UNITED STATES
1	330127589282832384	2013-05-03 01:12:05.0	Alguien me lleva a ver Iron Man 3 ? :)	NULL
2	330127589433831424	2013-05-03 01:12:05.0	Iron Man Fucking 3. Yeeessssss!!!!!!	ECUADOR
3	330127590075535360	2013-05-03 01:12:05.0	I'm tryna fuck widd iron man 3 tomorrow	UNITED STATES
4	330127590767591425	2013-05-03 01:12:05.0	Iron Man 3 Premier tonight 😜✌	UNITED STATES

Hive

Question-

From which country tweet with id 330044004693598208 was tweeted?

- UNITED STATES
- ARGENTINA
- CANADA

In this step, we will create _l1_, _l2_ and _l3_ views which will help us in calculating sentiment of each tweet.

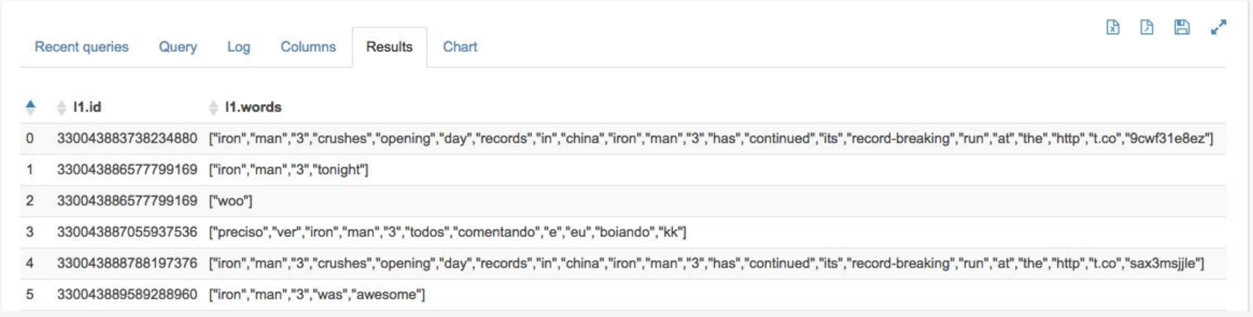
Steps

Create view _l1_. _l1_ view converts each tweet into lower case and explodes it into a list of words. Run below command in Hive query editor in Hue.

```
create view l1 as select id, words from tweets_raw lateral view
explode(sentences(lower(text))) dummy as words;
```

1.

Sample rows of view _l1_ are



	l1.id	l1.words
0	330043883738234880	["iron","man","3","crushes","opening","day","records","in","china","iron","man","3","has","continued","its","record-breaking","run","at","the","http","t.co","9cwf31e8ez"]
1	330043886577799169	["iron","man","3","tonight"]
2	330043886577799169	["woo"]
3	330043887055937536	["preciso","ver","iron","man","3","todos","comentando","e","eu","boiando","kk"]
4	330043888788197376	["iron","man","3","crushes","opening","day","records","in","china","iron","man","3","has","continued","its","record-breaking","run","at","the","http","t.co","sax3msjjle"]
5	330043889589288960	["iron","man","3","was","awesome"]

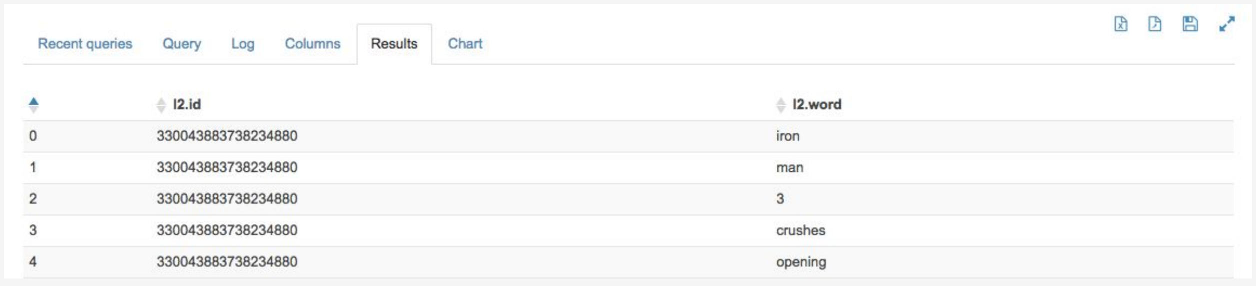
Create view _l2_. _l2_ view stores every word of a tweet in a new row. Run below command in Hive query editor in Hue.

```
create view l2 as select id, word from l1 lateral view explode( words ) dummy as word
;
```

2.

Sample rows of view _l2_ are

Hive



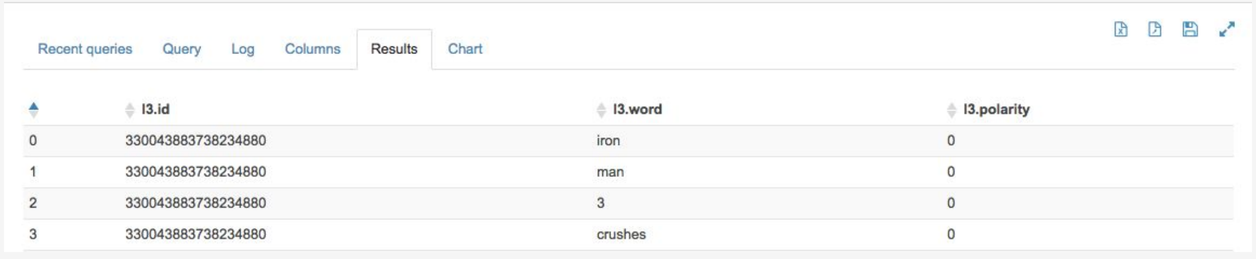
	l2.id	l2.word
0	330043883738234880	iron
1	330043883738234880	man
2	330043883738234880	3
3	330043883738234880	crushes
4	330043883738234880	opening

Create view _l3_. _l3_ view joins _l2_ view with _dictionary_ table and stores polarity of each word. Run below command in Hive query editor in Hue.

```
create view l3 as select
id,
l2.word,
case d.polarity
when  'negative' then -1
when 'positive' then 1
else 0 end as polarity
from l2 left outer join dictionary d on l2.word = d.word;
```

3.

Sample rows of view _l3_ are



	l3.id	l3.word	l3.polarity
0	330043883738234880	iron	0
1	330043883738234880	man	0
2	330043883738234880	3	0
3	330043883738234880	crushes	0

Question-

What is the polarity of word "crushes"?

- 0
- 1
- 2
- 3

In this step, we create a new table _tweetsbi_. We join _tweets_clean_ and _tweets_sentiment_ tables and store sentiment of each tweet. Each row of the _tweetsbi_ table contains tweet id, timestamp, tweet text, country and its sentiment

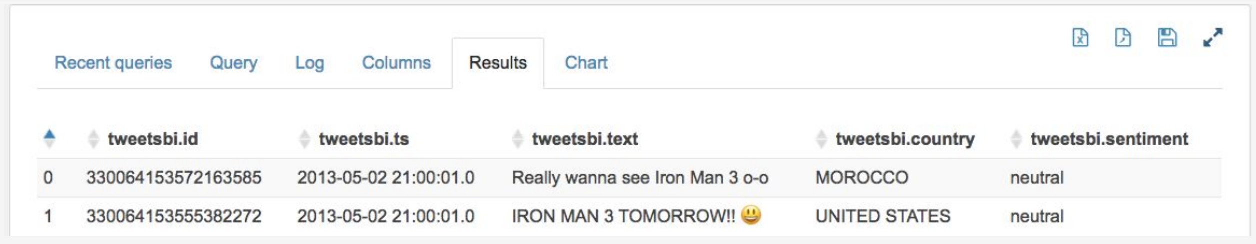
Hive

Steps-

Create _tweetsbi_ table. Run the below command in Hive query editor in Hue.

```
CREATE TABLE tweetsbi
STORED AS ORC
AS
SELECT
t.*,
s.sentiment
FROM tweets_clean t LEFT OUTER JOIN tweets_sentiment s on t.id = s.id;
```

1. Sample rows of _tweetsbi_ table are



	tweetsbi.id	tweetsbi.ts	tweetsbi.text	tweetsbi.country	tweetsbi.sentiment
0	330064153572163585	2013-05-02 21:00:01.0	Really wanna see Iron Man 3 o-o	MOROCCO	neutral
1	330064153555382272	2013-05-02 21:00:01.0	IRON MAN 3 TOMORROW!! 😃	UNITED STATES	neutral

Now we have a sentiment of each tweet along with country from where this tweet was tweeted. In the next steps, we will visualize the sentiment of "Iron Man 3" movie in different countries.

Question-

What is the country and sentiment of the tweet with id as 330043924896968707?

- SPAIN, neutral
- INDIA, negative
- MOROCCO, positive
- None of the above

Create table nyse (

stockexchange STRING,

symbol STRING,

ymd STRING,

price_open FLOAT,

price_high FLOAT,

price_low FLOAT,

price_close FLOAT,

volume INT,

price_adj_close FLOAT

)

```
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

```
load data local inpath 'NYSE_daily' overwrite into table nyse;
```

```
load data inpath 'hdfs:///user/noahsheldon063907/NYSE_daily/NYSE_daily' into table nyse;
```

stockexchange STRING, **symbol** STRING, **ymd** STRING, **price_open** FLOAT, **price_high** FLOAT, **price_low** FLOAT, **price_close** FLOAT, **volume** INT, **price_adj_close** FLOAT

```
CREATE TABLE nyse_hdfs(
exchange1 STRING,
symbol1 STRING,
ymd STRING,
price_open FLOAT,
price_high FLOAT,
price_low FLOAT,
price_close FLOAT,
volume INT,
price_adj_close FLOAT
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```